Project Title	Wide – Impact cyber Security Risk framework
Project Acronym	WISER
Grant Agreement No	653321
Instrument	Innovation Action
Thematic Priority	Cybersecurity, Privacy & Trust, Risk Management, Assurance Models
Start Date of Project	01.06.2015
Duration of Project	30 Months
Project Website	www.cyberwiser.eu

D3.4 CYBER RISK MODELLING LANGUAGE AND GUIDELINES, FINAL VERSION

Work Package	WP 3, WISER Modelling
Lead Author (Org)	Atle Refsdal (SINTEF), Gencer Erdogan (SINTEF)
Contributing Author(s) (Org)	Giorgio Aprile (AON), Sara Poidomani (AON), Romina Colciago (AON), Alejandra Gonzalez (AON), Antonio Alvarez (ATOS), Susana González (ATOS), Carlos Hernán Arce (ATOS), Paolo Lombardi (Trust-IT), Roberto Mannella (REXEL)
Due Date	31.03.2017
Date	29.03.2017
Version	1.0

Dissemination Level

X	PU: Public
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

The work described in this document has been conducted within the project WISER. This project has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 653321. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content.

Versioning and contribution history

Version	Date	Author	Notes
0.1	06.02.2017	Atle Refsdal (SINTEF)	Initial version based on D3.2.
0.2	19.02.2017	Alejandra Gonzalez (AON)	Section 8 rewritten and appendices Appendix II and Appendix III introduced.
0.3	23.02.2017	Atle Refsdal (SINTEF)	Inserted new section heading for integration of assessment algorithms in the Risk Assessment Engine
0.4	27.02.2017	Atle Refsdal (SINTEF)	Updates in Sect. 1
0.5	08.03.2017	Atle Refsdal (SINTEF)	Updates in Sect. 2, removed outdated appendix.
0.6	08.03.2017	Atle Refsdal (SINTEF)	Updates in Sect. 8, removed outdated appendices and inserted new appendix for validation. Minor updates in other sections.
0.7	10.03.2017	Atle Refsdal (SINTEF)	Correction of responsibility assingnments for Sect. 11 and Sect. 12
0.8	10.03.2017	Antonio Álvarez (ATOS), Alejandra González (AON)	Contribution to section 12.
0.9	10.03.2017	Susana González (Atos), Carlos Hernan Arce Plata (Atos)	Added Section 10
0.10	10.03.2017	Atle Refsdal (SINTEF)	Added contents in Appendix V
0.11	10.03.2017	Alberto Biasibetti (AON)	Input to Section 11
0.12	13.03.2017	Alejandra González (AON), Antonio Álvarez (ATOS)	Refinement of Section 12
0.13	14.03.2017	Atle Refsdal (SINTEF)	Updated Section 1, some minor corrections
0.14	14.03.2017	Atle Refsdal (SINTEF)	Updated Executive Summary, Section 13 and Figure 13
0.15	14.03.2017	Alberto Biasibetti (AON), Alejandra González (AON)	Refinaments to Section 11. References [24]--[27] added.

0.16	15.03.2017	Atle Refsdal (SINTEF)	Minor corrections and updates in several sections.
0.17	15.03.2017	Gencer Erdogan (SINTEF)	Section 5 – target modelling.
0.18	15.03.2017	Atle Refsdal (SINTEF)	Minor updates in Executive summary, Sections 1, 5 and 13, consolidation for internal review
0.19	20.03.2017	Alberto Biasibetti (AON), Romina Colciago (AON), Alejandra González (AON)	Internal review, first round.
0.20	20.03.2017	Antonio Álvarez (ATOS)	Internal review, first round
0.21	21.03.2017	Alberto Biasibetti (AON), Alejandra González (AON)	Updates after review: minor corrections to sections 4.2, 8 and 11. References [29]--[33] added.
0.22	22.03.2017	Atle Refsdal (SINTEF)	Corrections and changes after first internal review
0.23	22.03.2017	Carlos Hernan Arce (ATOS)	Addressing comments to sections 10.2.5, 10.2.6, and 10.2.7
0.24	22.03.2017	Susana González Zarzosa (ATOS)	Addressed comments in chapter 10.
0.25	23.03.2017	Atle Refsdal (SINTEF)	Minor corrections in several sections, deleted comments that have been addressed
0.26	23.03.2017	Atle Refsdal (SINTEF)	Accepted all track changes and removed all comments (except one in Section 10.2.1)
0.27	27.03.2017	Alejandra Gonzalez (AON)	Second internal review and minor corrections in several sections.
0.28	28.03.2017	Carlos Hernan Arce (ATOS), Susana González Zarzosa (ATOS)	Corrections to sections 10.2.1 and 10.2.7
1.0	29.03.2017	Atle Refsdal (SINTEF)	Clean version for submission

Disclaimer

Table of Contents

List of Tables

List of Figures

Executive Summary

This document reports on the work carried out to offer support for cyber risk modelling in the WISER framework. This support is provided in the form of modelling languages and corresponding guidelines and structures that are specifically aimed at facilitating development of cyber risk models to fulfil the role implied by the framework design documented in D2.3 and implemented in D2.4.

Developing the parts of the design that define the role of the cyber risk modelling and its interaction with the other parts of the framework was an important part of the work undertaken within WP3. Understanding this role is important. A description of the role of cyber risk modelling is therefore included in this report. The main goal of the modelling is to arrive at executable assessment algorithms that can be used for continuous cyber risk monitoring based on dynamically updated indicators obtained from the WISER infrastructure. These algorithms are also able to trigger proposals for mitigation options if the risk level reaches a defined threshold.

We describe the overall method for cyber risk modelling and provide guidelines for how to employ each of the selected languages in the context of the WISER framework. The modelling method consists of two main steps. The purpose of Step 1 is to establish and document an understanding of the ways in which cyber risks may materialize. This is done using the CORAS risk modelling language [15]. The purpose of Step 2 is to define an algorithm for continuous risk monitoring based on dynamic indicators, using either **R** [23] for quantitative assessments or DEXi [7] for qualitative assessments. By supporting both quantitative and qualitative assessment, we cater to different user needs and service delivery models. Our rationale for selecting the languages CORAS, **R** and DEXi is explained, and we give an overview of each of these languages.

To support development of models and algorithms, we offer guidelines for how to develop CORAS models in Step 1, as well as for defining assessment algorithms in Step 2 based on CORAS models. Separate guidelines are provided for using either **R** for quantitative assessments or DEXi for qualitative assessments. Here we take a modular approach, explaining how to represent fragments of a CORAS diagram by corresponding fragments of an **R** script or DEXi model. A separate section is also dedicated to explaining the integration of the assessment algorithms in the Risk Assessment Engine, which is responsible for their runtime execution.

Following the guidelines referred to above, we developed quantitative and qualitative assessment algorithms for each of the 10 risk patterns documented in D3.1. Our method for validating these algorithms can be understood as a refinement of the overall method guidelines that was specifically tailored to our context in the WISER project. A detailed description of this particular validation approach, illustrated by an example, is therefore provided in an appendix.

In the context of WISER, CORAS models are primarily meant to serve as an aid in the identification of threats, vulnerabilities, potential incidents and assets, as well as for assessing the likelihood of incidents. For assessment of the economical or societal impact of risk, we offer detailed guidelines that go beyond the support offered by CORAS. Notice, moreover, that use of CORAS is optional from a purely technical point of view. This is because the purpose of the CORAS models in WISER is to help establish and document an understanding of cyber-risk meant for human actors to support the development of machine-readable algorithms. Therefore, although we think that CORAS is very well suited for the purpose, users are free to use any other approach if they prefer. Of course, WISER cannot then provide specific guidelines.

When developing risk models, it is important to establish a good understanding of the target of analysis. We therefore provide guidelines for target modelling. For users who want to use the ready-made WISER risk patterns, the target can be captured simply by selecting characteristics from a predefined list. For more advanced users who wish to develop risk models tailored to their specific target, we offer guidelines that are more general. These do not prescribe any particular modelling language, as users will have very different preferences, depending on their training and background, as well as the desired abstraction level of the analysis. Moreover, these guidelines are purely methodological and do not in any way depend on the technical WISER infrastructure, which means that users are free to perform and document the target modelling in any way they prefer.

1 Introduction

1.1 Purpose and Scope

This document presents the final report on the work carried out to provide support for cyber risk modelling in the WISER framework. This support is provided in the form of modelling languages and corresponding guidelines that are specifically aimed at facilitating development of models to fulfil the role implied by the framework design as defined by WP2. In addition to guidelines for the modelling of risk models and corresponding assessment algorithms, we provide dedicated guidelines for modelling the target of analysis, i.e. the system for which cyber risks are assessed. Guidelines are also provided for economic and societal impact assessment.

Developing the aspects of the design that define the role of the cyber risk modelling and its interaction with the other elements of the framework was an important part of the work undertaken within WP3. A description of the role of cyber risk modelling is included in this report. A separate section on the more technical integration of the risk assessment algorithms resulting from the risk modelling in the technical framework (more specifically, the Risk Assessment Engine) is also offered.

In order to cater to different types of user needs and service delivery models, WISER supports qualitative as well as quantitative modelling and assessment. While quantitative assessment is clearly needed for any form of detailed economic impact assessment, some clients will prefer a qualitative approach for other purposes due to its perceived simplicity. Such an approach may also be better suited for assessing risk with respect to assets that can sometimes be hard to measure in terms of monetary values, such as reputation and societal impact.

As the title suggests, a preliminary version of this document, D3.2, was submitted earlier in the project, more specifically at M12 (May 2016). However, D3.4 can be read as a standalone document, covering the final version of all aspects of the cyber risk modelling language and guidelines at the point in time where WP3 terminates, which is at M22 (March 2017). This means that D3.4 contains some sections that are completely new, as well as some sections that are more or less heavily modified versions of similar sections in D3.2. For readers already familiar with D3.2, an overview of the relations between the contents of D3.2 and D3.4 is offered in Section 1.3.

While there is always some degree of overlap between work packages in a project like WISER, D3.4 represents the main source of documentation for all WP3 tasks except T3.3 (Cyber risk patterns), which is documented in D3.1, and T3.7 (Communication and protocols), which is partly documented in D3.3 and partly in Section 10 of D3.4.

1.2 Structure of the document

After completing the introduction in this section, we continue by explaining the role of the cyber risk modelling in the WISER framework in Section 2. Next, in Section 3 we explain our rationale for selecting the three WISER risk modelling languages, which are CORAS for human-readable risk models, DEXi for qualitative risk assessment algorithms, and **R** for quantitative risk algorithms. In Section 4, we give a short overview of each of these languages, in order to provide some background for the rest of the document aimed at readers not familiar with the languages, as well as references for further information. We then move on to the actual guidelines. In Section 5, we present simple guidelines for modelling the target of analysis, while Section 6 presents the overall method for cyber risk modelling in WISER. This overall method is the same whether one chooses to use qualitative or quantitative assessment. Section 7 provides specific guidelines for creating CORAS models, which is the first step of the overall method and is performed independently of whether qualitative or quantitative assessment will be used. In Section 8, we give guidelines for quantitative algorithms using **R**, while Section 9 offers similar guidelines for defining qualitative assessment algorithms based on a CORAS diagram using DEXi. Having thus provided methodological support for developing assessment algorithms, we explain the integration of such algorithms in the Risk Assessment Engine in Section 10.

Although the representation of impact assessment in the **R** and DEXi algorithms is included in Section 8 and Section 9, these sections do not provide thorough methodological guidelines for impact assessment. In Section 11, we present such guidelines for economic impact assessment, while societal impact assessment is addressed in Section 12. Notice that, unlike Section 8 and Section 9, the guidelines provided in Section 11 and Section 12 are meant to support users making assessments (typically in order to be able to answer business configuration questions), rather than to create algorithms to be executed by the Risk Assessment Engine. We present our conclusion in Section 13.

This document also contains five appendices. Appendix I shows the naming convention used for model elements occurring in CORAS diagrams and corresponding R and DEXi algorithms. Appendix II introduces Bayesian networks for risk modelling, which play an important role in the development of the quantitative algorithms. In Appendix III, we present numerical results from a worked example of application of the guidelines for quantitative algorithms. Appendix IV shows the result of application of the guidelines for qualitative algorithms. Finally, in Appendix V we give a detailed description of the method we have used to validate the quantitative and qualitative algorithms for the ten risk patterns that were documented in D3.1.

1.3 Relationship to other project outcomes

By motivating the selection of modelling languages and offering guidelines for their use in the WISER framework, this document explains how to fulfil the intended role of cyber risk modelling in the overall framework. It therefore complements and builds on the overall framework design and descriptions provided by WP2, in particular D2.3 (Framework design, final version). It also relates to WP4 outcomes by the use of indicators obtained from the monitoring infrastructure, which serve as input for the risk assessment algorithms developed as a part of the cyber risk modelling. During run-time, the risk assessments algorithms are executed by the Risk Assessment Engine (see Section 10) to provide risk level assessments (and triggers mitigation proposals) to be presented to end users in the dashboard to support decision-making; this relates the work presented here with the WP5 outcomes. Moreover, the cost-benefit analysis, which is one aspect of T3.6, has been documented in D5.1 (Section 5.3) and D5.2 (Section 5.1.3), since the support for cost-benefit analysis is provided in the Decision Support _System. We try out the modelling approach and resulting algorithms in the context of the pilots performed in WP6. The role of the modelling approach in the WISER exploitation plan and business model is explained by WP8, more specifically by D8.7 (Exploitation plan & business models, first version) and the upcoming D8.8 (Exploitation plan & business models, final version)

Obviously, D3.4 document is also closely related to D3.3 (Cyber risk modelling tool), which presents the editors supporting the method and languages presented here, and to D3.1 (Cyber risk patterns), which offers risk patterns in the form of CORAS diagrams. The guidelines presented here support the creation of assessment algorithms expressed using **R** or DEXi based on those patterns. The closest relationship, however, is to the preliminary version of the cyber risk modelling languages and guidelines, i.e. D3.2. Table 1 provides an overview of the relation between D3.4 and D3.2. This table also shows to which task each section of D3.4 is most closely related.

D3.4 section	D3.2 section	Relation/comment	Main task(s)
1 Introduction	1 Introduction	Updated.	All WP3 tasks
2 The role of risk modelling in WISER	2 The role of risk modelling in WISER	Updated to address inclusion of treatment proposals in risk models.	All WP3 tasks
3 Rationale for selection of modelling languages	3 Rationale for selection of modelling languages	No significant update.	T3.2
4 Overview of the selected risk modelling languages	4 Overview of the selected risk modelling languages	No significant update.	T3.2
5 Target modelling	N/A	This is a new section in D3.4.	T3.1
6 Overall method for cyber risk modelling	5 Overall method for cyber risk modelling	A new small subsection was added to refer to the specific method for validation of the algorithms for the risk patterns in WISER. (The method is described in detail in Appendix V.)	T3.2
7 Creating a CORAS diagram with indicators	6 Guidelines for creating a CORAS diagram with indicators	No significant update.	T3.2
8 Defining quantitative assessment algorithms using **R** from CORAS diagrams	8 Defining quantitative assessment algorithms using **R** from CORAS diagrams	Completely rewritten in order to describe the combined use of an actuarial approach exploiting a Bayesian Network variant and the pure CORAS calculus that was described in D3.2.	T3.2
9 Defining qualitative assessment algorithms using DEXi from CORAS diagrams	7 Defining qualitative assessment algorithms using DEXi from CORAS diagrams	No significant update.	T3.2
10 Integration of assessment algorithms in the Risk Assessment Engine	N/A	This is a new section in D3.4.	T3.7
11 Economic impact assessment	9 Economic impact assessment	This has been significantly updated. Since cyber insurance is a general mitigation strategy that is closely related to economic impact assessment, insurance	T3.5, T3.6

		policy guidelines have been included. Please notice that the general cost-benefit analysis has been documented in D5.1 and D5.2, since the support for such analysis is provided in the Decision Support System.	
12 Societal impact assessment	10 Societal impact assessment	This has been significantly updated. A DEXi model for assessment of societal impact has been included.	T3.4
13 Conclusions	11 Conclusions	Updated	All WP3 tasks
14 References	12 References		
App. I Naming conventions for CORAS, DEXi and R model elements	App. II Naming conventions for CORAS, DEXi and R model elements	No significant update.	T3.2
App. II Bayesian networks for risk modelling	N/A	This is a new appendix in D3.4.	T3.2, T3.5
App. III Numerical results for the worked example in Section 8.3	N/A	This is a new appendix in D3.4.	T3.2, T3.5
App. IV Guidelines application example: DEXi model	App. IV Guidelines application example: DEXi model	No significant update.	T3.2
App. V Validation of the WISER risk pattern algorithms	N/A	This is a new appendix in D3.4.	T3.2, T3.5

Table 1. Relation between D3.4 and D3.2

2 The role of cyber risk modelling in WISER

The overall goal for the risk modelling in WISER is to provide machine-readable risk assessment algorithms (sometimes referred to as Model rules) that can be executed in real-time by the Risk Assessment Engine in order to provide a list of risks along with a risk level assessment for each risk, to be presented to the end-user via the dashboard. The algorithms also provide proposals for mitigation options (i.e. potential countermeasures to reduce the risk) that are triggered if the risk level reaches a defined threshold. The risk level is determined by the following two factors:

1. The likelihood of the incident (a successful attack causing harm to one or more assets) to occur.
2. The impact (i.e. degree of damage) caused by the incident on the asset(s).

To facilitate risk level assessment, these two factors are further decomposed, as shown in later sections.

Assessments are based on the information collected by the WISER framework through the business configuration questionnaire, the vulnerability scan results, the network layer sensors and the application layer sensors. In the context of the risk modelling, we refer to such information pieces as *indicators*. There are four different types of indicators:

- *Business configuration indicators* are obtained manually through single/multiple-choice questions asked to the user when configuring WISER.
- *Vulnerability test result indicators* are obtained through non-intrusive vulnerability scans initiated by the user.
- *Network monitoring indicators* are obtained from network-layer sensors deployed in the running target infrastructure.
- *Application monitoring indicators* are obtained from application-layer sensors deployed in the running target infrastructure.

The former two will be updated only when initiated by a person, either by making changes to the business configuration or by triggering a new vulnerability scan. The interval between updates is therefore expected to be in the order of weeks, months, or even longer. The latter two, on the other hand, are continuously and automatically updated by the sensors. Notice, however, that indicators do not generally represent direct sensor readings, as some processing and aggregation of observed events is performed by the monitoring infrastructure offered by WP4.

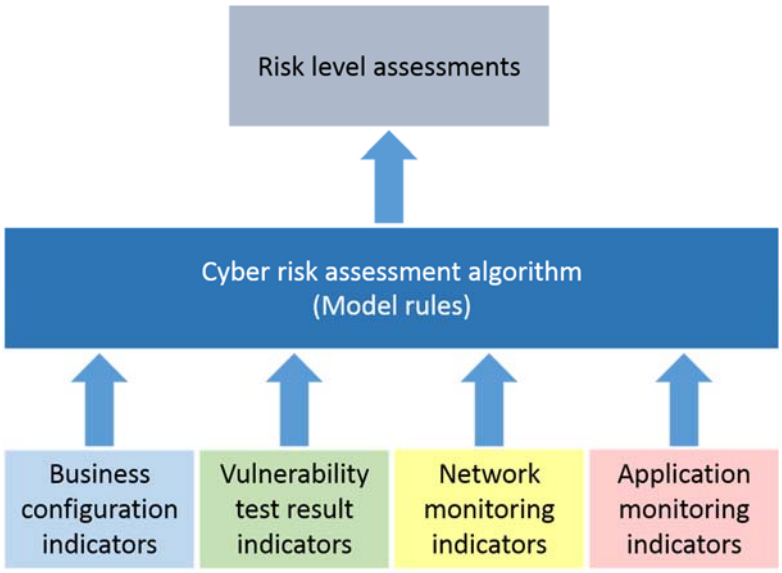


Figure 1. Definition of aggregation from indicators to risk level assessments

Figure 1 illustrates how the cyber risk assessment algorithms define the aggregation of all available and relevant information obtained by the WISER infrastructure in order to provide a continuously updated assessment of the risk level of identified risks. This means that the algorithms define the bridge between the (relatively) detailed information represented by the indicators and the risk information needed to support decision-making at the business level.

A risk assessment algorithm thus defines, in a formal machine-readable language, how to derive risk level assessments from a set of indicators, as well as conditions for triggering proposals for mitigation options. Defining this algorithm is the task of a human modeller[1]. However, before doing this, the modeller needs to establish a good understanding of the assets, relevant risks, the ways in which these risks may materialize, and the relation between these elements and the available indicators that can be employed to assess the risk and the involved threats, vulnerabilities and threat scenarios. Furthermore, the modeller needs to identify the mitigation options of relevance for the risks in question. The overall method for cyber risk modelling in WISER therefore consists of the two main steps illustrated in Figure 2.

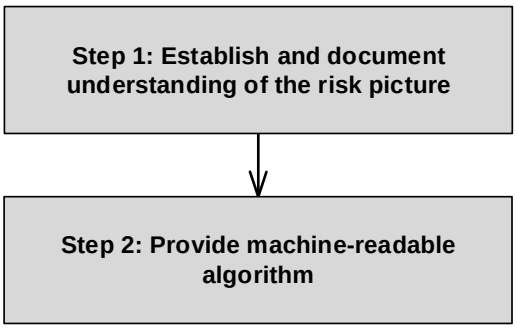


Figure 2. Outline of overall method for cyber risk modelling

This method is further refined and described in Section 5; we include the outline here in order to prepare for the discussion of the languages in the next sections.

[1] For simplicity, we write as if there is only one modeller, although it may equally well be a team.

The first step is performed using CORAS [15] as the modelling language. For the machine-readable algorithms to be developed in the second step of the method, WISER adopted two different languages to support qualitative as well as quantitative assessments in order to cater to different client preferences and service delivery modes. In the next section, we motivate the selection of languages.

3 Rationale for selection of modelling languages

The two steps of the method described in Figure 2 (and further refined in Section 5) serve quite different purposes. Therefore, the set of requirements for the modelling languages for these steps are not identical. We therefore present them in separate subsections. In the following, we motivate the requirements and discuss how these are met by the selected modelling languages. We use the notation LR1a/b/c.., for language requirements to Step 1 and LR2a/b/c... for language requirements for step 2.

Notice that in this section we will use the term 'requirements' in the 'soft' sense. This means that they do not necessarily refer to properties or conditions that can be objectively established to either hold or not, but rather to properties or conditions that can in general hold to a greater or lesser degree, depending at least partly on subjective opinion.

3.1 Modelling language for establishing and documenting the risk picture

3.1.1 Requirements

The purpose of the first step of the method outlined in Figure 2 is to establish and document the risk picture. This risk picture is used as a starting point for defining the risk assessment algorithm in step 2 of the method. It also serves as an aid to help communicate to other human stakeholders what threats, vulnerabilities, risks, and assets are considered in the model, and how these elements relate to each other. We therefore include the following requirement:

> LR1a: *The language should be simple to understand for human stakeholders without requiring extensive training*.

Risk related concepts, such as threats and vulnerabilities, are frequently mixed up when discussing risk. To communicate clearly, avoid misunderstandings, and generally facilitate clear reasoning, it is necessary to be consistent and explicit when using such concepts. We therefore include the following requirement:

> LR1b: *The language should include risk concepts such as threat, vulnerability, incident, risk and asset as first class entities*.

The indicators obtained from the WISER infrastructure represent the dynamic input on which we base the WISER risk assessments. Hence, tying these indicators to the appropriate elements of a risk model is essential. We therefore include the following requirement:

> LR1c: *The language should offer support for capturing indicators of all the four types presented in Section 2*.

For people who want to learn the language, good documentation is possibly the single most important aid. Therefore:

> LR1d: *The language should be well documented*.

The need to purchase an editor for editing risk models could significantly hamper the willingness of potential users to adopt the WISER framework. Hence, it is important to ensure that this will not be necessary. We therefore include the following requirement:

> LR1e: *The language should offer a freely available editor*.

As outlined in Section 2 and further explained in Section 5, an important purpose of the risk model developed in Step 1 of the method is to serve as an aid to define the assessment algorithm in step 2. We therefore include the following requirement:

LR1f: *The language should have a semantics and structure that offer support for defining an algorithm for risk assessment*.

3.1.2 CORAS fulfilment of the requirements

CORAS was selected as the language for establishing and documenting the risk picture. Notice that although the use of CORAS was foreseen already when writing the proposal for the WISER project, its exact role was not determined; this evolved as the design of the overall WISER framework progressed. In Table 2, we show the evaluations that lead to the decision to use CORAS for Step 1 of the method outlined in Figure 2 and presented in more detail in Section 5.

Requirement	Evaluation of CORAS
LR1a	The graphical CORAS language has been developed specifically to be easily understandable by stakeholders with different backgrounds. Its comprehensibility has been tested empirically [9].
LR1b	The CORAS language has separate constructs for all the concepts listed. Notice, however, that in CORAS threat diagrams, a risk is captured implicitly by an incident together with a relation to an asset.
LR1c	The original CORAS language does not offer such support, although it is possible to use coloured notes to represent indicators. However, an extension of the CORAS tool has been developed with specific support for indicators. We are not aware of any other risk modelling that offer similar support for indicators as well as the risk concepts listed in LR1b.
LR1d	The CORAS language is extensively documented in the CORAS book [15], as well as a number of papers [1], [3], [9], [22].
LR1e	The CORAS editor is freely available from http://coras.sourceforge.net/downloads.html (accessed 13/5-2016).
LR1f	CORAS diagrams are directed acyclic graphs with a semantics and calculus to support likelihood assessment. This provides good support for definition of assessment algorithms. The guidelines in sections 9 and 8 show how to exploit the structure of the CORAS diagrams to develop assessment algorithms

Table 2. Evaluation of CORAS fulfilment of requirements to Step 1

3.2 Modelling language for machine-readable algorithms

3.2.1 Requirements

In addition to offering risk patterns, WISER allows risk models and assessment algorithms to be specifically developed for each client organization. If the language for defining assessment algorithms is easy to understand, it means that more of the relevant stakeholders can be involved in the development and validation of the algorithm. It could also mean that more organizations decide to take the extra effort to ensure that the algorithms are tailored to their needs. We therefore include the following requirement:

LR2a: *The language should be simple to understand for human stakeholders without requiring extensive training*.

There are a number of different ways to compute risk levels, from qualitative approaches to more or less advanced mathematical and statistical approaches using, for example, operations on exact numbers, intervals or distributions. Different stakeholders will have different preferences, depending on their background and training, as well as the needs of the organization they represent. Ideally, the language for defining assessment algorithms should support all such preferences. We therefore include the following requirement:

LR2b: *The language should have strong expressive power*.

Unlike the CORAS risk models, the assessment algorithms will be run by an execution engine, namely the Risk Assessment Engine (RAE), which is a part of the WISER framework. In order to make the implementation of the RAE practically feasible, we therefore include the following requirement:

LR2c: *The language should offer a clean interface for execution in the RAE.*

For the next two requirements, the motivation is identical to the motivation for LR1d and LR1e in Section 3.1.1:

LR2d: *The language should be well documented*.

LR2e: *The language should offer a freely available editor*.

To cater to different user preferences and service delivery models, we decided to support the use of two different languages for defining the assessment algorithms; one qualitative and one quantitative. For qualitative assessments, we chose DEXi, while for quantitative assessments we chose **R**. In the following, we show the evaluations of these two languages that lead to our choice.

3.2.2 DEXi fulfilment of the requirements for qualitative modelling

Table 3 shows the evaluation of DEXi against the requirements presented in Section 3.2.1.

Requirement	Evaluation of DEXi
LR2a	The simple hierarchical tree-structure of DEXi models, where concepts are decomposed into underlying or contributing concepts, is intuitive and easy to understand. Moreover, the clean separation between the tree structure and the utility functions means that understanding of a model can be obtained in a stepwise fashion, for example by first concentrating on each part of the structure and then on the utility function.
LR2b	As DEXi only allows simple qualitative assessments, this requirement is not fulfilled to a large degree for DEXi. Simplicity and expressive power is hard to combine, and DEXi was chosen for its simplicity, rather than its expressive power.
LR2c	DEXi comes with an execution engine as well as an API, thereby making implementing this part of the RAE very simple.
LR2d	A good and simple user manual is freely available at http://kt.ijs.si/MarkoBohanec/DEXi/html/DEXiDoc.htm (accessed 13/5-2016). A number of publications addressing several different uses of DEXi is also available from the same place.
LR2e	The DEXi editor is freely available from http://kt.ijs.si/MarkoBohanec/DEXi/html/DEXiNew502.htm (accessed 13/5-2016).

Table 3. Evaluation of DEXi fulfilment of requirements to Step 2

3.2.3 R fulfilment of the requirements for quantitative modelling

Table 3 shows the evaluation of **R** against the requirements presented in Section 3.2.1.

Requirement	Evaluation of R
LR2a	**R** is a quite complicated language, based only on textual representation of scripts (although graphical presentations of results are offered). Even if it is possible to write simple scripts in **R**, we would not claim that **R** is simple to understand. Therefore, this requirement is not fulfilled to a large degree. **R** was chosen for its expressive power, rather than its simplicity. Only advanced WISER users (for example consultants) are expected to write their own **R** scripts.

LR2b	**R** supports a number of advanced mathematical and statistical functions, for example interval arithmetic and operations on distributions. This requirement is therefore fulfilled to a very large degree.
LR2c	**R** comes with a software environment for executing **R** scripts, which can be exploited by the RAE.
LR2d	**R** is very well documented. Manuals can be downloaded from https://cran.r-project.org/manuals.html, while https://www.r-project.org/ provides links to sites where additional material, such as FAQs, books and other material can be found.
LR2e	The **R** environment, which includes an **R** script editor as well as an execution environment, can be freely downloaded from any of the sites listed on https://cran.r-project.org/mirrors.html (accessed 13/5-2016). Notice also that since **R** scripts are purely textual, they can be written using any text editor.

Table 4. Evaluation of **R** fulfilment of requirements to Step 2

4 Overview of selected risk modelling languages

In the following, we provide a brief overview of each of the three modelling languages selected for risk modelling in WISER.

4.1 CORAS

The CORAS language has been specifically developed to capture risk models in an intuitive way that can be easily understood by human actors. In WISER, CORAS is used to establish and document the understanding of the risk picture in the first step of the risk modelling process, as illustrated by Figure 7. Below we explain the CORAS language. Notice that the description is based on the one given in D3.1. We include it also here for the sake of completeness.

Figure 3 shows an example of a CORAS risk model.

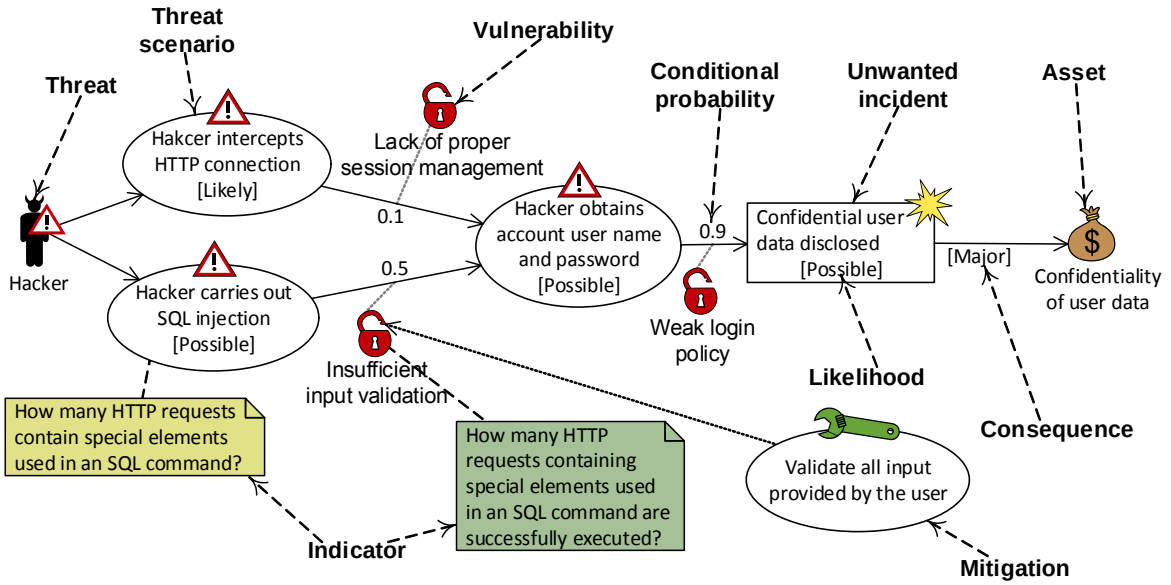


Figure 3. Example of a CORAS risk model

The dashed arrows in the figure are not part of the model and are only used to point out the various constructs in the CORAS language. As illustrated, a CORAS risk model is a directed acyclic graph where every node is of one of the following kinds.

- *Threat*: A potential cause of an unwanted incident.
- *Threat scenario*: A chain or series of events that is initiated by a threat and that may lead to an unwanted incident.
- *Unwanted incident*: An event that harms or reduces the value of an asset. Notice that we frequently use *incident* rather than *unwanted incident*.
- *Asset*: Something to which a party assigns value and hence for which the party requires protection. Notice that this means that the term asset is used in a wide sense; it can include any tangible or intangible entity of value for the party in question. In the context of cyber security, some typical examples are confidentiality, availability and integrity of information, as well as the reputation of the party.
- *Mitigation*: An appropriate measure to reduce risk level. Notice that we often use the term *treatment* with the same meaning.

Risks correspond to pairs of unwanted incidents and assets. If an unwanted incident harms exactly one asset, as illustrated in Figure 3, then the unwanted incident represents a single risk. If an unwanted incident harms two assets, then the unwanted incident represents two risks, etc. Vulnerabilities are also represented in a CORAS risk model. Before explaining what vulnerabilities are, we consider the three kinds of relations in a CORAS risk model. Notice that the visual representation of the three different kinds of relations is the same; they are all shown as an arrow with an open arrowhead.

- *Initiates relation*: A relation that goes from a threat *A* to a threat scenario or an unwanted incident *B*, meaning that *A* initiates *B*.
- *Leads to relation*: A relation that goes from a threat scenario or an unwanted incident *A* to a threat scenario or an unwanted incident *B*, meaning that *A* leads to *B*.
- *Impacts relation*: A relation that goes from an unwanted incident *A* to an asset *B*, meaning that *A* impacts *B* with some consequence.
- *Vulnerability*: A weakness, flaw or deficiency that opens for *A* leading to *B*. Vulnerabilities are modelled as open locks, and are attached on the *initiates* relations or the *leads-to* relations.

To support risk estimation, the CORAS language uses the following three measures.

- *Likelihood values*: May be assigned to a threat scenario or an unwanted incident *A*, estimating the likelihood of *A* occurring. Likelihood is typically measured in terms of frequency of occurrence.
- *Conditional probabilities*: May be assigned to the *leads-to* relations going from *A* to *B*, estimating the probability that *B* occurs given that *A* has occurred.
- *Consequence values*: May be assigned on the *impacts* relations going from *A* to *B*, estimating the consequence that the occurrence of *A* has on *B*. Consequence is often measured in terms of money, although other measures may be used, depending on the asset in question. For example, if the asset is availability, then the consequence is often measured in terms of downtime.

What has been described to this point is part of the core CORAS language. The reader is referred to Lund et al. [15] for a further explanation of the CORAS approach and the various constructs in the CORAS language. However, in the context of WISER it is necessary to extend the CORAS language with additional constructs for indicators of the type described in Section 2.

Figure 3 illustrates two indicator types: network-layer monitoring and test results. The indicator "How many HTTP requests contain special elements used in an SQL command" is of type network-layer

monitoring and is assigned to the threat scenario "Hacker carries out SQL injection". The indicator "How many HTTP requests containing special elements used in an SQL command are successfully executed?" is of type test results and is assigned to the vulnerability "Insufficient input validation".

4.2 DEXi

DEXi [7] is a language for the development of qualitative multi-criteria decision models and the evaluation of options. Multi-criteria (also called multi-attribute) models are a class of models used in Decision Analysis that evaluate options according to several, possibly conflicting, goals or objectives. In this section, we provide a brief overview of the DEXi language, for a detailed description we refer to the DEXi User's Manual [2] and to D3.3 for details on the DEXi tool. The following presentation is to a large degree based on the DEXi User's Manual.

A multi-attribute model decomposes a decision problem into a tree structure where each node consists of sub-problems, which are smaller and less complex than the overall problem. DEXi models consists of:

- attributes, organized into a tree structure;
- utility functions

Attributes are organized hierarchically into a tree, which can have one or more root attributes. According to their position in the tree, the attributes are either:

- Basic attributes: terminal nodes ("leaves") of the tree;
- or aggregate attributes: intermediate nodes in the tree, which can be decomposed into one or more descendant attributes appearing one level below the "parent" aggregate attribute in the tree.

Basic attributes are the inputs of the multi-attribute model. The values of the basic attributes are the possible options which the decision maker can select. The set of values which an attribute can take is called the scale of the attribute. In qualitative models, attributes are qualitative: scales are characterized by a finite set of symbolic values, typically consisting of words rather than numbers; this is different from "quantitative" decision models, which are characterized by continuous numerical scales or preferences. Examples of qualitative scales are:

- no, yes
- low, medium, high (e.g., for a "Quality" attribute)
- high, medium, low (e.g., for a "Price" attribute)
- unacceptable, acceptable, good, excellent

Aggregate attributes represent **option evaluations**: such evaluations are carried out by means of decision rules called *utility functions*. For a given aggregate attribute A (with scale SA) having n descendants D1, …, Dn (with scales SD1, …, SDn), the utility function of A maps each combination of values of the descendants into a value for A. The utility function of A is therefore a function from the product set of SD1, …, SDn into SA:

$$f: SD1 \times SD2 \times \ldots \times SDn \rightarrow SA.$$

In qualitative models, utility functions are tables of rules rather than numerical formulae (such as weighted sums), as would be the case for quantitative models. Figure 4 shows a simple example of an aggregate attribute "CAR" with two descendants "PRICE" and "TECH.CHAR" having scales of the attributes as follows:

PRICE: high, medium, low

TECH.CHAR: bad, acceptable, good, excellent

CAR: unacc, acc, good, exc

Figure 5 shows an example of utility function for the tree in Figure 4: a table associating a value for the aggregate node (CAR) to each combination of values for the basic nodes PRICE, TECH.CHAR [2].

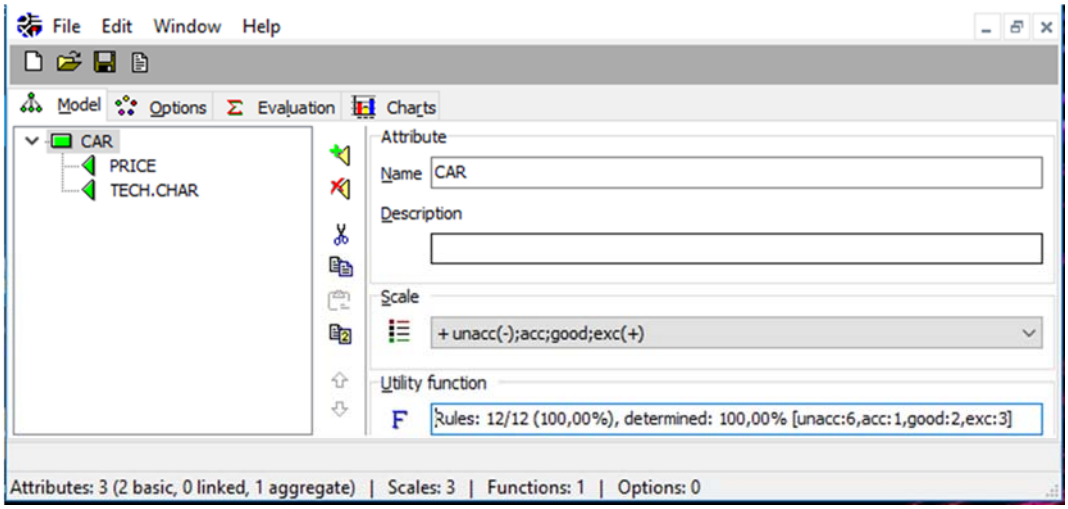


Figure 4. DEXi tree for the aggregate attribute “CAR” with two descendants “PRICE”and “TECH.CHAR”.

	PRICE	TECH.CHAR	CAR
1	high	bad	unacc
2	high	acc	unacc
3	high	good	unacc
4	high	exc	unacc
5	medium	bad	unacc
6	medium	acc	acc
7	medium	good	good
8	medium	exc	exc
9	low	bad	unacc
10	low	acc	good
11	low	good	exc
12	low	exc	exc

Figure 5. Definition of utility function in DEXi.

In DEXi models, an option is a set of values such that one value is associated to each attribute in the tree. The set of values is partitioned as follows:

- option descriptions: a vector of values assigned to each basic attribute;
- intermediate evaluation results: values assigned to all aggregate attributes other than the roots of the tree;
- overall evaluation results: the values assigned to the root(s) of the tree.

In the simple example of Figure 5, an option is e.g. the following set of values: high (for basic attribute PRICE), acc (for basic attribute PRICE), unacc (for aggregate root attribute CAR); in this case, there

are no intermediate evaluation results. For a given multi-attribute model, an option is evaluated with a bottom-up aggregation procedure:

- the option assigns values to each basic attribute (highest level of the tree);
- all aggregate attributes are then considered, which have exclusively basic attributes as descendants: values are calculated for a given aggregate attribute of said type, using its utility function and the values assigned to its descendants;
- the process is iterated for the aggregate attribute at increasingly lower levels of the tree;
- the overall evaluation of the option is obtained as the value of the root attribute(s) of the model.

On this basis, the decision maker can compare and rank the options, and possibly identify and select the best one. DEXi models allow for several additional features, such as:

- usage of numerical weights (to support the definition of utility functions);
- allowing undefined values for one or more basic attributes;
- linked attributes.

4.3 R

R [20] [23] is an environment for statistical computing and graphics. It is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It runs on a wide variety of platforms, including Linux, UNIX, Windows and MacOS.

R provides an integrated suite of software facilities for interactive data manipulation, statistical modelling and graphical display. It includes

- an effective data handling and storage facility;
- a suite of operators for calculations on arrays, in particular matrices;
- a large, coherent, integrated collection of intermediate tools for data analysis and statistical modelling;
- graphical facilities for data analysis and display either on-screen or on hardcopy;
- a complete object-oriented programming language which includes conditionals, loops, user-defined recursive functions, input and output facilities.

R was first created by Ross Ihaka and Robert Gentleman at the University of Auckland in 1993, and since then the project leadership has grown to include more than 20 leading statisticians and computer scientists from around the world. The **R** language allows users to develop additional functionality by defining new collections of functions, datasets and documentation called packages. Thousands of users from academia and industry have contributed packages through the years [4], implementing a vast set of data manipulation tools, statistical models and charts: this includes not only standard methods, but also advanced state-of-the-art algorithms developed by researchers in statistics and predictive modelling. As a result, there is a vibrant and growing community of **R** users on-line [18], with a rich set of learning and reference resources for both the beginners and expert **R** users. We refer to the **R** documentation [19] for a thorough description and to D3.3 for details on the **R** tool.

5 Target modelling

Target modelling is an activity carried out to document and obtain an understanding of the parts and aspects of the system that are the subject of the risk assessment, and thus the parts and aspects of the system that are potentially exposed to cyber risks.

In WISER, target modelling is carried out in two different ways for different purposes. Either, (1) the target is modelled simply by choosing, from a pre-defined list, the set of target characteristics that best describes the target (as part of setting up the WISER framework). Or (2) the target is modelled in a

more free form that allows the modeller to capture the parts and aspects she considers relevant, typically in terms of a graphical diagram.

The purpose of (1) is to provide, in a quick and easy manner, the WISER framework with information that describes the target of analysis, which in turn helps the WISER framework selecting a set of relevant risk patterns. Hence, this lightweight approach is aimed at those who wish to employ the ready-made WISER risk patterns. The purpose of (2) is to describe a specific target in more detail, in particular the parts and aspects that are potentially exposed to cyber risks. This approach is suitable for those who wish to develop a new risk model for a specific target.

In Section 5.1, we first provide an overview of general target characteristics used in WISER to identify relevant risk patterns (1), and then, in Section 5.2, we provide some simple general guidelines for target modelling (2).

5.1 General target characteristics for identification of relevant risk patterns

As described in D2.3, the model selection view & logic is used to select relevant risk models for the client when setting up the WISER framework. The framework proposes relevant risk models by matching the relevant parts of the business configuration information inserted by the client with the relevance criteria associated with each existing risk pattern. The relevance criteria consist of general target characteristics and affected security assets (integrity, confidentiality, and availability). In the following, we provide a more detailed description of the general target characteristics.

The target characteristics for each risk pattern currently provided by WISER are grouped in three categories: *computer systems organization*, *networks*, and *information systems* (see D3.1). These categories are based on the ACM Computing Classification System, which serves as the de facto standard classification system for the computing field [28]. The classification relies on a semantic vocabulary as the single source of categories and concepts that reflect the state of the art of the computing discipline and is receptive to structural change as it evolves in the future [28]. As new patterns are developed in WISER other categories in this classification system may be of relevance and added to the WISER framework.

The classification system is organized as a tree-structure, which currently consists of 14 main categories. Each of these three categories consists of a large number of sub-categories. The whole tree-structure of category *computer systems organization* consists of 60 sub-categories (including sub-sub-categories etc.), *networks* consists of 117 sub-categories, and *information systems* consists of 323 sub-categories. Current risk patterns in WISER address three out of the 14 main categories, and in total four sub-categories. The following points list all main categories currently provided by the ACM Computing Classification System [28], as well as the (sub)-categories addressed by current patterns in WISER (in italic).

- General and reference
- Hardware
- *Computer systems organization*
    - *Client-server architecture*
    - *Distributed architectures*
- *Networks*
    - *Application layer protocols*
- Software and its engineering
- Theory of computation
- Mathematics of computing
- *Information systems*
    - *Web-applications*

- Security and privacy
- Human-centered computing
- Computing methodologies
- Applied computing
- Social and professional topics
- People, technologies and companies

The reason to why the abovementioned categories (including sub-categories) represented in italic font are addressed by current patterns in WISER is that these categories were the most relevant for the risk patterns, which in turn represent the most common attack scenarios. As the WISER framework evolves more categories will be relevant and available for the client.

Thus, when setting up the WISER framework, the client selects one or more categories to describe the target of analysis. Based on that (and the relevant security assets) the WISER framework suggests relevant risk patterns.

5.2 Target modelling for tailored risk models

This section provides general guidelines for modelling the target of analysis, which is used as a basis for risk modelling. As mentioned above, the purpose of creating a graphical model to describe the target of analysis is to obtain a more detailed description of the parts and aspects that are potentially exposed to cyber risks. However, in contrast to the first approach to target modelling in WISER, this model is not to be used as input to the technical WISER tools, but is used only to support clients or consultants when creating dedicated risk models for specific targets. Hence, the relation to the WISER framework is purely methodological. This allows us to provide high-level guidelines of general relevance, rather than forcing modellers to employ any particular modelling language. Such flexibility is very important, because different actors will have very different preferences with respect to the modelling approach, depending on their background and competence, as well as the focus and desired level of abstraction for the analysis.

Target modelling is carried out during context establishment, which consists of a number of additional activities. For example, defining goals and objectives of the risk assessment, defining the scope of the assessment, and defining the focus of the assessment. However, in this section we will only consider guidelines for target modelling, assuming the aforementioned activities have already been carried out. The guidelines provided in this section are mainly based on guidelines provided in [21]. The reader is referred to [21] for a thorough explanation of all activities during context establishment and cyber-risk assessment in general.

What distinguishes target modelling in the context of cyber-risk assessment from the general case is that we need to understand and document how the target of analysis makes use of and interacts with cyberspace. By cyberspace, we mean a collection of interconnected computerized networks, including services, computer systems, embedded processors, and controllers, as well as information in storage or in transit. This gives a basis for understanding how and where cyber-threats arise, as well as which assets are relevant to focus on.

As part of the description of the target of analysis, we therefore include the interface to and interaction with the cyberspace and other relevant parts of the environment. Understanding and documenting the interface to the cyberspace is important for cyber-risk management in general and for identification of cyber-risks in particular. The cyber-threats arise in or via the cyberspace, and the interface between the cyberspace and the target of analysis overlaps with the attack surface. The attack surface is all of the different points where an attacker or other threat source could get into the cyber-system, and where information or data can get out.

Typical assets of concern in the setting of cyber-risk assessment are the confidentiality, integrity, and availability of information and information infrastructures, including software, services, and networks.

Moreover, it is important to think about the level of abstraction, at which the risk assessment is carried out, as well as the knowledge and experience of the target group. For example, a high-level view of the target of analysis may be appropriate if managers at business level participate in the assessment, while a more detailed low-level view may be more suitable if technicians and developers at source-code level carry out an assessment focusing on technical details. In the first case, an informal notation that can be easily understood without any specific training may be sufficient for capturing the target, while in the second case a more formal notation, such as UML state machines, class diagrams or sequence diagrams may be more useful. This also depends on the preferences and the experience of the involved stakeholders.

In the following, we provide a small example. Assume we are interested in identifying risks at a high level of abstraction of an online electronics store. This means that we choose to carry out the risk assessment at a high level of abstraction and therefore choose to model the target at a high level of abstraction. We choose in this example to employ an informal notation for capturing the target.

Let us further assume that this online store consists of two main parts: end-user (the customers), and the electronics store. The end-user may access the online store using the web-browsers of a computer or a smartphone/tablet. To communicate with the online store, the web browsers uses the https protocol over the Internet. The online store is hosted by a web server, which is located at the premises of the owners of the online store. In order to save all information about customers, their credit card information, and other information such as orders placed by the customers, the online store uses a dedicated customer database. Figure 6 shows a high-level model of the target.

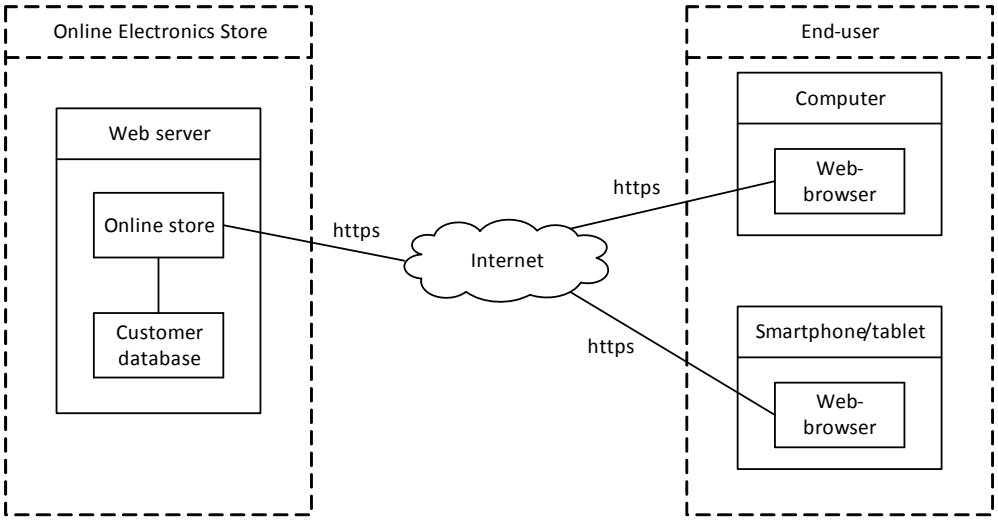


Figure 6. High-level overview of the target (online electronics store)

The infrastructure documented in Figure 6 is part of a larger cyberspace involving the Internet. The interface between the target of analysis and the cyberspace therefore consists of the following:

- The connection point between web-browser on the computer and the Internet.
- The connection point between the web-browser on the smartphone/tablet and the Internet.
- The connection point between the online store on the web server and the Internet.

Attacks can be launched remotely targeting each of these connection points; hence, they are all included in the attack surface.

Based on the above, the client may create risk models fitting the context captured by the simple target model shown above. Assume we are interested in identifying risks that may have an impact on the confidentiality of end-user data stored in the customer database. Then, one risk model that may fit the above target is the risk-model example illustrated in Figure 3.

6 Overall method for cyber risk modelling

In this section, we explain the overall method for risk modelling in WISER. The risk modelling described here will typically be a part of a wider risk management process, such as ISO 31000 [1], and can be "plugged into" any such process. In this report, we focus only on the methodological aspects that are special in the context of WISER, which are the following:

- For risk level assessment, the goal is not to perform a snapshot assessment of one particular point or period in time, but rather to develop algorithms for continuous automated assessment. Such algorithms are either qualitative (and expressed using DEXi) or quantitative (and expressed using **R**).
- Identification of threats, vulnerabilities, threat scenarios, incidents and risks is done using CORAS diagrams.
- For the elements of a risk model described above, we also identify relevant indicators that can be provided by the WISER framework and serve as useful input for the assessment algorithms.

Figure 7 shows the overall method for cyber risk modelling, considering these aspects.

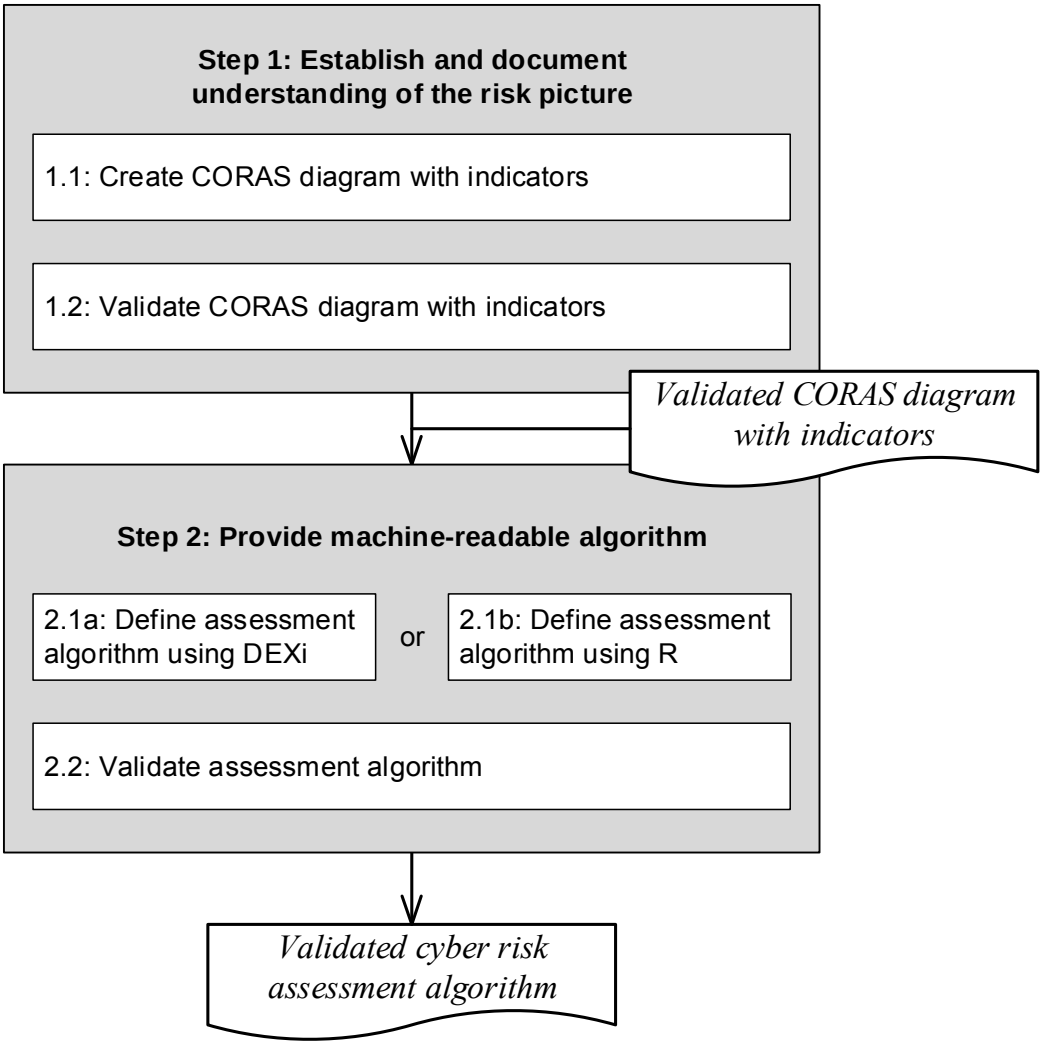


Figure 7. Overall method for cyber risk modelling

The outcome of the first step is a validated CORAS diagram with indicators. This diagram captures the relevant assets, risks, the ways in which these risks may materialize, and the relation between these elements and the available business configuration indicators, vulnerability test result indicators, network monitoring indicators and application monitoring indicators that can be employed to assess the risk and the involved threats, vulnerabilities and threat scenarios.

The outcome of the second step is a machine-readable algorithm for risk level assessment (and mitigation proposals) that can be automatically executed by the Risk Assessment Engine. The dynamic input for this assessment algorithm consists of the indicators identified in the first step.

In the following, we explain the steps of the overall method. Sections 7, 8 and 9 then provide specific modelling guidelines for each of the three modelling languages CORAS, DEXi and **R**.

6.1 Establish and document understanding of the risk picture

6.1.1 Create CORAS diagram with indicators

As illustrated by Figure 7, this step is the same irrespective of whether the aim is to develop a qualitative or a quantitative assessment algorithm. The reason is that the purpose of this particular step is not to assess risk levels or define an assessment algorithm, but to identify the potential chains of events that may lead to risks materializing. This includes identifying all the threats, vulnerabilities, threat scenarios and incidents involved in such chains. Moreover, we identify the indicators that can provide information about all risk elements that can serve as useful input for the assessment algorithm to be developed in Step 2.

6.1.2 Validate CORAS diagram with indicators

The CORAS diagram provided in Step 1.1 serves as the basis for developing the machine-readable algorithm in Step 2. Therefore, before moving on, it is essential to ensure that the CORAS diagram reflects, as far as possible, the actual reality with respect to potential threats, vulnerabilities, threat scenarios and risks.

Of course, as risk assessments concern what might happen in the future, there is no way we could ensure that a CORAS diagram (or any other form of risk model) is objectively correct and complete with respect to reality. Instead, what we aim for here is a convincing argument that the diagram reflects available knowledge and beliefs among qualified cybersecurity experts. Such an argument can be established, for example, by showing that the CORAS diagram faithfully captures information available from well-reputed standards, repositories, text books, research papers or similar sources; some examples include ISO 27001 [11], ISO 27005 [12], ISO 27032 [13], CAPEC [16] and OWASP [17]. If possible, the validation of the CORAS diagram should be carried out by a group of cybersecurity experts who, after relevant information sources has been identified and obtained, go through each part of the diagram in a systematic manner to identify elements that need to be added, removed, or otherwise improved. The validation terminates when no such elements are found.

6.2 Provide machine-readable algorithm

6.2.1 Define assessment algorithm

The CORAS diagram with indicators obtained from the previous step show how risks may materialize through chains of events initiated by threats exploiting vulnerabilities, as well as which indicators that can provide information about these elements that are useful for assessing the risk level. However, it does not define the details of how this assessment will be done. The purpose of this step is to define a machine-readable assessment algorithm that can be automatically executed to provide risk level assessments calculated from the indicators identified in the CORAS diagram. This means that the input to the algorithm consists of all the indicators included in the diagram. For each identified risk, the risk level will depend on all incoming paths consisting of threats, vulnerabilities and threat scenarios leading to the risk. Each indicator is attached to at least one of the elements of these paths. Hence, the structure of the CORAS diagram offers significant support for defining the assessment algorithm. In fact, CORAS comes with a calculus for reasoning about the likelihoods of threat scenarios and

incidents. In Section 9 we provide specific guidelines for how to exploit the structure of a CORAS diagram, as well as the CORAS calculus, when defining a qualitative assessment algorithm using DEXi. In Section 8 we provide similar guidelines for defining a quantitative algorithm using **R**.

6.2.2 Validate assessment algorithm

6.2.2.1 General considerations

The point regarding the impossibility of establishing objective correctness and completeness of a CORAS diagram discussed in Section 6.1.2 applies, of course, also for the corresponding assessment algorithm. Lenstra and Voss [14] (p. 392) states the point nicely when discussing the subjectivity of risk management: "The best one can aim for is consistency within the model, overall soundness of the model, and an on average high level of user acceptance and appreciation of the results."

The goal of the validation of the assessment algorithm is therefore to establish its consistency and overall soundness, in order to obtain user acceptance and confidence that the outputs from the algorithm provide useful information that reflects reality reasonably well. In our context, soundness means that if the input obtained from the indicators reflects reality, then the risk level assessments provided by the algorithm also reflect reality.

The CORAS calculus has been developed in order to support identification of inconsistencies in a CORAS diagram annotated with likelihood assessments. We have designed the guidelines presented in Section 8 and Section 9 with the aim of ensuring that consistency results from faithfully following the guidelines.

With respect to soundness of the algorithm, this should ideally be established by triangulation, which means that the results from the algorithm are compared to results obtained through other means. Depending on the data and resources available, there are different ways this can be done.

- If historical data are available for indicators as well as past incidents, then the assessment algorithm can be run using sets of indicator values that applied at specific points in the past. We can then compare the risk level assessments produced by the algorithm to the risks that actually materialized over a defined period, to see if there is a correlation. This approach has the obvious benefit of using fully realistic historical data. However, such data are not always available.
- If historical data of past incidents are not available, then the results from the assessment algorithm can be compared to risk level assessments done following other risk assessment methods, independently of the algorithm. Unfortunately, since risk assessment processes normally require quite a lot of resources, such a process may be quite costly.
- A thought experiment can also be employed for triangulation. This can be done by providing a group of experts, who should not know the assessment algorithm, the information contained in a set of realistic indicator values. We then ask them to provide their own risk level assessments based on that information. The expert assessments are then compared to the assessments provided by the algorithm to check the correlation.

Irrespective of which method of triangulation is used, a high correlation between the assessments produced by the algorithm and the alternative assessment gives reason to have good confidence in the algorithm.

6.2.2.2 Validation of algorithms developed in WISER

In WISER, we developed ten risk patterns (documented in D3.1), each with corresponding **R** and DEXi algorithms. We developed our own approach for validating these algorithms, which was tailored to the nature of these patterns and the specific circumstances of the WISER project, in particular the following:

- The patterns represent generic risk models, rather than addressing a specific target system. There was no concrete stakeholder (client) for the assessment. Instead, we assumed that the

stakeholder was a typical European SME. This affected the choice of empirical data sources, as well as the assessments made based on these sources.

- For each pattern, there is a quantitative algorithm (expressed in **R**) and a qualitative algorithm (expressed in DEXi), rather than one or the other. This means that it is possible to consider results from both during the validation.
- Finally, we had to ensure that the procedure could be carried out by members of the WISER team within the time frame and resources available.

A detailed description of the validation approach for the WISER risk patterns is given in Appendix V.

6.3 A comment on the flexibility of the method

As explained earlier, the method illustrated by Figure 7 assumes that CORAS will be used for establishing and documenting the risk picture in Step 1 of the method, and that a CORAS diagram will be used as the basis for defining an assessment algorithm in Step 2. The guidelines presented in the next sections are also based on this assumption, as they explain how to define DEXi or **R** algorithms from the structure of a CORAS diagram.

Notice, however, that the role of CORAS diagrams is purely methodological in WISER. The diagrams are only meant to be read and understood by human stakeholders, and are not used by the technical components of the framework except from the CORAS editor. This means that clients or consultants who want to create their own models are not forced to use CORAS if they have other preferences. They could use, for example attack trees, risk tables, or any other approach, or even go straight to Step 2 and define the algorithm without any preparatory modelling. Although we would hardly recommend the latter, it does not really matter which approach they follow, as long as they arrive at an appropriate assessment algorithm to be run by the Risk Assessment Engine using either DEXi or **R**. Of course, WISER cannot offer guidelines to support all possible approaches. We have chosen a method involving the use of CORAS because we believe this is well suited for the purpose.

7 Creating a CORAS diagram with indicators

In this section, we offer guidelines for creating CORAS diagrams. More specifically, we address the use of CORAS diagrams to identify and document threats, vulnerabilities, threat scenarios, incidents and assets, and to show how these risk model elements relate to each other. We also include mitigations, as triggering of mitigation proposals is supported by the WISER modelling approach.

Thorough guidelines for the use of CORAS for risk identification have been published earlier [3], [15]. Our intention here is not to repeat these in detail, but rather to provide a simple and short introduction. The CORAS language was originally developed to support risk assessment processes where brainstorming sessions involving stakeholders with different backgrounds play an important part in the risk identification. However, CORAS threat diagrams can of course also be employed to create risk models based on other sources, for example repositories such as CAPEC [16].

We do not make any assumptions about the context in which the CORAS diagrams are created. Our goal is to show how to arrive at a syntactically correct CORAS diagram and to offer some simple questions to support the identification of each type of risk model element. If the CORAS tool [6] is used, this will help ensuring syntactical correctness.

Notice that, unlike the rest of the risk model elements included here, the use of indicators is not a part of the core CORAS language as presented by M. Lund et al [15]. Indicators have been added in the WISER approach to exploit the information obtained by the WISER infrastructure for continuous monitoring of risk levels.

We present the addition of risk model element types one by one. This does not mean that this order has to be followed for the risk identification process; in fact, it is quite common to go back and forth between the steps. However, we strongly recommend that the assets are identified first, and documented in the CORAS diagram. The reason is that we need to know what the client wants to protect before we can determine which threats, threat scenarios, vulnerabilities and incidents are

relevant. In fact, assets are typically identified as part of the context establishment before the risk identification starts. More advice about the order of identification of risk model elements, as well as useful information sources for the identification steps, can be found in [21].

For each type of diagram element, we follow the same format. After giving a reminder on the definition of the model element, possibly supported by some additional comments, we present one or more guiding questions that may help during the identification. Then we list the syntactical constraints that must be respected. Finally, we present a diagram snapshot from the CORAS editor illustrating the stepwise development of a CORAS diagram as each new element type is added. For this, we employ the risk model illustrated in Figure 3.

7.1 Asset identification

An asset is something to which a party (typically the client) assigns value and wants to protect. Assets should be placed on the right-hand side of the diagram, as the harming of an asset due to an incident represents the last part of a chain resulting in the materialization of a risk.

7.1.1 Guiding questions

- What are the tangible or intangible entities of value for the party (client) that could potentially be harmed by a cyber-incident? Here we should pay special attention to confidentiality, integrity or availability of data or services that the party is responsible for or dependent upon.

7.1.2 Syntactical constraints

- An asset must be the target node of at least one *impacts* relation. It cannot be the target node of any other types of relation.
- An asset cannot be the source node of any relation.

7.1.3 Example diagram

Figure 8 shows the start of a CORAS diagram, where assets has been inserted on the right-hand side. Only a single asset is included, although there could have been more. If so, we would insert them above or below the one already there.

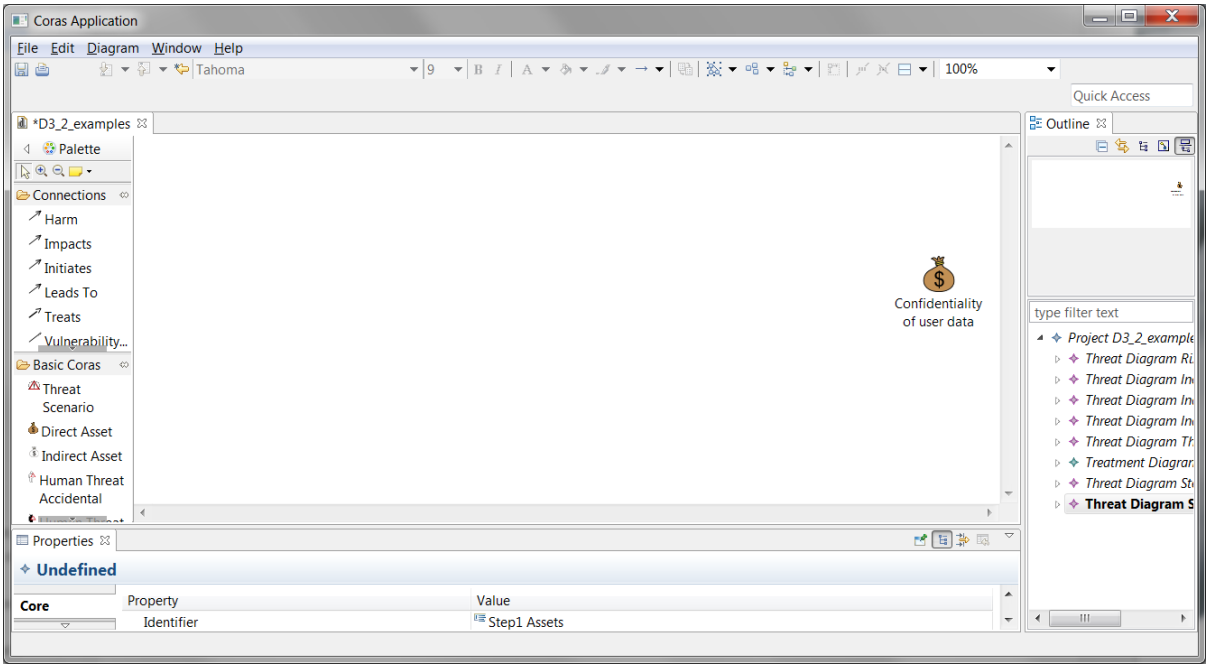


Figure 8. Adding assets to a CORAS diagram

7.2 Threat identification

A threat is a potential cause of an incident. In the context of cyber risk, we are often concerned about malicious human actors who deliberately launches attacks in order to harm our assets, and this is what we focus on here. However, human threats can also be non-malicious. For example, a thoughtless employee may publish confidential information on a website without wanting to cause any harm. Moreover, non-human threats, such as a power failure, should also be taken into account.

Threats should be placed on the left-hand side of the diagram, as they represent the initial cause of the chain leading to an asset being harmed.

7.2.1 Guiding questions

- Which malicious actors could want to perform a cyber-attack? Here we should consider all possible motives and intentions, including financial gain, revenge or grudges, political or religious agendas, espionage, or simply fun and a desire to prove one's ability.
- Which non-malicious actors could potentially initiate cyber-incidents through, for example, neglect or lack of competence?

7.2.2 Syntactical constraints

- A threat must be the source node of at least one *initiates* relation. It cannot be the source node of any other types of relation.
- A threat cannot be the target node of any relation.

7.2.3 Example diagram

Figure 9 illustrates the addition of threats on the left-hand side of the diagram. Again, the example includes a single threat, although more could have been added above or below.

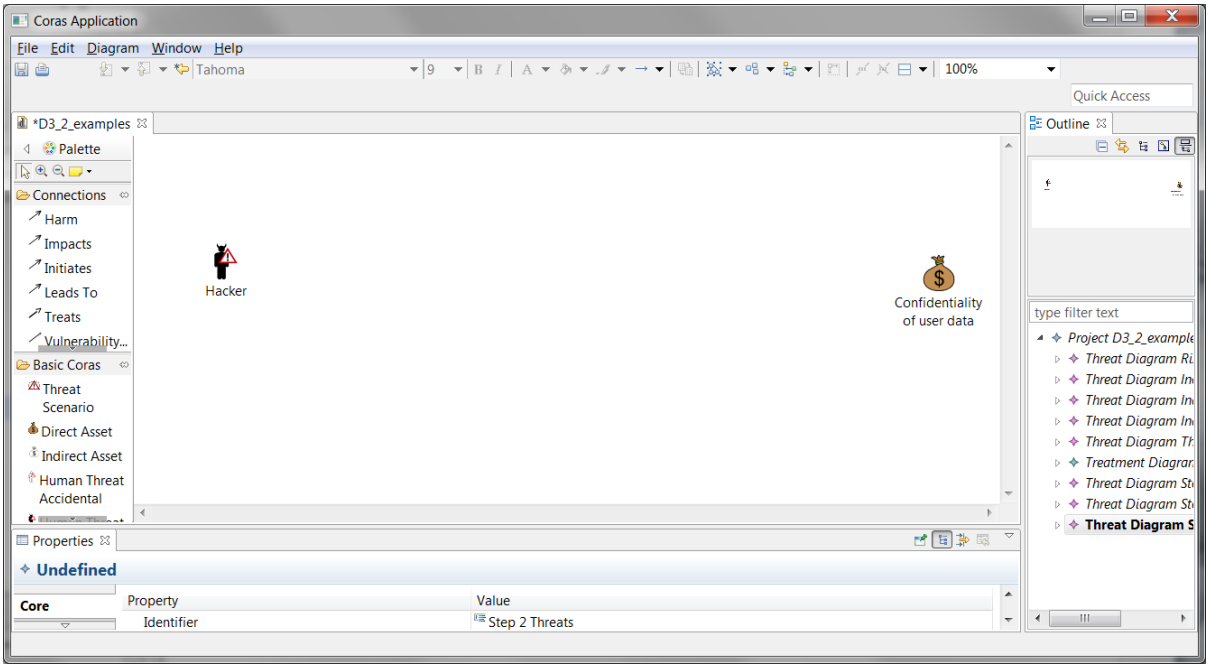


Figure 9. Adding threats to a CORAS diagram

7.3 Threat scenario identification

A threat scenario is a chain or series of events that is initiated by a threat and that may lead to an incident. Although defined as a chain or series, a threat scenario is represented by a single node (see Figure 3). It is up to the modeller to decide the level of abstraction for each threat scenario. Hence,

any chain or series of two or more events can be represented in CORAS either as a single threat scenario, or as a chain of two or more threat scenarios.

A threat scenario should be placed so that any incoming *leads-to* relations come from the left-hand side and any outgoing *leads-to* relations go to the right-hand side.

7.3.1 Guiding questions

- What types of attack can a threat initiate?
- Where are the interfaces between the target system and cyberspace, and how can attacks be launched through these interfaces?

7.3.2 Syntactical constraints

- A threat scenario must be the source node of at least one *leads-to* relation. It cannot be the source node of any other types of relation.
- A threat scenario must be the target node of at least one *initiates* relation or *leads-to* relation. It cannot be the target node of any other types of relation.

7.3.3 Example diagram

Figure 10 shows the addition of threat scenarios in the diagram. *Initiates* relations have also been added from the threat, as well as *leads-to* relations from one threat scenario to another. Both relation types are represented by an arrow with an open arrowhead.

Notice that the square brackets are meant for likelihood assessments. As we are only concerned with identification of risk model elements and their relations here, these have been left empty.

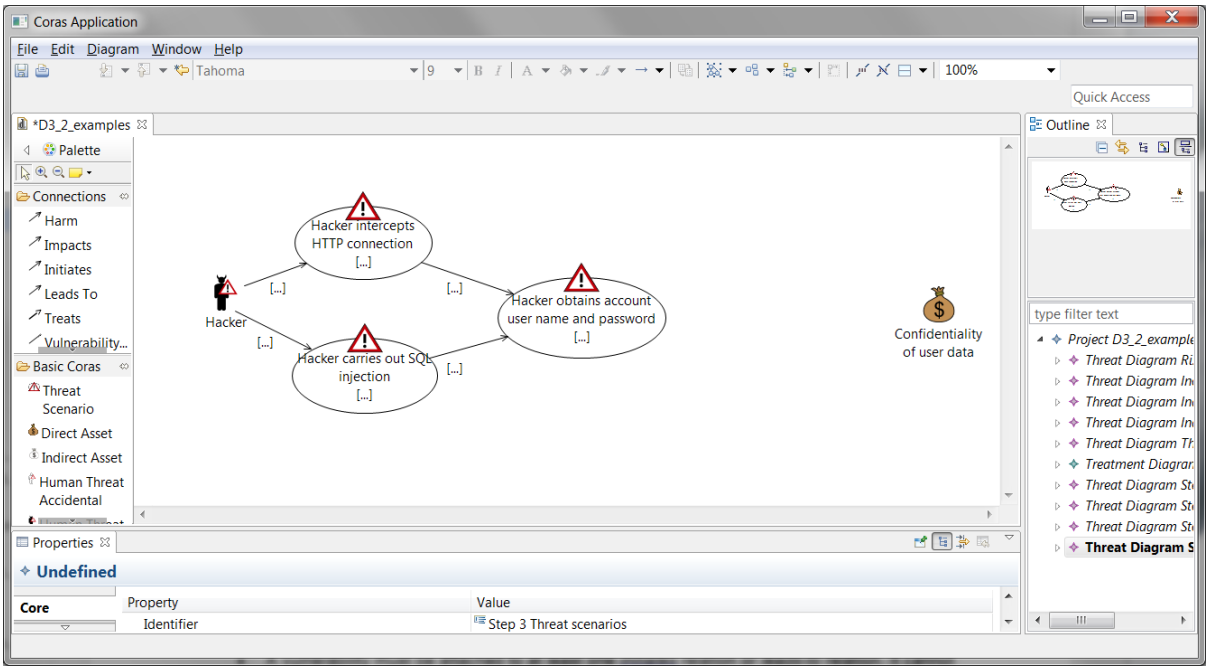


Figure 10. Adding threat scenarios

7.4 Vulnerability identification

A vulnerability is a weakness, flaw or deficiency that opens for *A* leading to *B*, where *A* and *B* can be threat scenarios or incidents (and *A* can also be a threat). Vulnerabilities can be attached to *initiates* relations and *leads-to* relations.

7.4.1 Guiding questions

- What makes it possible for an attack to succeed?
- Where are the weaknesses in our defence mechanisms?

7.4.2 *Syntactical constraints*

- A vulnerability must be attached to at least one *initiates* relation or *leads-to* relation. It cannot be attached to any other risk model element.

7.4.3 *Example diagram*

Figure 11 shows the addition of vulnerabilities in the diagram. In this example, vulnerabilities have only been attached to *leads-to* relations from threat scenarios, but they could also have been attached to *initiates* relations from a threat.

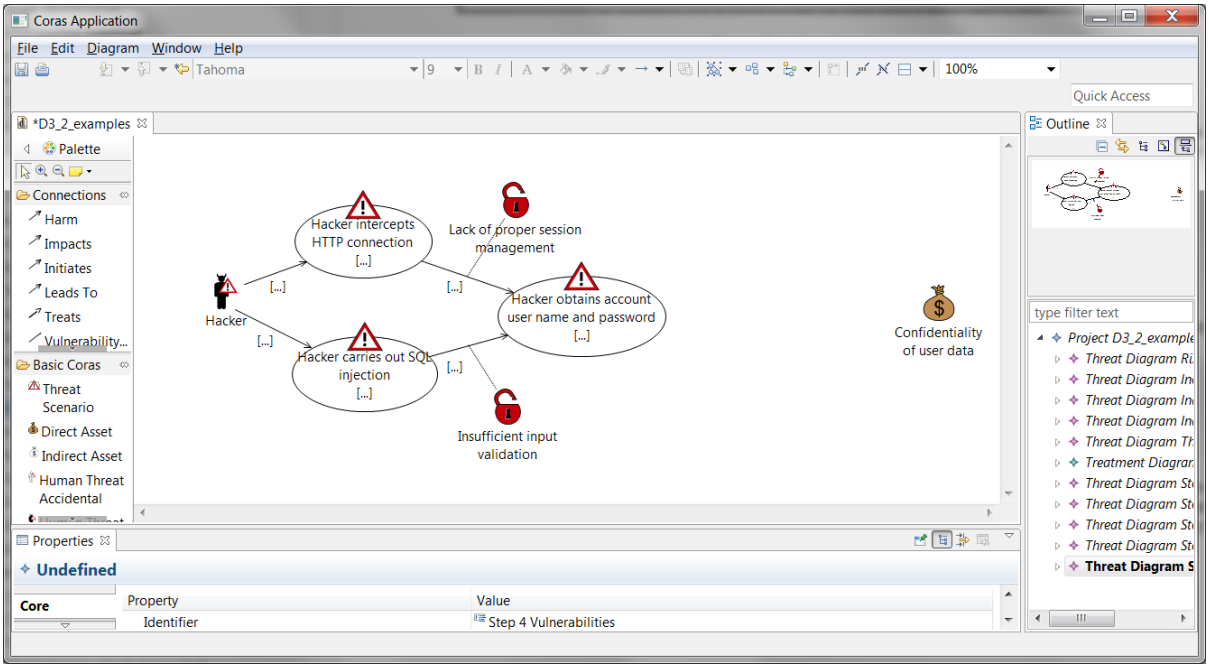


Figure 11. Adding vulnerabilities

7.5 Incident identification

An (unwanted) incident is an event that harms or reduces the value of an asset. Notice that this is an essential difference between a threat scenario and an incident. By definition, an incident *always* impacts at least one asset, whereas a threat scenario by itself *never* impacts an asset, even if it may lead to an incident. Incidents should be placed to the left of the assets.

7.5.1 *Guiding questions*

- What incidents could directly harm the identified assets?
- What incidents could result from a successful attack?

7.5.2 *Syntactical constraints*

- An incident must be the source node of at least one *impacts* relation.
- An incident must be the target node of at least one *initiates* relation or *leads-to* relation. It cannot be the target node of any other types of relation.

7.5.3 *Example diagram*

Figure 12 shows the addition of incidents in the diagram. In this particular example, there is only a single incident. We have also added an incoming *leads-to* relation to the incident, as well as an

impacts relation from the incident to the asset. This means that all threats, threat scenarios, incidents and assets are now properly connected to other elements of the model.

In addition to connecting the incident to the rest of the model, we have also added another vulnerability on the incoming *leads-to* relation to the incident. This means that all *leads-to* relations have an attached vulnerability. Although this is not a requirement, it may be a good idea to support the identification of indicators in the next step.

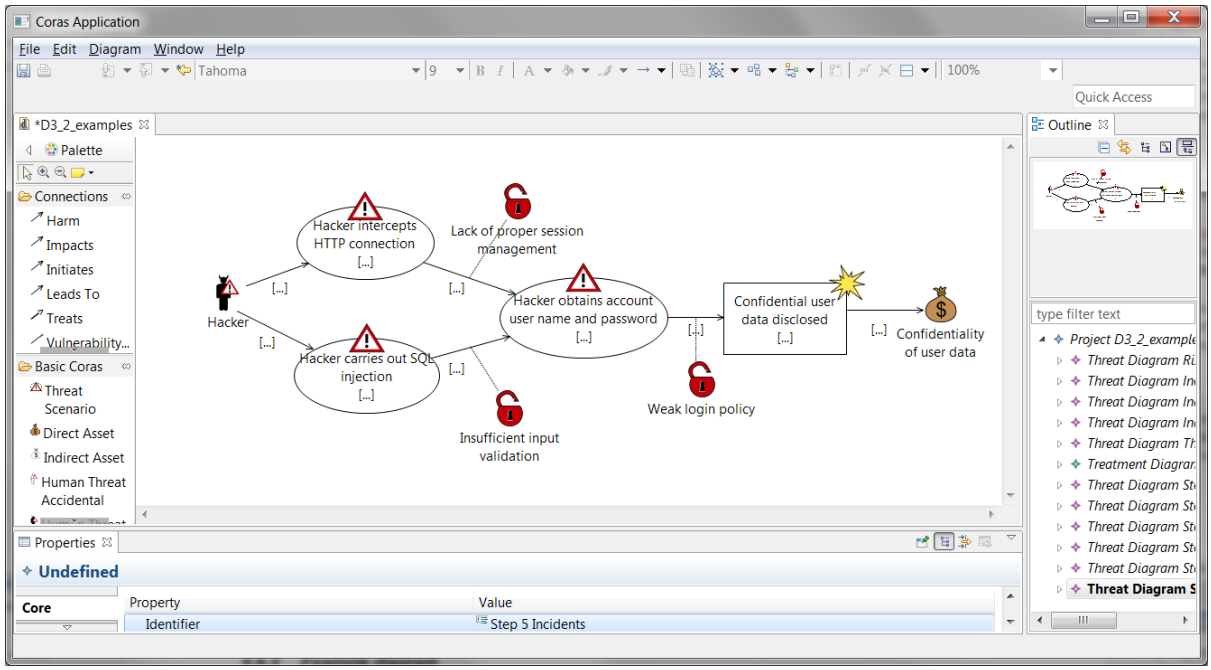


Figure 12. Adding incidents

7.6 Indicator identification

Recall from Section 2 that an indicator is a piece of information that can provide useful input for risk level assessment that can potentially be obtained from the WISER infrastructure. Indicators can relate to any risk model element.

7.6.1 Guiding questions

- What observable events at the network layer could give useful information about the likelihood/frequency of attacks? (Network-layer indicators.) This question should be asked for each identified threat scenario and incident.
- What observable events at the application layer could give useful information about the likelihood/frequency of successful or unsuccessful attacks? (Application-layer indicators.) This question should be asked for each identified threat scenario and incident.
- What information can we get from vulnerability scanners or security tests? (Test result indicators). This question should be asked for each identified vulnerability.
- What do we otherwise know about the threats, vulnerabilities, threat scenarios, incidents or assets that could help us assess the level of cyber-risk? (Business configuration indicators.) These questions should be asked for each element of the risk model.

7.6.2 Syntactical constraints

- An indicator can be attached to any risk model element except from an indicator. It must be attached to at least one risk model element.

7.6.3 Example diagram

Figure 13 shows the addition of indicators in the diagram. Two indicators have been added. The first is attached to a threat scenario, meaning that it will be used as input for assessing the likelihood of the threat scenario. The other is attached to a vulnerability attached to a *leads-to* relation, meaning that it will be used as input for assessing the conditional likelihood of the relation.

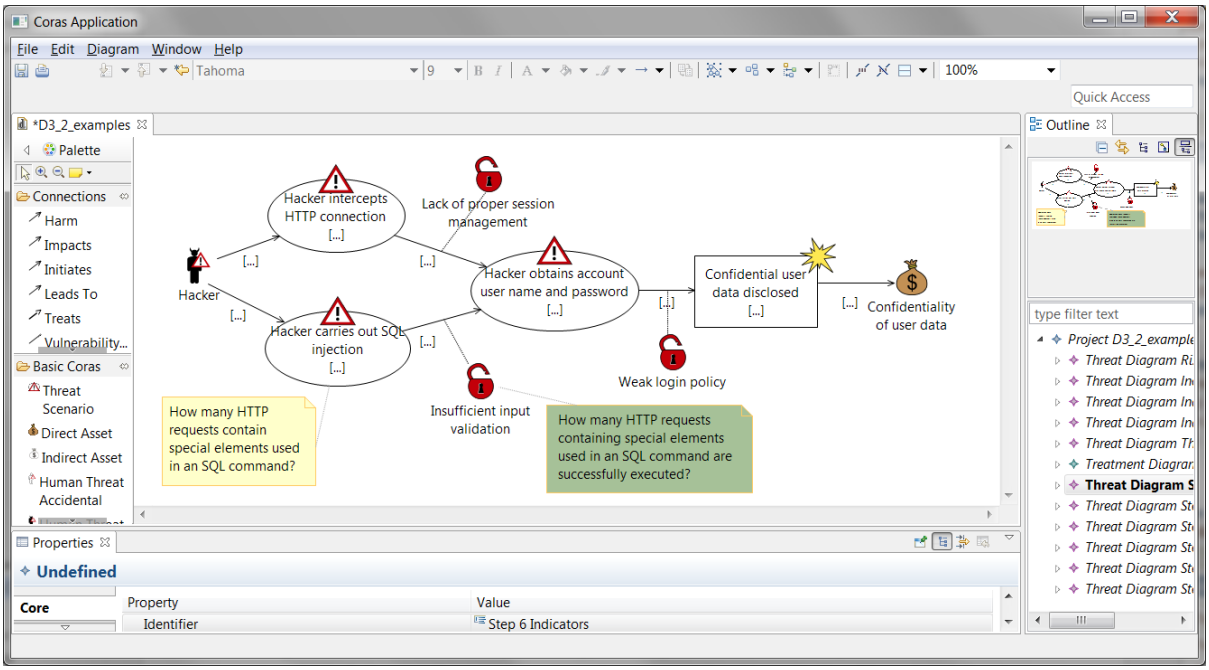


Figure 13. Adding indicators

7.7 Mitigation identification

A mitigation (also called a treatment) is a measure taken to reduce the risk level. Such measures will often aim to reduce the likelihood of incidents, typically by reducing or removing vulnerabilities. However, other alternatives are also available, and it is common to consider four main categories of mitigations: risk reduction, risk retention, risk avoidance and risk sharing. Cyber-insurance is an example of the latter, where part of the risk is transferred to a third party.

7.7.1 Guiding questions

- How can we reduce the vulnerabilities?
- How can we reduce the consequence of the incidents?
- How can we reduce the likelihood that the threats will initiate an attack?
- Are there other ways to reduce the likelihood of threat scenarios and incidents?

7.7.2 Syntactical constraints

- A mitigation can be added to a threat, a vulnerability, a threat scenario, an incident, or an asset.

7.7.3 Example diagram

Figure 14 shows the addition of a mitigation in the diagram. This mitigation addresses the vulnerability concerning input validation. Implementing the mitigation can therefore be expected to reduce the conditional likelihood of the *leads-to* relation to which the vulnerability is attached, although likelihood values are not shown in the diagram.

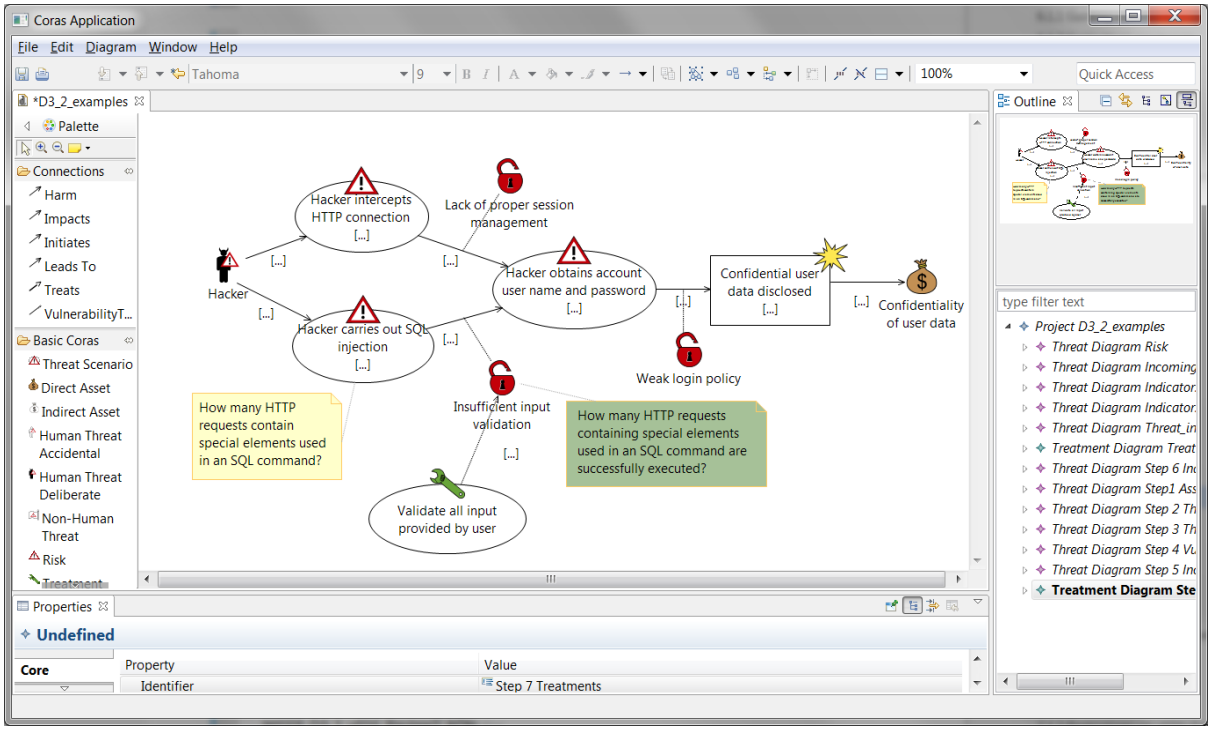


Figure 14. Adding mitigations

8 Defining quantitative assessment algorithms using R from CORAS diagrams

In this section, we provide guidelines for defining quantitative assessment algorithms based on CORAS diagrams. We recall that, in the WISER framework, a *risk* corresponds to a pair consisting of an *unwanted incident* and a *security asset impacted by the incident* (see Section 5 in Deliverable D3.1). For example, the unwanted incident might be a DDoS attack to a web server and the asset might be the availability of the applications hosted on the server.

To develop quantitative assessment algorithms we adopted an *actuarial approach*, where the frequency of occurrence of unwanted events and the severity (consequence) of the impacts are separately modelled through the probabilistic framework of Bayesian Networks (BNs) [8]. Due to the scarcity of publicly available data on economic losses associated to cyber events, we adopted a '*scenario analysis*' approach [26] to model the severity (see Section 8.2 for more details).

The BNs used are implemented with HydeNet [5]: this is an **R** package providing a powerful interface to construct BNs and perform inference. The package also handles hybrid BNs, that is networks where the random variables are not bound to be discrete or (conditionally) Gaussian; the underlying calculations are performed by MCMC (Markov-Chain-Monte-Carlo) using the '*rjags*' package; decision and influence networks are also supported, allowing to perform econometric analyses (maximum expected utility under competing policies, value of information). Appendix I provides a very concise introduction to the usage of BNs as tools to assess, model and measure risk.

These are the main steps to construct a BN model from a CORAS model:

1. construct a network skeleton (i.e. without probability distributions on nodes) containing CORAS graph and such that each *'leads-to'* relation and each *'impacts'* relation corresponds to one node;
2. model the probability distribution for the *frequency* of unwanted incidents (expressed as *number of events per year*);

3. model the probability distribution for the *severity* of the impacts on the affected assets (expressed as *Euro per incident*);
4. model probability distribution for the *annual aggregate loss* (expressed as *Euro per year*).

In the following, we first explain the reasoning behind the modelling of frequency and severity of unwanted incidents, before presenting a worked example.

8.1 Modelling the frequency of an unwanted incident

As explained in Section 4.1, a CORAS model captures unwanted incidents, the impacted assets and related variables affecting the risk of the unwanted incident occurring. In our approach, the frequency of unwanted incidents is not quantified directly. Instead, it is calculated following the logic of the CORAS models. Frequencies are assigned, on the ground of expert judgement or historical data (when available), to the initiating threat scenarios of the CORAS model, that is the threat scenarios which are only targets of an *initiates* relation; then, frequencies are propagated through the *leads-to* relations that link the initiating threat scenarios to the unwanted event. This will be made clear in the example below, where probability distributions are assigned to the threat nodes S1 and S2 in Figure 15, whereas the probability distributions of the threat node S3 and of the unwanted incident U1 are calculated depending on S1 and S2 (and other intervening indicators and variables).

8.2 Modelling the severity of the losses generated by an unwanted incident

The main problem in identifying suitable parametric families of probability distributions to model the severity (consequence) of the losses generated by cyber risk events is that there is very little historical data regarding such losses, which prevents using standard fitting algorithms such as maximum likelihood.

The following distribution families are currently used in actuarial practice to model losses: *lognormal, gamma, Weibull and other heavy-tailed (mixtures of) distributions*. The aforementioned families all contain at least two parameters; in general, the parameters are estimated from data using methods such as the maximum likelihood or the method of moments. This, however, requires having a sufficient number of datapoints; as a rule of thumb, 50-100 datapoints seems a reasonable minimum. In scenario analysis performed for operational risk management, however, a different approach is followed.

For a given risk, a two-parameter distribution is chosen for the severity (e.g. the lognormal) and experts are requested to provide two values: ***typical case loss*** and ***worst case loss***. This combination is chosen because it provides the minimal amount of information required to describe the main features of the distribution, that is a value which is experienced frequently and a value which is extreme (experienced rarely).

Usually, the typical case loss is identified with a location index, such as the median or the mode of the distribution. The mean is often considered a less stable location index, since it is influenced by the presence of outliers. The worst case loss is identified with a suitably large quantile of the distribution (e.g. the 99.9% percentile). This gives a nonlinear system of two equations in two variables (the parameters of the distribution), which can be solved by a Newton-like numerical approximation method.

Typically, in operational risk loss modelling one assigns a relatively small value to the typical case, while the worst case might be significantly larger. This entails that the ensuing distribution is characterised by strong asymmetry (skew) towards the upper right tail. This characteristic is in fact often observed for operational loss data.

We adopted the **Lognormal distribution** for modelling the severity (consequence) of the losses generated by a unwanted incident. This distribution has cumulative distribution function (CDF):

$$\mathrm{F}(\mathrm{x}, \mu, \sigma) = \Phi\left(\frac{\ln(x - \mu)}{\sigma}\right),$$

Where x denotes the lognormally distributed variable (i.e. the severity for the asset), Φ denotes the CDF of the standard Normal distribution and μ and σ denote the mean and standard deviation of the natural logarithm of x, respectively.

As explained above, the median is typically considered in risk management, as it provides a more robust location index than the mean. For the lognormal distribution, the median is given by $\exp(\mu)$.

To estimate Lognormal parameters, numerical values for *at least* two data points (x_1, x_2) are required. For example, assuming that one knows an estimate x_1 for the median (typical case) and an estimate x_2 for a high percentile (worst case), say corresponding to a probability level p (e.g. 99.9%), then one can estimate the parameters of the Lognormal distribution by solving the following system of equations in the two unknowns (μ, σ):

$$F(x_2; \mu, \sigma) = p, \qquad \exp(\mu) = x_1.$$

The two data points (x_1, x_2) can be obtained by asking typical and worst case loss information from the WISER user, who might have more information regarding the context where these losses arise. Guidelines for this kind of economic impact assessment are provided in Section 11. For the WISER risk patterns, we have defined default values for the case where the user does not provide this input.

8.3 Worked example

Let us consider the risk model described by the CORAS diagram in Figure 15. This shows the same model as in Figure 13, except that we have added labels following the convention presented in Appendix I. We distinguish the following elements in the risk model:

1) Threat scenarios:
   a) S1:Hacker intercepts HTTP connection;
   b) S2: Hacker carries out SQL injection;
   c) S3: Hacker obtains account user name and password.
2) *'leads to'* relations:
   a) cl_S1_to_S3;
   b) cl_S2_to_S3;
   c) cl_S3_to_U1.
3) Unwanted incidents:
   a) U1: Confidential user data is disclosed.
4) *'impacts'* relations:
   a) c_U1_A1
5) Affected assets:
   a) Confidentiality.
6) Vulnerabilities:
   a) Lack of proper session management;
   b) Insufficient input validation;
   c) Weak login policy.
7) Mitigations:
   a) M1: Validate all input provided by user.

8) Indicators:

   a) I1: How many HTTP requests contain special elements used in a SQL command?

   b) I2: How many HTTP requests containing special elements used in a SQL command are successfully executed?

   c) I3: What is the consequence of U1 on the asset Confidentiality, given that U1 occurs?

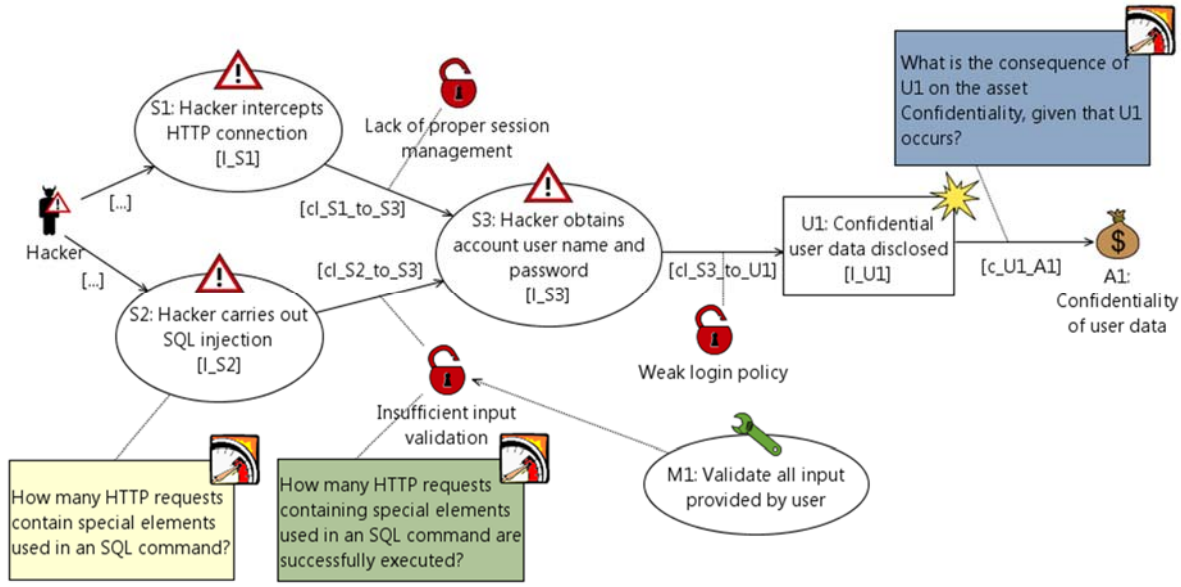


Figure 15. CORAS risk model with identifiers and variable names.

The first step on defining a BN model from a CORAS diagram is to define a BN skeleton corresponding to the CORAS graph (i.e. without probability distributions on nodes). Figure 16 shows the BN skeleton for the CORAS model in Figure 15, where we represented all nodes as empty ovals, because no probability distribution has been assigned yet. The **R** script generating the BN skeleton is given in Table 5. The severity for the affected asset is represented by node `severity_C`: this will contain a *mixture distribution* consisting of an atom at 0 and a positive loss in the node `c_U1_to_C` (modelled by a Lognormal distribution). This represents the idea that, given that the unwanted incident occurs, not all such occurrences necessarily produce an economic loss. In particular, we set `f_C` to be the fraction of unwanted events leading to a loss, as the parameter of a Bernoulli distribution in the father node `frac_U1_to_C`. The severity node `severity_C` is then a deterministic node whose value is given by the product of the values of the two nodes `c_U1_to_I` and `frac_U1_to_I`. We decided to assume that unwanted events always lead to losses, hence `f_C=1` and `c_U1_to_C = severity_C`.

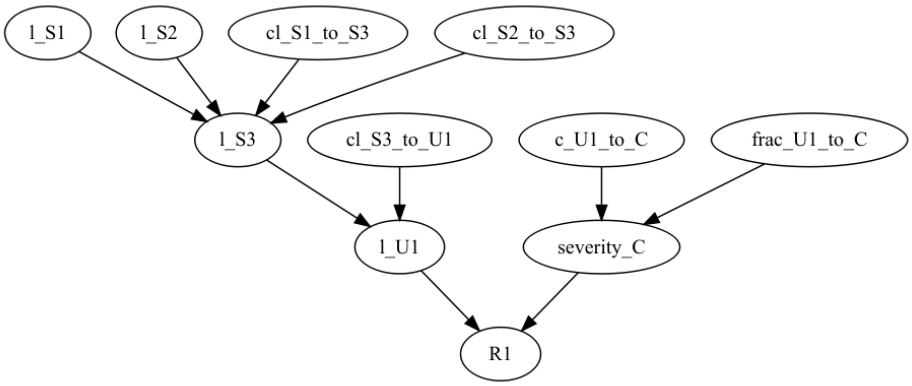


Figure 16. Hydenet skeleton for the BN corresponding to the CORAS model in Figure 15.

```
##Hydenet skeleton
>net <- HydeNetwork(~ l_S1
                    + l_S2
                    + l_S3 | l_S1*cl_S1_to_S3*l_S2*cl_S2_to_S3
                    + l_U1 | l_S3*cl_S3_to_U1
                    + c_U1_to_C
                    + frac_U1_to_C
                    + severity_C | c_U1_to_C * frac_U1_to_C
                    + R1 | l_U1 * severity_C )
>plot(net)
```

Table 5. **R** code for Hydenet skeleton for the CORAS model in Figure 13.

The probability distribution for the frequency of the unwanted incident `U1` is determined as follows, (see Table 6 for the **R** code):

1. assign to nodes `l_S1` and `cl_S1_to_S3` the uniform distribution in an interval with extremes determined by historical data and expert judgement;
2. assign to node `l_S2` (and to `cl_S2_to_S3`) the uniform distribution in an interval with extremes determined by historical data, expert judgement and indicator I1 (and indicator I2);
3. assign to node `cl_S2_to_S3` the uniform distribution in an interval with extremes determined by historical data, expert judgement and indicator I2;
4. assign to the node `l_S3` the distribution given by: `l_S1*cl_S1_to_S3 + l_S2*cl_S2_to_S3;`
5. assign to node `cl_S3_to_U1` the distribution given by: `l_S3 * cl_S2_to_U1.`

Notice that both the extremes of the intervals for the distributions for `l_S2` and `cl_S2_to_S3` depend, respectively, on indicators I1 and I2. Moreover, nodes `l_S3` and `l_U1` are deterministic, since their values are calculated by a formula from the values of their parent nodes. Table 6 contains the **R** code for the definition of the probability distribution of the unwanted incident U1 in the CORAS diagram Figure 15.

```
##Distributions for the frequency nodes
  net <- setNode(net, l_S1,
                 nodeType="dunif", a=par_l_S1$a, b=par_l_S1$b)
```

```
net <- setNode(net, cl_S1_to_S3,
               nodeType="dunif", a=par_cl_S1_to_S3$a, b=par_cl_S1_to_S3$b)

net <- setNode(net, l_S2,
               nodeType="dunif", a=par_l_S2$a, b=par_l_S2$b)
net <- setNode(net, cl_S2_to_S3,
               nodeType="dunif", a=par_cl_S2_to_S3$a, b=par_cl_S2_to_S3$b)

net <- setNode(net, l_S3,
               nodeType="determ", define=fromFormula(),
               nodeFormula = l_S3 ~ l_S1 * cl_S1_to_S3 + l_S2 * cl_S2_to_S3 )

net <- setNode(net, cl_S3_to_U1,
               nodeType="dunif", a=par_cl_S3_to_U1$a, b=par_cl_S3_to_U1$b)
net <- setNode(net, l_U1,
               nodeType="determ", define=fromFormula(),
               nodeFormula = l_U1 ~ l_S3 * cl_S3_to_U1)
```

Table 6. Definition of distributions for the frequency nodes

To estimate the parameters of the lognormal distribution in `c_U1_to_C`, the following information is required:

1. typical case;
2. worst case;
3. the probability assigned to the worst case.

These values should be provided by the user of the model (and assigned to the consequence indicator IN_CC); if the user is unable to do so, default values have to been set by collecting publicly available information. The parameters of the Lognormal distribution are then obtained solving the following two equations:

$$F(x_2; \mu, \sigma) = p, \qquad \exp(\mu) = x_1,$$

where x_1, x_2 and p are, respectively, the typical case, the worst case and the probability for the worst case. Table 7 shows the function in **R** used to compute the Lognormal parameters from the typical case, the worst case and the probability assigned to the worst case. HydeNet uses the parametrization with the inverse (tau) of the variance.

```
par.lognormal <- function(typical, worst, prob){
                                        if( typical > 0 & worst > 0 ){
                                           mu    <- log(typical)
                                           sigma <- (log(worst)-mu)/qnorm(prob)
                                           tau   <- 1/sigma^2
                                         }else{
                                                  mu <- tau <- NA
                                         }
                                         return(c(mu,tau))
                                    }
```

Table 7. **R** script for the function computing Lognormal parameters used in Hydenet.

The probability distribution for *the aggregate annual loss* (and hence the risk level) for Confidentiality (node `R1`) is then defined as the multiplication of the distributions of `l_U1` (frequency) and `severity_C` (Table 8).

```
## Distributions for consequence nodes
  ## Confidentiality
  if( is.na(par_c_U1_C$mu) ){
    net <- setNode(net, c_U1_to_C,
                   nodeType="determ", define=fromFormula(),
                   nodeFormula =  c_U1_to_C ~ 0)
  }else{
    net <- setNode(net, c_U1_to_C,
                   nodeType="dlnorm", mu=par_c_U1_C$mu, tau=par_c_U1_C$tau)
  }
  net <- setNode(net, frac_U1_to_C,
                 nodeType="dbern", p=par_c_U1_C$frac, validate=FALSE)
  net <- setNode(net, severity_C,
                 nodeType="determ", define=fromFormula(),
                 nodeFormula = severity_C ~ c_U1_to_C * frac_U1_to_C)
  net <- setNode(net, R1,
                 nodeType="determ", define=fromFormula(),
                 nodeFormula = R1 ~ l_U1 * severity_C)
```

Table 8. Definition of distribution for consequence nodes.

Figure 17 shows the BN model for CORAS diagram in Figure 15. The elliptic blue nodes represent random variables, the green rectangular nodes represent deterministic nodes.

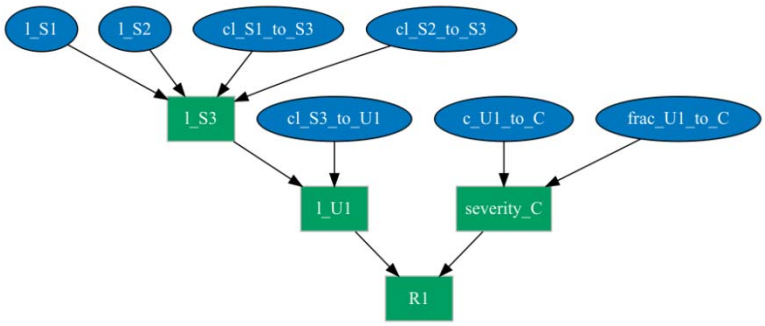


Figure 17. BN for CORAS model in Figure 15.

Appendix III contains numerical results of a Monte Carlo simulation with 10^5 iterations different scenarios for the BN in Figure 17 as well as the corresponding **R** script.

8.4 Mitigation proposal triggering

One mitigation option, M1, has been identified in Figure 15. Table 9 shows an **R** fragment used to trigger a proposal for implementing mitigation M1.

```
#### Assigning True/False (Yes/No) values for the mitigation triggers. True means suggest
mitigation, false means do not suggest mitigation.

    freqThreshold <- 0.035;
    lossThreshold <- 120;
```

```
# M1: Validate all input provided by user
M1 <- mean.freq >= freqThreshold | typical.R1 >= lossThreshold;
```

Table 9. **R** fragment for triggering a mitigation proposal

This is done by assigning M1 a Boolean value, i.e. either TRUE or FALSE. If the value is TRUE, it means that the system should propose M1 as a mitigation option (possibly one among many). Here, the proposal is triggered one or both of the following conditions are fulfilled:

1) The mean frequency of the incident has reached a set frequency threshold.
2) The typical loss from the risk has reached a set loss threshold.

Other conditions can of course be defined.

9 Defining qualitative assessment algorithms using DEXi from CORAS diagrams

In this section, we provide guidelines for defining qualitative assessment algorithms using DEXi based on a CORAS diagram. As will be shown, the structure of a CORAS diagram can be exploited to create a corresponding DEXi model. We also show how to trigger mitigation proposals. An example of a DEXi model resulting from following the guidelines based on the CORAS model in Figure 3 is provided in Appendix IV.

CORAS defines a number of different diagram types [15]. However, for our purposes we primarily exploit the structure of the threat and/or treatment diagrams (with indicators). Hence, by CORAS diagram, we therefore mean this particular types of diagram unless otherwise specified.

In a CORAS diagram, there are two primary 'occurrences' for which a likelihood is assigned: threat scenarios and incidents. As these are treated in the same way with respect to likelihood assessment, we use the common term 'node' to refer to threat scenarios and incidents.

We use a modular approach for the guidelines. In each of the following subsections, we use a common structure. First, we present a fragment of a CORAS diagram. Then, we explain how to represent this fragment as a single root node with sub-nodes in a DEXi model. Finally, we present restrictions on the utility function that define the aggregation of the values of the sub-nodes to the root node in the DEXi model. These restrictions are not intended to eliminate the need for subjective judgment when defining the utility function, but serves as an aid to help ensuring the soundness of the approach.

Notice that for all the qualitative scales to be used for the attributes in the DEXi model, we will use ordered scales where a low value represents or implies low risks (or low risk contributions), so that increasing values imply increasing risk. We find it easier to reason with such scales. This means that in our case, lower values are actually desirable.

9.1 Risk level

9.1.1 CORAS representation

A risk is the likelihood of an incident and its consequence for an asset. Hence, in order to assess the risk level, we need to assess the likelihood of the incident and its consequence for the asset in question. Figure 18 illustrates how a risk is represented in a CORAS threat diagram as a combination of an incident, an asset, and an 'impacts'-relation from the incident to the asset. Our naming convention is shown in 0. Notice that the square brackets are normally used to hold likelihood and consequence assessments. We have inserted the variable/node names to be used in the corresponding DEXi fragment, in order to make it easier to understand the connection.

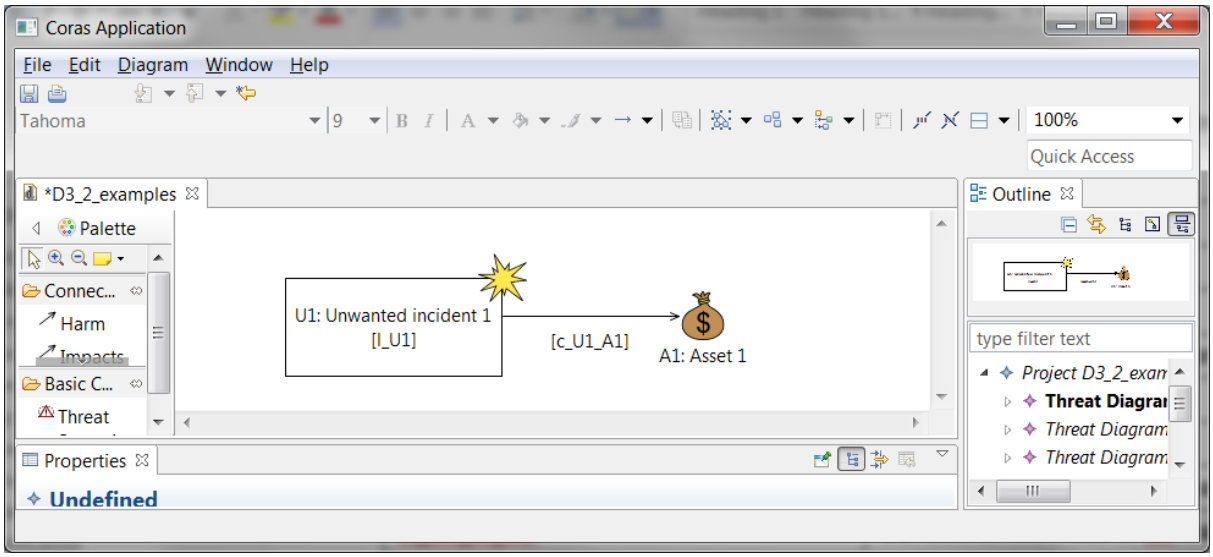


Figure 18. CORAS fragment representing a risk

9.1.2 DEXi representation

In the DEXi model, a risk node has two sub-nodes, one representing the likelihood and one representing the consequence for the asset in question. Figure 19 shows the DEXi-representation of the CORAS fragment shown in Figure 18, where R1 represents the risk, I_U1 represents the likelihood of the incident U1, and c_U1_A1 represents the consequence of U1 for asset A1. Notice that in the CORAS diagram, the risk does not have a separate name as it is not represented by a separate node, but by the combination of the incident, the asset and the relation between them. Moreover, although the number/index is identical for the risk, the incident, and the asset in this particular example, this need not necessarily be the case.

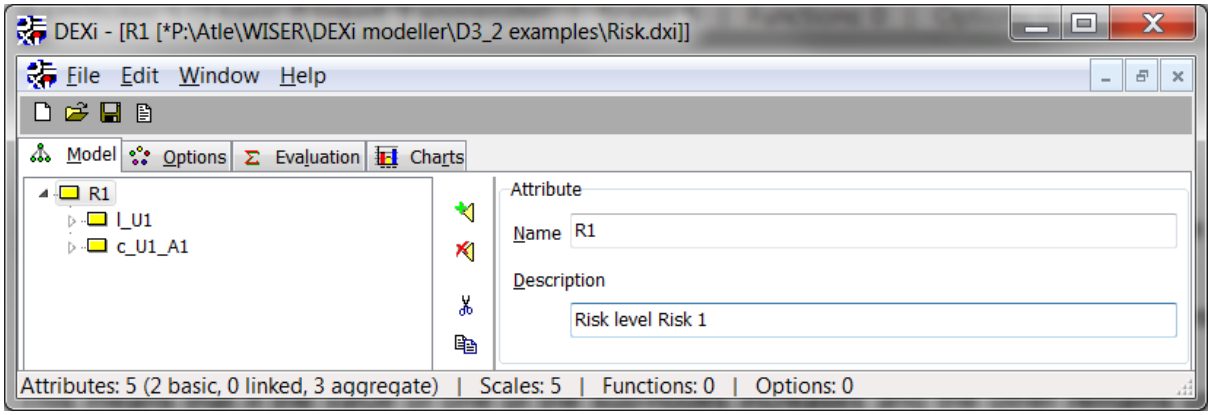


Figure 19. DEXi fragment representing a risk

9.1.3 Restrictions on utility function

Increasing the likelihood or the consequence of a risk can never lead to a reduction of the risk level. Therefore, the utility function of a risk node should ensure the following:

- The value of the risk node is monotonically increasing in both its sub-nodes. It does not have to be *strictly* increasing.

This means that if the value of one of the sub-nodes increases and the other remains unchanged, then value of the risk node should either increase or remain unchanged.

Example: In Figure 19, R1 should be monotonically increasing in I_U1 and c_U1_A1.

Figure 20 shows an example of how a utility function fulfilling this restriction might be defined.

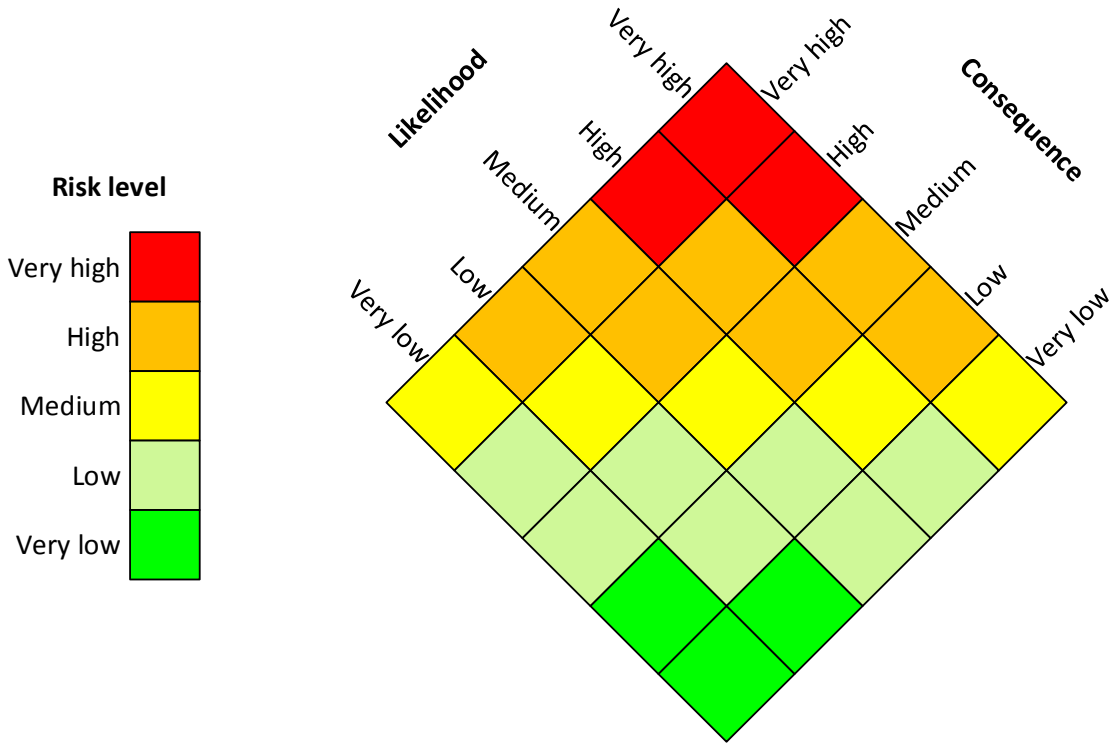


Figure 20. Example of utility function defining risk level as a function of likelihood and consequence

Here we have used the same five-step scale consisting of the steps Very low; Low; Medium; High; Very High for risk level, likelihood and consequence. The colour indicates the risk level, which is a function of the likelihood and the consequence. For example, if the likelihood is Low and the consequence is High, then the risk level is Medium, as indicated by the yellow colour.

9.2 Incoming 'leads-to' relations to a node

9.2.1 CORAS representation

Figure 21 shows a fragment of a CORAS diagram showing two nodes (threat scenarios S1 and S2) that may each lead to another node (threat scenario S3). This is represented by the 'leads-to' relation from each of S1 and S2 to S3. The likelihood of S3 thus depends on the likelihood of S1 and the conditional likelihood of an occurrence of S1 actually leading to an occurrence of S3, and similarly for S2.

Notice that the diagram is meant to represent an example of a more general case, where one or more nodes may lead to another node. Moreover, even if all nodes in this particular fragment are threat scenarios, each of them could equally well have been replaced by an incident without having any impact on the reasoning presented here.

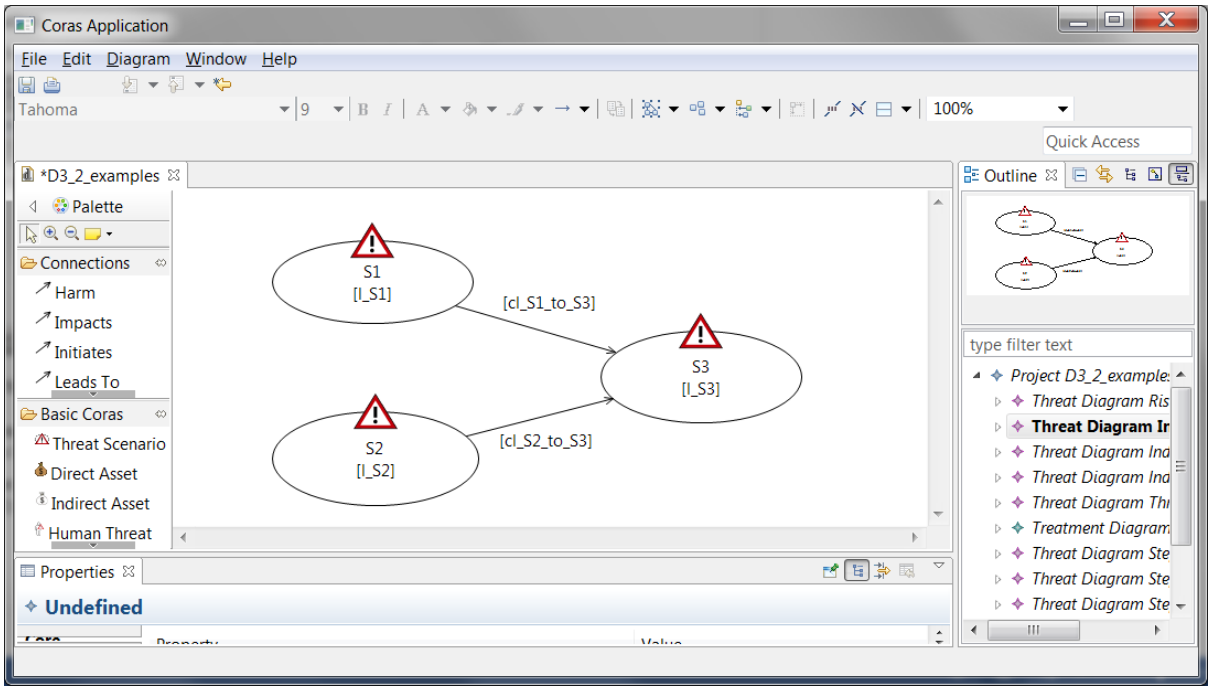


Figure 21. CORAS fragment representing incoming 'leads-to' relations

9.2.2 DEXi representation

Figure 22 shows a DEXi fragment corresponding to the CORAS fragment in Figure 21.

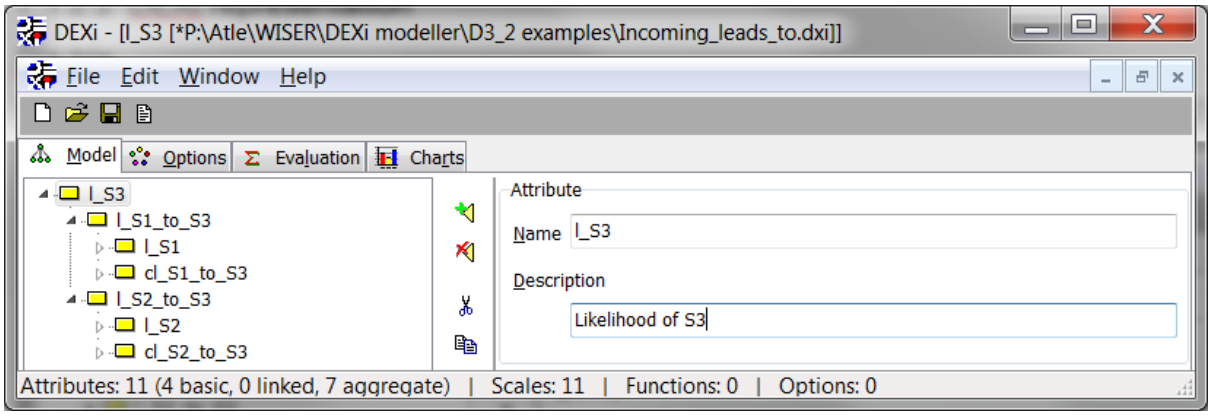


Figure 22. DEXi fragment representing incoming 'leads-to' relations

The root node (I_S3) represents the likelihood of S3. This has two direct sub-nodes, as it depends on the likelihood contribution from S1 (I_S1_to_S3) and the likelihood contribution from S2 (I_S2_to_S3).

The likelihood contribution from S1 to S3 again has two direct sub-nodes, showing that it depends on the likelihood of S1 (I_S1) as well as the conditional likelihood of an occurrence of S1 actually leading to S3 (cl_S1_to_S3). Similarly, the likelihood contribution from S2 to S3 depends on the likelihood of S2 (I_S2) as well as the conditional likelihood of S2 leading to S3 (cl_S2_to_S3).

Figure 22 shows only one example, where there are two incoming branches to S3. In general, the number of direct sub-nodes to S3 will be equal to the number of incoming branches. However, it is important to avoid having to too many incoming branches to a node, as this makes it hard to define the utility function. When using five-step scales as in the example, even three incoming branches would give 125 possible combinations. This is can already be hard to handle, and more branches

would be completely unfeasible. In such cases, we recommend restructuring the model, as further explained in the DEXi manual [1].

Observe that the nodes representing likelihoods of S1 and S2 occur at the bottom/leaf layer of the DEXi fragment in Figure 22. As these may again depend on incoming branches, the model allows any finite number of levels in the DEXi tree.

9.2.3 Restrictions on utility function

Since the DEXi tree structure addressed in this chapter has tree levels, it involves two layers of utility functions. The first defines, at the level of the root node (l_S3), the aggregation of the likelihood contributions from all incoming branches (l_S1_to_S3 and l_s2_to_S3). The second defines, at the level of each incoming branch (either l_S1_to_S3 or l_S2_to_S3), the aggregation of the likelihood of the source-node (either l_S1 or l_S2) and the conditional likelihood that an occurrence will lead to the target node (either cl_S1_to_S3 or cl_S2_to_S3). We therefore define two separate restrictions on the utility function.

Increasing the likelihood contribution from a branch can never lead to a decreased likelihood for the target node. Therefore, for the aggregation at the level of the root node, the utility function should ensure the following:

- The value of the root node (l_S3 in Figure 22) is monotonically increasing in all its direct sub-nodes (l_S1_to_S3 and l_S1_to_S3 in Figure 22). It does not have to be *strictly* increasing.

Example: In Figure 22, l_S3 should be monotonically increasing in l_S1_to_S3 as well as l_S2_to_S3.

Increasing the likelihood of a source node or the conditional likelihood that an occurrence will lead to the target node can never reduce the likelihood contribution to a target node. Moreover, the impact of the conditional likelihood can only affect the target node to the extent that the source node actually occurs. Therefore, for the aggregation at the level of each incoming branch, the utility function should ensure the following:

- The value of the likelihood contribution from an incoming branch should be monotonically increasing in the likelihood of the source node as well as the conditional likelihood that an occurrence will lead to the target node.
- The value of the likelihood contribution from an incoming branch should never be higher than the likelihood of the source node.[2]

Example: In Figure 22, l_S1_to_S3 should be monotonically increasing in l_S1 and cl_S1_to_S3. Moreover, l_S1_to_S3 should never by higher than l_S1.

9.3 Indicators attached to a node

9.3.1 CORAS representation

Indicators can be attached to a node in order to show that the indicators are used as input for assessing the likelihood of the node. Figure 23 shows a fragment of a CORAS diagram where two indicators, I1 and I2, have been attached to a node S1. The indicators are represented as 'notes', where the colour denotes the indicator type. However, the indicator type is not important for our purposes here, as they are all treated the same with respect to the guidelines.

[2] This restriction can, however, be lifted if we assume that one occurrence of the source node can lead to several occurrences of the target node.

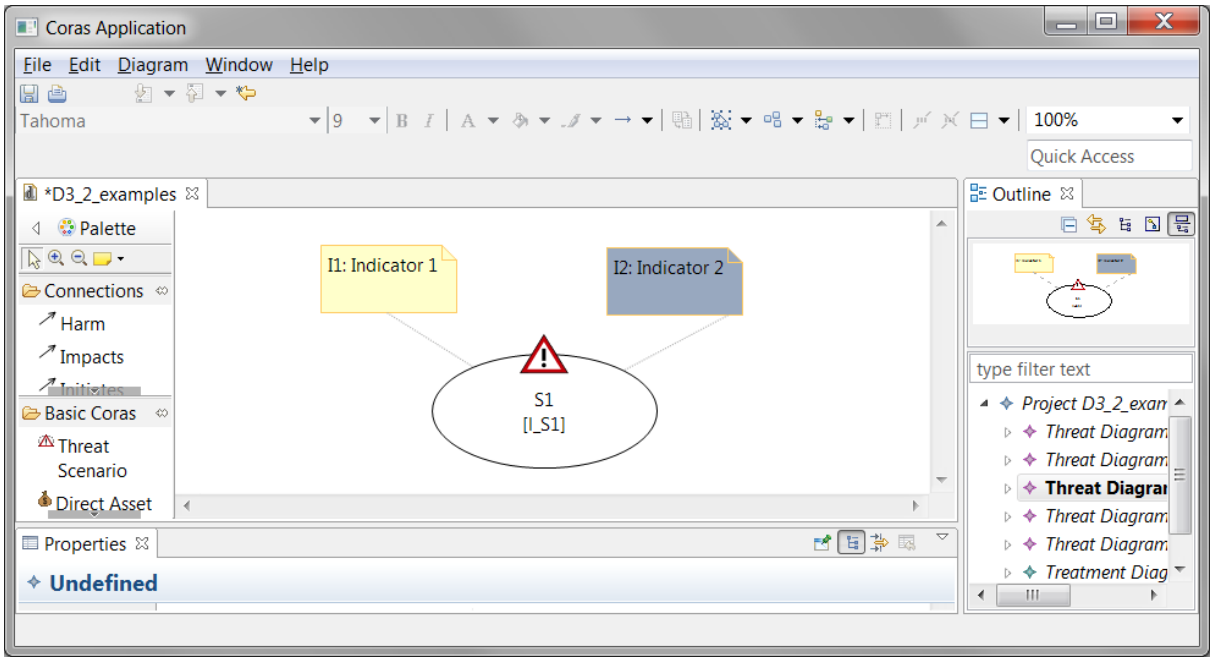


Figure 23. CORAS fragment representing a node with attached indicators

Notice that in CORAS diagrams, a branch always starts with a threat initiating a node. However, we rarely assign likelihoods to the threats themselves or to the 'initiates' relation from a threat to a node, but rather to the node. Any indicators assigned to a threat or to an 'initiates' relation can therefore be handled as if it was assigned directly to the node, following the guidelines of this subsection.

9.3.2 DEXi representation

Figure 24 shows a DEXi fragment corresponding to the CORAS fragment in Figure 23.

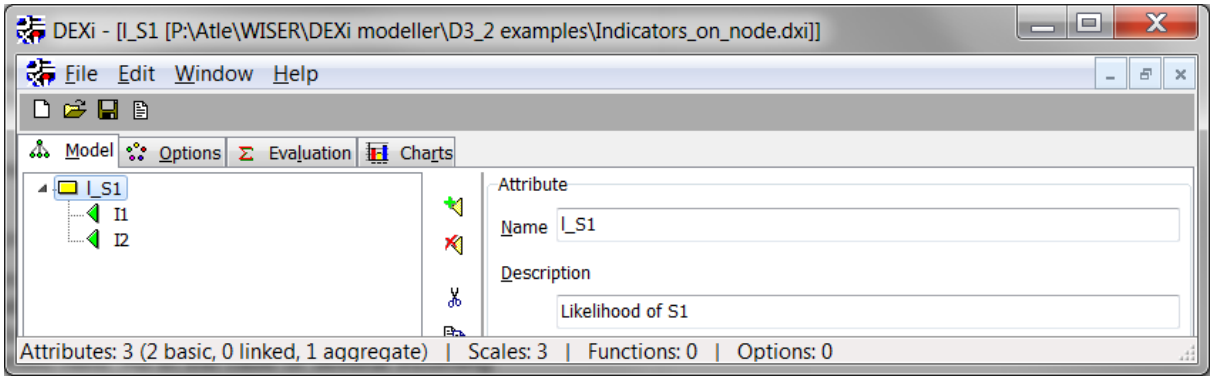


Figure 24. DEXi fragment representing a node with attached indicators

Here, there is one direct sub-node (which is also a leaf-node, and hence shown as a triangle) to the root node for each attached indicator. Hence, the likelihood of the root node (I_S1) depends on these indicators.

Before the utility function of I_S1 can be defined, an ordered scale has to be defined for each indicator. Although the indicators do not necessarily represent a likelihood, we make sure to define the scale in such that a low value implies a low risk contribution.

For example, assume that a threat scenario representing initiation of a HTTP Request/Response splitting is included in a risk model for client-server protocol manipulation. To this threat scenario, we

attach the indicator 'Has any network reconnaissance attempt been detected in the past?' Since this is a yes/no question, the scale for the indicator only has two steps: Yes and No. A positive answer may indicate that someone has tried to prepare for an attack, and hence an increased likelihood. Therefore, for this indicator scale, the order from lowest to highest value would be No; Yes.

9.3.3 Restrictions on utility function

Assuming that all indicator scales have been defined as above, the likelihood of the node can never decrease if an indicator increases. Therefore, the utility function of a node with attached indicators should ensure the following:

- The likelihood of the node is monotonically increasing in all its attached indicators. It does not have to be *strictly* increasing.

Example: In Figure 24, l_S1 should be monotonically increasing in I1 as well as I2.

Notice that we may have cases where a node has incoming branches, as addressed in Section 9.2 in addition to attached indicators. In such cases, the likelihood of the node depends on the incoming branches as well as the attached indicators. The utility function should then fulfil the conjunction of the restrictions from Section 9.2.3 and the restriction presented here. As in the case of several incoming branches to a node, it may be necessary to restructure the model to avoid combinatorial explosion.

9.4 Indicators attached to a 'leads-to' relation

9.4.1 CORAS representation

Indicators can be attached to a 'leads-to' relation from one node to another to show that the indicators are used as input for assessing the conditional likelihood of an occurrence of the source node leading to the target node. Normally, this is done by attaching the indicators to a vulnerability on the 'leads-to relation', as the indicators typically say something about the presence or severity of the vulnerability. Figure 25 shows a fragment of a CORAS diagram where two indicators, I1 and I2, have been attached to a vulnerability on the 'leads-to' relation between two nodes.

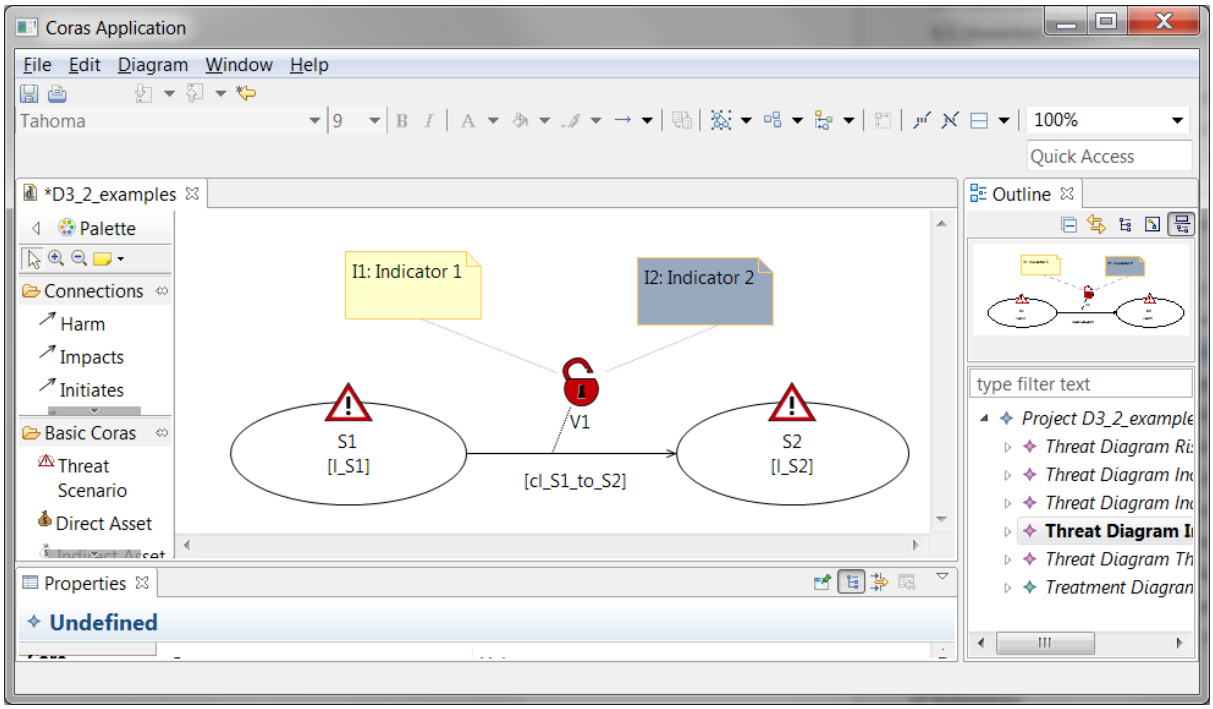


Figure 25. CORAS fragment representing 'leads-to' relation with indicators

9.4.2 DEXi representation

Figure 26 shows a DEXi fragment corresponding to the CORAS fragment in Figure 25.

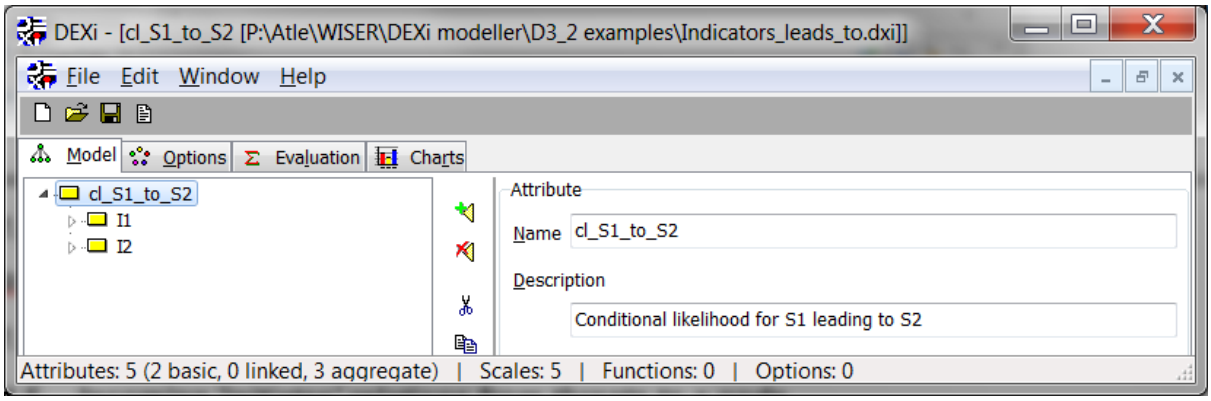


Figure 26. DEXi fragment representing 'leads-to' relation with indicators

The root node (cl_S1_to_S2) represents the conditional likelihood that an occurrence of the source node (S1) will lead to the target node (S2). Here, there is one direct sub-node to the root node for each attached indicator. Hence, the likelihood of the root node (cl_S1_to_S2) depends on these indicators.

As for the case with indicators attached to a node, before the utility function of cl_S1_to_S2 can be defined, we have to define an ordered scale for each indicator. This is done in the same way as described in Section 9.3.

9.4.3 Restrictions on utility function

Assuming that all indicator scales have been defined such that low values imply low risk contributions, the conditional likelihood of the 'leads-to' relation can never decrease if an indicator increases. Therefore, the utility function of a 'leads-to' relation with attached indicators should ensure the following:

- The conditional likelihood of the 'leads-to' relation is monotonically increasing in all its attached indicators. It does not have to be *strictly* increasing.

Example: In Figure 26, cl_S1_to_S2 should be monotonically increasing in I1 as well as I2.

9.5 Mitigation proposal triggering

9.5.1 CORAS representation

CORAS diagrams can be used to show risk mitigation options by attaching these to the different elements of a risk model. Figure 27 shows such a diagram.

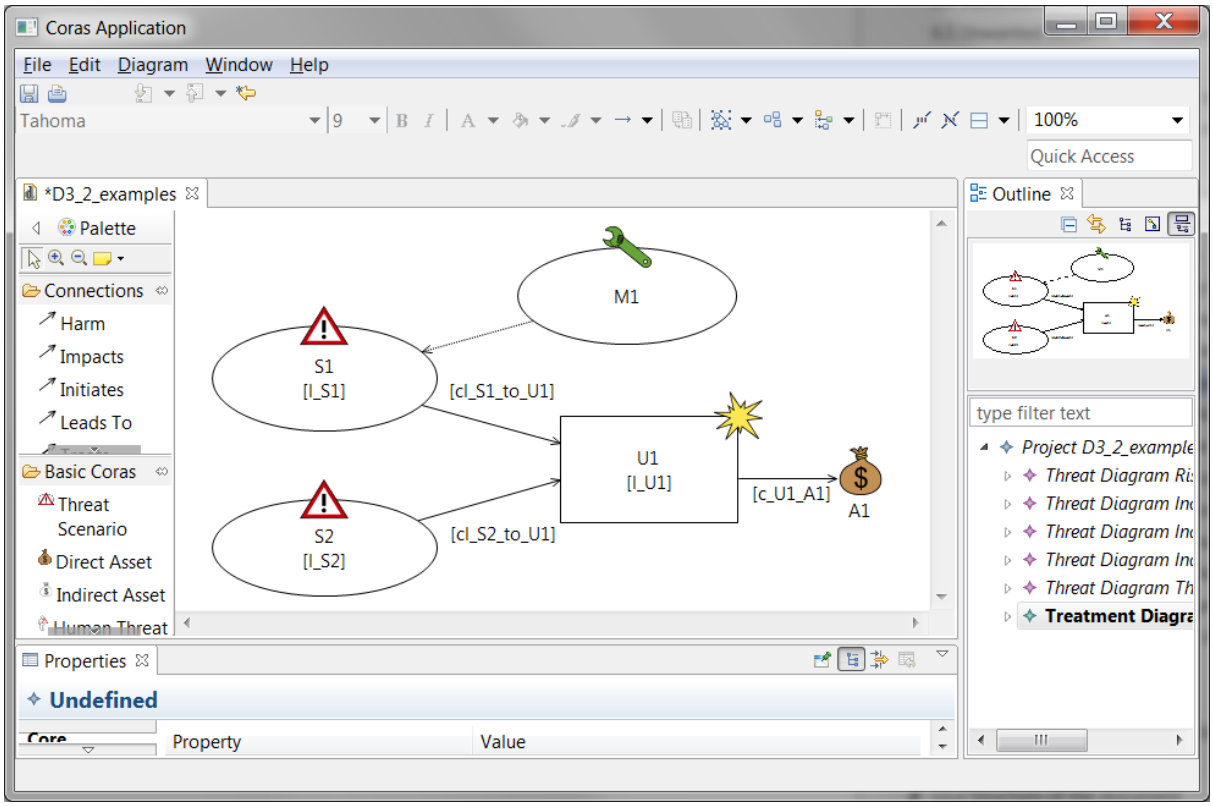


Figure 27. CORAS fragment associated with mitigation proposal

Here, the mitigation option M1 is attached to threat scenario S1, indicating that implementing M1 will reduce the likelihood of S1, which could also reduce the likelihood of U1, and hence the associated risk.

9.5.2 DEXi representation

Figure 28 shows a DEXi fragment used to trigger a proposal for implementing mitigation M1.

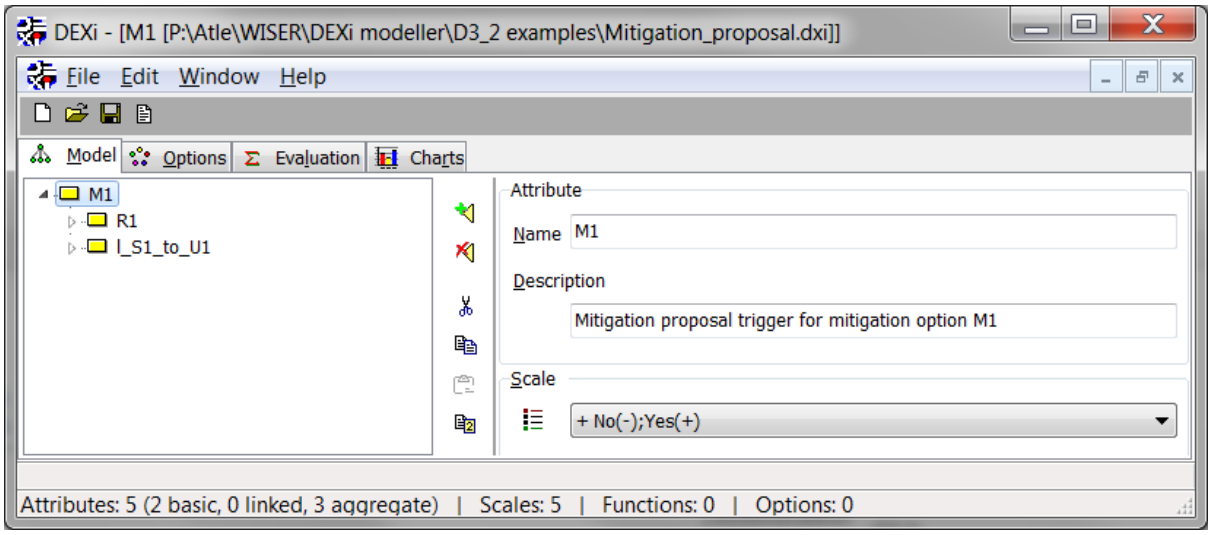


Figure 28. DEXi fragment for triggering a mitigation proposal

The scale of the root node M1 has only two steps: No and Yes. If the value is Yes, this means that the system should propose M1 as a mitigation option (possibly one among many). This should only be done if the following holds:

1) At least one risk that M1 has the potential to reduce is sufficiently high to warrant the proposal, and
2) the contribution to this risk from the branch to which M1 is attached is sufficiently high that a reduction of this contribution can significantly reduce the risk level.

According to Figure 27, the only risk that M1 has the potential to reduce is the risk of incident U1 harming asset A1. In Figure 28, this is represented by R1. Hence, the first condition above is only fulfilled if R1 is sufficiently high. Moreover, the second condition is only fulfilled if the contribution to risk R1 from the branch that includes S1 is also sufficiently high.

Therefore, the root node M1 has two direct sub-nodes: R1, representing the relevant risk level (and hence the first condition above), and l_S1_to_U1, representing the likelihood contribution from S1 to U1 (and hence the second condition above).

9.5.3 Restrictions on utility function

The scale of a mitigation proposal should include two ordered steps, No and Yes, where No is the low value, indicating the mitigation should *not* be proposed, and Yes is the high value indicating that the mitigation *should* be proposed.

A mitigation proposal should never be turned off due an increase in the relevant risk level or an increase in the likelihood contribution from the branch to which the mitigation proposal is attached. Therefore, the utility function should ensure the following:

- The value of the mitigation proposal (on the simple two-step scale No; Yes) should be monotonically increasing in all its direct sub-nodes. It does not have to be *strictly* increasing.

Example: In Figure 28, the value of M1 should be monotonically increasing in R1 and l_S1_to_U1.

10 Integration of assessment algorithms in the Risk Assessment Engine

In this section, we report the most relevant changes and improvements concerning the integration of assessment algorithms into the WISER Risk Assessment Engine component, until achieving the final integration. Our reference point is the already released deliverable D5.2, which documents such integration in its Section 3 (WISER real-time assessment infrastructure). This document was delivered in M16 (September 2016), so the main highlights that happened from that point onwards will be described in this section. No major changes from the functional point of view took place, hence this section rather refers to the implementation aspect.

10.1 Triggering Cases

The Risk Assessment Engine does not continuously evaluate the risk assessment of an organization, it is triggered only when something relevant happens. Below, we explain how the triggering works in the case of CyberWISER-Essential and CyberWISER-Plus. This description is an updated version of that given in section 3.1.4 of D5.2.

The triggering cases are the same as described in D5.2 but we have changed the way the Risk Assessment Engine is notified of any change in the business indicators or in the selected risk models. To simplify the architecture, currently the WISER Dashboard only interacts with the Data Warehouse. For this reason, it is the Data Warehouse component that is responsible for informing the 'Triggering Detector' module of the Risk Assessment Engine when some change takes place in the tables where the company profile and the selected risk models are stored.

Now, this Triggering Detector module subscribes to the RabbitMQ queue called "**rae.td-dwhnotifs**" to receive a notification when there is a change in an indicator (business, testing, network or application

indicator) or if there is a change in the selected risk models or in a specific model. The following shows an example of such a notification:
{"object_ID": 1, "action": "CREATED", "object_class": "IndicatorValue"}

Three actions can be notified in this way from the DWH to the Risk Assessment Engine: “CREATED”, “DELETED” or “UPDATED”. The object_class parameter will be “IndicatorValue”, “RiskModel” or “SelectedRiskModel” depending on the type of notification.

Figure 29 illustrates the functioning of the Risk Assessment Engine when there is a change in the model/s used to produce the risk assessment report. This is specific for CyberWISER-Essential and CyberWISER-Plus. In the following explanation, the changes with respect to D5.2 refers to step (3) and is highlighted in bold.

First, the user selects new model/s to base the risk evaluation on. To do so, he interacts with the Modelling module by means of the Dashboard (1). The reference to the active model is stored in the Data Warehouse (2). **The Data Warehouse informs the triggering detector, by means of a notification sent to RabbitMQ, about the change in the model/s** (3). This module invokes the DEXi Model Rules Executor and the **R** Model Rules Executor (4a)(4b), in charge of carrying out the qualitative and quantitative evaluations respectively. The invocation of the DEXi and/or **R** Model Rules Executor will depend on the selection done by the user, who can select, for each risk model, whether to use the qualitative (DEXi) model or the quantitative (**R**) model, or both. In case one of them (DEXi or **R**) has not been selected by the user, the associated rules executor will not be invoked. Then, the DEXi Model Instantiator and the **R** Model instantiator are respectively called (5a)(5b). Both instantiators retrieve from the Data Warehouse the model/s being used and the indicators to populate these models. The former was provided by the Modelling module and the latter by the indicator values generator (6a)(7a)(6b)(7b). Then, the instantiators produce the different model instances and gives back the control to the DEXi and/or **R** Model Rules Executor (8a)(8b). Both Model Rules Executors use those model instances to perform the calculations aimed at obtaining the risk assessment report, and store the results in the Data Warehouse (9a)(9b). A report per infrastructure element is produced. Then, both DEXi and **R** Model Rules Executor call the Aggregator (10a)(10b). The Aggregator retrieves the reports produced for each of the infrastructure elements (11)(12) and produces the global reports (both qualitative and quantitative) where the risk associated to the infrastructure as a whole and the likely mitigation measures are reflected (13).

Figure 30 illustrates the functioning of the Risk Assessment Engine when this is triggered because of a change in the business variables (indicators). Again, we highlight changes with respect to D5.2 in bold in the following description.

The user, by means of the Dashboard, changes something in the configuration and this change is stored in the Data Warehouse (1). By means of the events message queue, the Indicator Values Generator is informed about the change in the configuration (2). The Indicator Values Generator is executed to produce the updated values of the indicators and they are stored in the Data Warehouse (3). **Once this is done, the Data Warehouse notifies the Triggering Detector** (4). Since it is likely that more than one business indicator values change when the user updates the questionnaire, the Triggering Detector waits for a while and after a timeout it invokes the DEXi Model Rules Executor and the **R** Model Rules Executor (5a)(5b). In turn, they respectively call the DEXi and **R** model instantiators (6a)(6b). Both instantiators retrieve from the Data Warehouse the current indicator values along with the models to populate (7a)(7b)(8a)(8b) and produce the instances of the qualitative and quantitative models. Then, both instantiators return the control to their respective Model Rules Executors (9a)(9b). Both executors retrieve their respective model instances and compute the qualitative and quantitative risk evaluation per infrastructure target, storing this information in the Data Warehouse (10a)(10b). The Aggregator is then called by each of the executors (11a)(11b). It retrieves the qualitative and quantitative risk reports per target (12a)(12b)(13a)(13b), and obtains the aggregated reports addressing the infrastructure as a whole, and stores them in the Data Warehouse (14a)(14b).

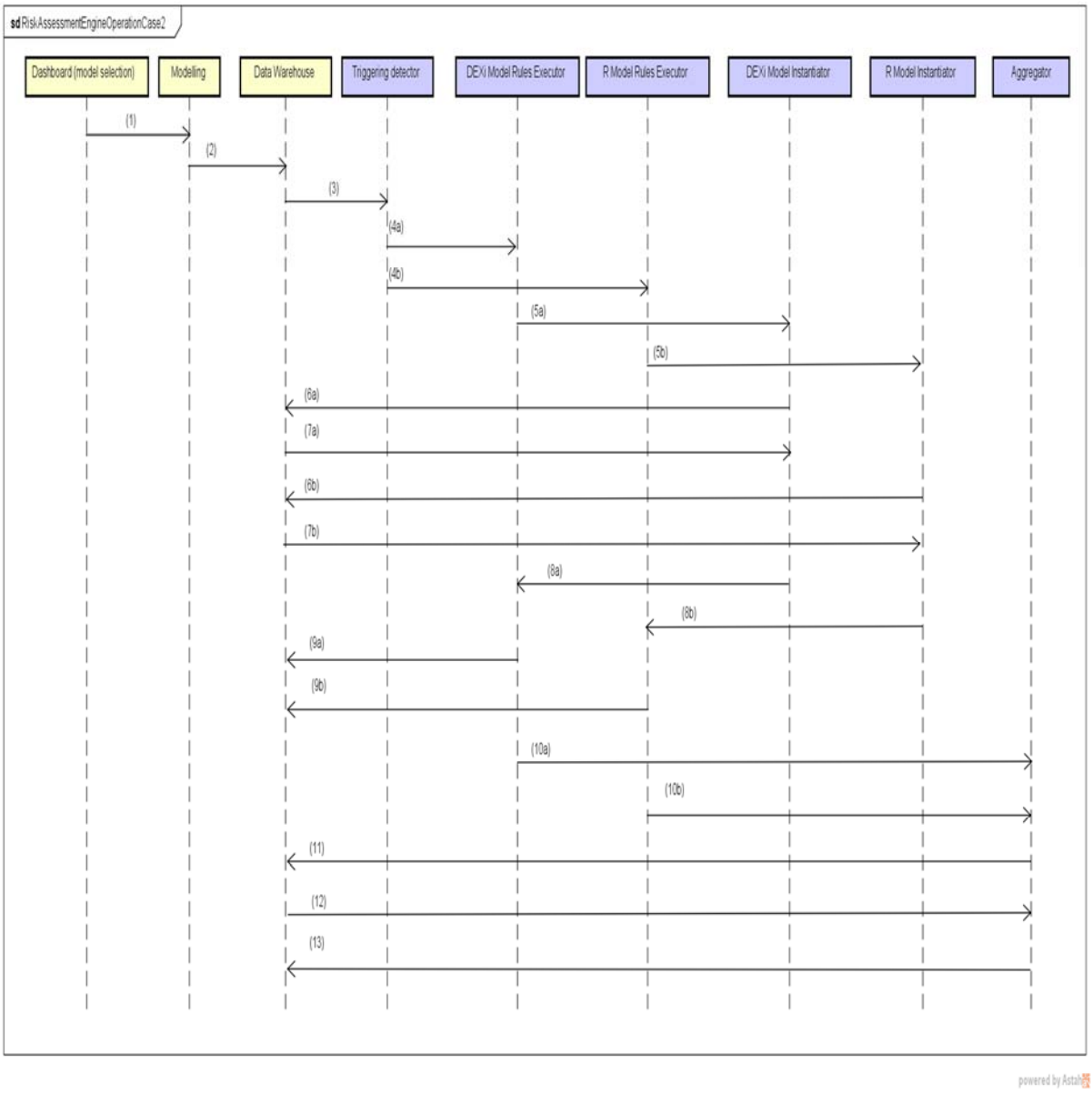


Figure 29. Risk Assessment Engine operation case 2. Change in model

The sequence when the Risk Assessment Engine detects a change in the monitoring or testing variables (indicators) is the same except for the fact that this time the Indicator Value Generator receives the events, alarms and vulnerabilities detected from the Monitoring module instead of from the Data Warehouse. The Dashboard (the user interface) does not play a relevant role in these triggering cases.

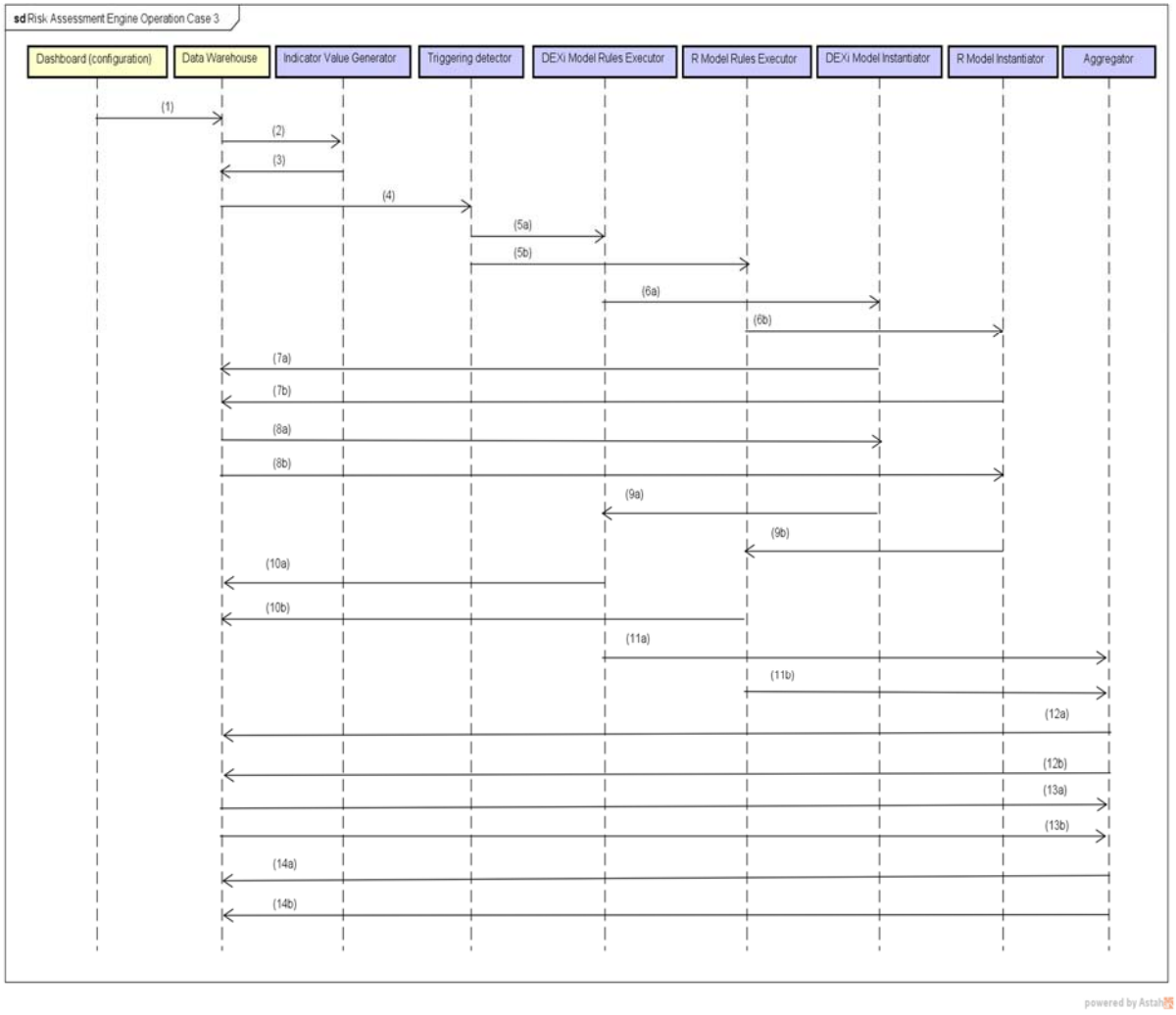


Figure 30. Risk Assessment Engine operation case 3. Change in business variable

10.2 Implementation

The following description of the Risk Assessment Engine implementation is an updated version of the one given in Section 3.2 of D5.2. Rather than repeating the complete description of each module, in the following we focus only on the changes with respect to D5.2.

10.2.1 Indicator Value Generator

The Indicator Value Generator module is in charge of receiving the different inputs required for the generation of the indicators values used in the Risk Assessment Engine and storing those indicator values in the Data Warehouse for its usage by the rest of modules.

These are the main changes in this module with respect to the description provided in D5.2 (section 3.2.1):

- **Default Values**
  Each indicator used in WISER has a predefined default value stored in the Data Warehouse in the Indicators Catalogue (/rae/indicators). These default values are used by the Risk Assessment Engine to initialize the indicator values associated to a target when the user selects a new model. The usage of these default values is also required in the case of risk

models associated to CyberWISER Plus service including specific application indicators which are not be available for CyberWISER Essential service users.

The current format of the indicators as they are stored in the Data Warehouse in the Indicators Catalogue is the following one:

```
{
 "id":<indicator id>,
  question":<Short description of the indicator>,
  "data_type":<Boolean|String|Integer>,
  “motivation":<motivation for the usage of this indicator>,
 "indicator_type":<1(bussiness)|2(testing)|3(network)|4(application)>,
"means":<alarm|event|asset|questionnaire|vulnerability>,
"rule":"<condition for the indicator>",
"mode_of_operation": <1(CyberWiserEssential)|2(CyberWiserPlus)>,
“default_value”: <default value for the indicator>
"sensor_types": [<list of sensors generating this indicator>]
}
```

- **Reset of Indicator Values**
  From the WISER dashboard, the user can mark as "resolved" risks included in the models selected for the risk assessment. When a risk is resolved, the values of the associated testing, network and application indicators are reset to their default values.

- **Definition of Rules for Indicators**
  Each indicator stored in the Indicator Catalogue includes a field called “rule” which is used by the Indicator Value Generator to determine which is the value of that indicator based on the set of conditions included.
  The definition of these rules supported by the current version of the Risk Assessment Engine is the same described in D5.2 and depends on the type of indicator. However, it is important to remark that application indicators are now also supported and with the same rules described for network indicators.
  The conditions that can be used in the rules are the same for any type of indicator. The difference is on the fields they are applied (e.g. “DST_IP” for a network or application indicator versus “W_risk_level” for a testing indicator). The following conditions are supported:
    - Equal: this can be defined with “=” to indicate a specific value to be checked
    - Not equal: this can be defined with “!=” to indicate a specific value
    - In range: this can be defined with “IN” and the range allowed between parenthesis and separated by “:”. For example: DST_PORT IN (0:1024)
    - Not in range: this can be defined with the operator “NOT IN”

  It is also possible to include several conditions joined by the words “AND” and “OR”. For example: plugin_id = (1001 OR 1002) AND plugin_sid IN (1:10)

10.2.2 TriggeringDetector

The Triggering Detector is responsible for launching the evaluation of the different risk assessment algorithms when a change in an indicator or a risk model is detected.

The main change with respect to the description provided in D5.2 is that the user can now select in the WISER Dashboard if both qualitative (DEXi) and quantitative (**R**) evaluations will be done for a risk model or just one of them (see Figure 31). This selection is considered by the Triggering Detector module to invoke the execution of the selected algorithms.

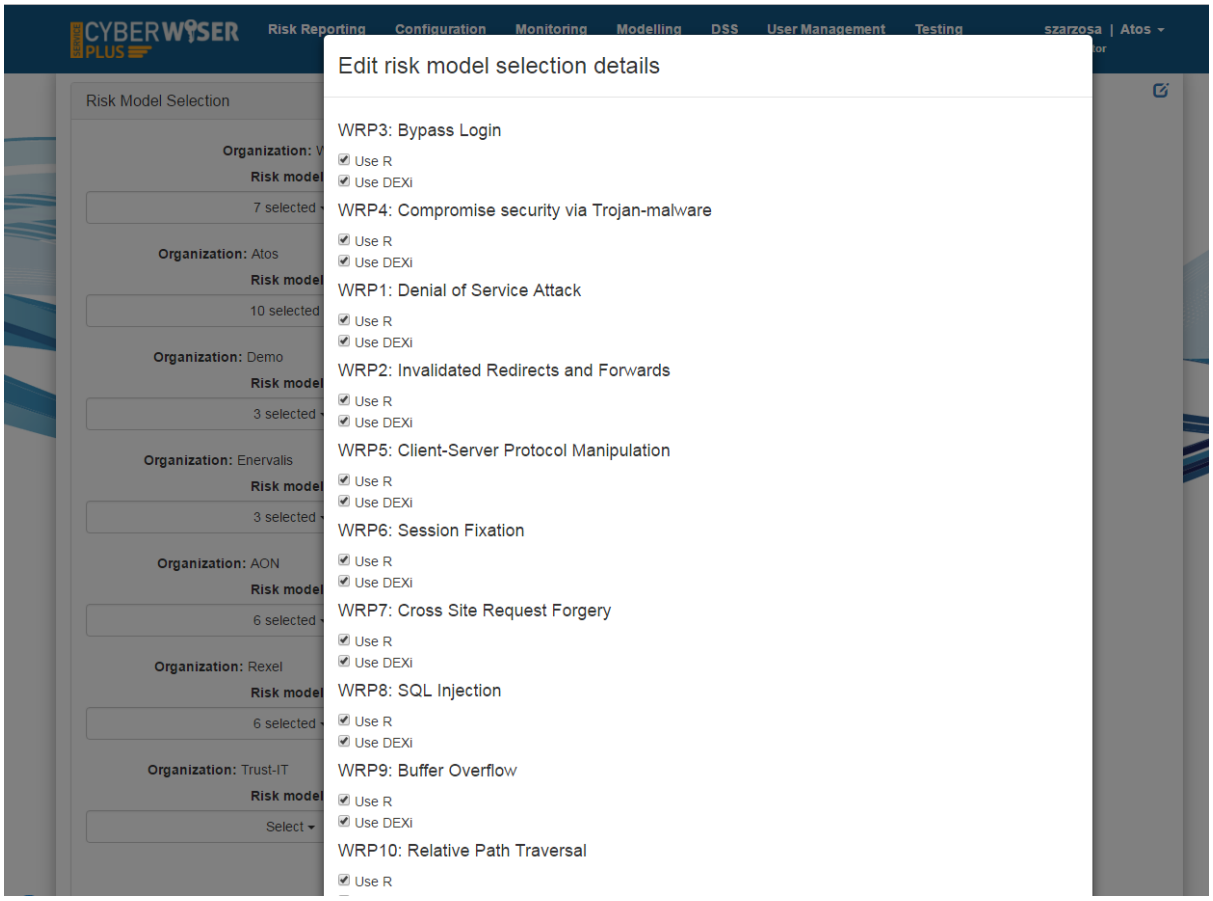


Figure 31. Risk model selection details in dashboard

10.2.3 DEXIModelInstantiator

The DEXi Model Instantiator module of the Risk Assessment Engine is invoked by the DEXi Model Rules Executor to do the instantiation of a risk model selected for a specific target.

The main improvement with respect to the description in D5.2 is the use of a new version 3.1 of the java open source library JDEXi3xEval.jar to evaluate DEXi models provided by Professor Bohanec [7]. This new library supports the use of linked attributes and consequently it is not required to instantiate the same indicator more than once even if it occurs several places in a model.

The final string prepared by the DEXi Model Instantiator to be used in the DEXi evaluation of the model is something like the following one:

```
IN-21=No;IN-20=No;IN-C31C=Very high;DummyNode=DummyValue1
```

10.2.4 DEXIModelRulesExecutor

The DEXi Model Rules Executor is the Risk Assessment Engine module in charge of the execution the qualitative risk assessment algorithm through the evaluation of the DEXi model file defined for a specific risk model and a specific target in an organization infrastructure.

In addition to the steps followed by this module described in D5.2 (Section 3.2.3), the current evaluation of the DEXi models also provides information about mitigation measures.

The execution of that DEXi java evaluator generates an output like the following one, where the attributes starting with 'M' indicates the evaluation about specific mitigation measures included in the model:

DEXi Output:

WRP6-R1=Very high;I_U1=Very high;I_S1_to_U1=High;I_S1=High;cI_S1_to_U1=High;I_S2_to_U1=Very high;I_S2=Very high;cI_S2_to_U1=Very high;c_U1_A1=Very high;M27=Yes;M28=Yes;M29=Yes;M30=Yes;M31=Yes;M32=Yes;M33=Yes;M35=Yes;M36=Yes;M37=Yes;M38=Yes;M14=Yes;M8=Yes;M34=Yes

Those mitigation measures whose DEXi evaluation output is 'Yes' will be stored in the Data Warehouse in the table /rae/risk_reports_mitigation_measures and they will be shown to the user in the CyberWISER Essential/Plus Dashboard.

10.2.5 RModelInstantiator

The main objectives of this module are first to get the indicator values from the DWH, and put them in the proper format for the **R** language that is used in the models, and second, it has to validate the models coming from the DWH before the RModelRulesExecutor runs them.

In the most recent version of the system, the Dashboard has an added functionality that allows admin users to upload their own **R** models so they can be executed by the RAE.

This functionality introduces a potential integrity problem to the overall system, along with a security hole. Since the **R** scripts can have access to the operative system functions (such as file system management, execution of arbitrary commands, download files from the internet, etc.) therefore these models have to be executed in a secure sandbox environment in order to protect the server running the RAE.

A Docker container was prepared to solve this security problem, and it is explained more in depth in Section 10.2.6.

The RModelInstantiator retrieves the corresponding custom **R** model from the DWH (if there is one available) and performs a basic analysis to see if it is invalid.

If the model coming from the DWH is found to be invalid, then the RModelInstantiator defaults to a local backup copy of the model that has been tested and proven to work.

As mentioned in the document D5.2, the execution of the **R** model is made with the Rscript binary in this form:

Rscript –e "indicator_values_as_string ; source(RmodelToExecute)"

The RAE takes a list of default indicator values from a local file to create a string that initialize the variables that are used in the **R** model which is executed with the 'source(R_File)' instruction.

Prior to the execution of the **R** model, the RAE concatenates this string with the default indicator values coming from the DWH, which will overwrite any previous duplicated variable. This is done as a failsafe measure, in the rare case there is a missing default indicator value in the DWH, so the RAE would still provide a result, and therefore, the whole system is more resilent.

So for example, if the **R** model uses the indicator "IN_9", and the local default file has that indicator assigned to FALSE, and the DWH has that indicator assigned to TRUE, the resulting string will be:

Rscript –e "IN_9 <- FALSE ; IN_9 <- TRUE ; source(RmodelToExecute)"

Therefore, the actual value the indicator will take is the last one (coming from the DWH), which takes priority from the local default. The DWH should always have a more up-to-date default indicator values, so these must overwrite the failsafe values.

If the **R** model passes the test, it will be initialized with the default indicator values coming from the DWH (or the local ones in case they are missing). It is then handed over to the RModelRulesExecutor in a similar way as presented in D5.2, with the difference that now the RModelInstantiator also includes the Consequence Indicators in the initialization which are needed for the Bayesian Network approach used by the new **R** models.

10.2.6 RModelRulesExecutor

This module has three purposes:

- Get the indicator values that are the output of the RModelInstantiator.
- Execute the **R** model script with the corresponding indicators.
- Store the result in the DWH to be used by the Aggregator module.

The introduction of the functionality to allow custom **R** models being stored and retrieved from the Dashboard, made it necessary to add the Docker container safeguard to the RAE system.

If the RModelInstantiator determines that the custom **R** model is valid, then the RModelRulesExecutor will take that custom **R** model and inject it to a specially prepared Docker container that has all the necessary libraries and commands to execute the **R** model as a sandbox environment.

If the **R** model is actually malicious, and attempts to do any harm to the system, it will only affect the container, and all the changes will be lost once the container ends its execution.

If the RModelInstantiator determines that the custom **R** model is not valid, then the RModelRulesExecutor will use a local copy of the corresponding model that has been proven to work.

The other change in this module is the adaptation to the format of the output used by the new Bayesian networks approach of the final **R** models.

The old output format was a range of values per risk, representing the minimum and maximum amount of euros per year that could be lost by the organization if it was affected by the risk. The new output format is now three numbers per risk:

- Mean frequency of the incident (mean occurrences per year)
- Typical aggregate loss per year (i.e. risk level; taking into account both frequency and consequence/severity)
  - Intuitive interpretation: With probability 50%, this value will not be exceeded
- 90th percentile of yearly aggregate loss
  - Intuitive interpretation: With probability 90%, this value will not be exceeded by the yearly aggregate loss

In addition, the mitigation measures (if any) are stored, represented by the letter M, followed by a number, and either true or false, if that mitigation number is suggested or not by the **R** model result. For example (M1: True).

In order to save space and also reduce the amount of code needed to be changed for the integration of the new Bayesian models, the RModelRulesExecutor stores the three values for each risk as a colon separated string in this form.

TypicalLoss**:**90percentile**:**FrequencyMean

As opposed to the old version of the output:

Minimum**:**Maximum

This string will be expanded by the Aggregator and do the corresponding calculations, explained in Section 10.2.7.

10.2.7 Aggregator

Aggregation of the risk assessment results is performed in the Aggregator module, both for qualitative (DEXi) and quantitative (**R**) models. The details of the aggregation procedure are different for the qualitative and quantitative models (and are reported below). However, the structure of the

aggregation levels is the same for both, as introduced in Section 3.2.7 of D5.2. We illustrate it here by means of an example, reported in Table 10, which is assumed to contain the results of a risk assessment.

There are three types of aggregation levels:

- **Aggregation by risk.** (step 1) This is obtained by aggregating the risk results corresponding to each risk in the report. In Table 10, risk assessment results for five risks are reported: WRPx1-Ry1, WRPx2-Ry2, up until WRPx5-R5. The risk assessment for Risk WRPx1-Ry1 is obtained by aggregating the results {R1,R2}; for Risk WRPx2-Ry2 we aggregate {R3, R4}, for WRPx3-Ry3 we aggregate {R5, R6} and so on.
- **Aggregation by section**. (step 2) For each section in the report, this is obtained by aggregating the risk results from Step 1. In our example in Table 10, there are two Sections. The aggregation for "Section 1: S1" is obtained by adding up the results for risks WRPx1-Ry1 and WRPx2-Ry2 obtained in Step 1. The risk for Section 2 is obtained similarly.
- **Overall aggregation for the organization.** (step 3) This is obtained by aggregating the risk results obtained in Step 2. Referring to the example of risk assessment in Table 10, the overall aggregate risk for the organization is obtained by aggregating the risk for Section 1 and the risk for Section 2.

Overall profile				Risk assessment result
	Section 1: S1			
		Risk WRPx1-Ry1		
			Target T1	R1
			Target T2	R2
		Risk WRPx2-Ry2		
			Target T1	R3
			Target T2	R4
	Section 2: S2			
		Risk WRPx3-Ry3		
			Target T1	R5
			Target T2	R6
		Risk WRPx4-Ry4		
			Target T1	R7
			Target T2	R8
		Risk WRPx5-Ry5		
			Target T1	R9
			Target T2	R10

Table 10. Example of risk assessment results for a given user.

Qualitative aggregation procedure

Let A represent the set of all qualitative risk assessment results for a given user. Given a subset S of A, we define a procedure to obtain an aggregate risk assessment for the risk results in S. First, we define a mapping between the qualitative scale and corresponding quantitative weights as defined in Table 11.

Qualitative scale	Corresponding numeric scale
Very High	15
High	9

Medium	4
Low	1
Very Low	0

Table 11. Mapping between qualitative and quantitative scales.

Intuitively, these weights can be understood as follows: one Very High risk is considered "equally bad" as 15 Low risks, while one High risk is "equally bad" as 9 Low risks, and so on. The quantitative weights have been subjectively chosen to reflect our perception of the relation between the qualitative descriptions of each step. Apart from respecting the order of the steps (meaning that the weights are increasing), there are no formal requirements that determine the "correct" weights.

Aggregation of the risk results in S is then performed in the following three steps:

1. Map each risk result in S onto its corresponding numeric value, according to the mapping in Table 11.
2. Add up all numeric values (obtained in the previous step), thereby obtaining a number N.
3. The aggregated qualitative value for the set S is obtained by selecting the highest qualitative value whose numeric value, according to the mapping in Table 11, is equal to or less than N.

Let us consider the following example. Let S={R1, R2, R3, R4}, where R1=Very low (0); R2=Medium (4); R3=Low (1); R4=High (9). This gives N=0+4+1+9=14. Since 14 is less than 15 (Very high), but at least as high as 9 (High), the aggregated value for S is High.

Unlike the approach described in D5.2, the current procedure ensures that adding a new to S risk will never reduce the aggregated risk value.

Quantitative aggregation procedure

The new quantitative algorithms produce an output for each couple of risk and target machine consisting of three values: mean frequency, typical loss and 90th percentile of the annual aggregate loss.
Since there is only one memory space available per target and per risk model for the quantitative result in the DWH table. The RAE stores these three values as a colon separated string like this:

“TypicalLoss:90Percentile:FrequencyMean”

This string is saved in the DWH, so the Aggregator can retrieve it later to perform the three levels of aggregation as explained in Section.10.2.7 and Table 10.
For each aggregation, the corresponding strings are split in the three fields again, using the colon marker as a guide, and then the values for each field are summed up, and stored in the DWH to obtain the aggregation levels.
So for example, Let Q1 and Q2 represent two quantitative results, with the corresponding strings:

Q1 = “TypicalLoss_1:90Percentile_1:FrequencyMean_1”
Q2 = “TypicalLoss_2:90Percentile_2:FrequencyMean_2”

Then, the aggregated QR = Q1 + Q2 would be:
”(TypicalLoss_1+TypicalLoss_2):(90Percentile_1+90Percentile_2):(FrequencyMean_1+FrequencyMean_2)”

This aggregated QR string is then stored in the DWH to be used by the next aggregation level.
At the end, the Dashboard take each corresponding level, so it can be displayed to the user accordingly.

11 Economic impact assessment and insurance guidelines

In this section, we provide basic guidelines on how to assess the economic impact of cyber risk and understand what part of this risk could be transferred to the insurance market.

Every organization should consider the financial statement impact of cyber exposures. In a best case scenario, a successful cyber attack will highlight the IT system weaknesses of an attacked company. In a worst case scenario, a cyber attack shuts down company operations, stolen data is sold on the dark web and financial losses are suffered by both the company and the company's customers. The key questions that every organization should answer are:

- Can we identify and quantify the damages from a successful cyber attack?
- Beyond IT solutions, how do we protect ourselves from cyber risk?

Organizations need to assure that the financial balance is protected in a cyber attack scenario: an insurance policy could be used to help achieve this. Insurance policies are designed on the basis of economic impact assessments: on the grounds of that, actuarial methods [27] can be used to study the risk reduction impact produced by different types of cover and hence decide what type of cover best fits an organization's risk appetite *("the amount and type of risk that an organisation is prepared to pursue, retain or take*", as defined by ISO/IEC Guide 73:2009).

11.1 Economic Impact Assessment guidelines

In this section we provide guidelines on how to assign values to the consequence indicators in a risk model. For a given company, assessing the economic impact of a cyber incident is a difficult problem, due to the large number of factors potentially affecting the outcome, such as:

a) the type of industry;
b) the geographical location where the company operates;
c) the company size;
d) internal policies, etc..

If a sufficient amount of economic losses data (internal and/or external) were available, the consequence indicators could be derived from quantitative analysis of such data. However, in general few companies systematically collect such data and, if they do, they tend to not disclose such data due to their sensitivity. Therefore, there is very little publicly available data regarding cyber losses at this moment. When the General Data Protection Regulation (GDPR) [29] takes effect, Data Breaches will be notified to Authority and User, and probably more data will be available.

Since a sufficient amount of data is likely to be unavailable, to obtain an economic assessment for a given company we propose *a scenario analysis approach* of the type used in operational risk management [26]. This combines the following elements:

- expert opinion from company members (management, IT); this provides information regarding the company's financial situation and main assets, internal policies, availability of qualified personnel, etc.
- expert opinion from an external consultant, non-publicly available data;
- publicly available studies and surveys, such as Ponemon's or ENISA reports [24][25]; this provides an overview of the main global drivers of the impacts, such as the industry type and geographical location, most common and most expensive attack types.

The main steps involved in an scenario analysis are:

- analysis of the company's financial situation and the main ICT infrastructure to choose the types of attacks to be analyzed and the loss types involved.
- building a questionnaire to quantify the loss types based on the information gathered in step 1.

- questionnaire interviews with the company's management and/or internal experts.
- estimating consequence parameters from the information contained in the answers.

11.1.1 Loss types and variables for economic impact assessment

This section contains a lists loss types to be considered on assessing economic impact, providing guidelines on how to obtain them through interview with the Company. The list is not intended to be exhaustive, but rather as a guide to the type of information that might influence economic impact of cyber-incidents. Each loss type is measured through one or more quantitative loss variables; the association of loss types and variables is summarised in Table 12. Loss types and Loss related variables considered for assessment of economic impact of cyber risk. We consider three different loss drivers:

1) losses related to business interruption;
2) losses related to data breaches;
3) losses related to digital assets disruption;

11.1.1.1 Losses related to Business Interruption (BI)

The Information and Communication Technology (ICT) supports business operations of a firm in very different ways, e.g. from providing ecommerce infrastructures to control room software in a power plant, hence a Business Interruption (BI) potentially have different loss effects. For example:

- loss of revenues due to business interruption, with the inability to perform business operations: inability to take orders, inability to produce goods, inability to deliver services or products, etc.;
- claims and lawsuits deriving by contractual obligations: claims due to inability to deliver products or services, claims related to breach of service level agreements, lawsuits related to breach of contracts, etc.
- fines or settlements imposed by regulators;
- reputational losses related to business interruption, in the form of economic consequences in the long term to retain or acquire new clients and in an increased cost of funding;
- recovery costs, in the form of expenses due to restore the initial situation.

Typically, the above losses are considered as having a direct impact on the firm Profit and Loss (P&L). In the case of Claims and Lawsuits these are considered in refunds and legal expenses form.

Modelling variables.

1. **Loss of revenues** due to business interruption:
   - if the company has no debts and has easy access to fresh cash: use the variable **BIMV_NE** defined as **Net Equity** (i.e., the sum Share Capital, Reserves, Net Income, and other voices) minus the **Net Profit** distributed to the shareholders or partners.
   - if the company has debts: use the variable **BIMV_EB** defined as the average of EBITDA (Earnings Before Interest, Taxes, Depreciation and Amortization) that covers the interest rates payable on the debt.

   The worst case economic loss due to a cyber event on a given target is then computed by the following formula:

$$Rev\_loss = \frac{v \cdot p \cdot N}{Y},$$

   where v represents either **BIMV_NE** or **BIMV_EB,** p is percentage of v allocated to the affected

target, N is the hypothesized number of hours of business interruption and Y is the number of hours per year.

2. **Costs for notification to authority:** from 35 to 500 pounds for each event; these values will be fixed until the GDPR will be in force [30]**.** Variable name: **Fine_Typ_Loss_Privacy_com.**

3. **Third party damages**: the cost for third Party damages. Variable name: **Third_party.**

4. **Legal costs**: the cost for Legal activities; considering the possibility for each client to file a lawsuit to the Company. Variable name: **Lg_costs.**

11.1.1.2 Losses related to digital asset disruption losses

A digital asset is intended in a broad sense, as a tangible asset involving the use of digital technology to record, process or transfer data, in order to support the business processes of a firm.

As a no exhaustive example, the following are intended as digital assets:

- The ICT infrastructure of a firm;
- A laptop used by an employee of the firm;
- An ecommerce web site;
- A USB key where client's data are recorded;
- The general ledger of a firm;
- A software running on a server.

To estimate the potential loss deriving from the disruption of digital assets and avoid double counting, we consider a digital asset disruption related loss as the cost involved with the asset reconstruction. This includes hardware cost, software cost, data recovery cost, personnel cost, forensic cost, and outsourcing cost. This loss amounts will not encompass business interruption, which is considered separately.

Modelling variables.

1. **Digital assets disruption**: monetary loss recorded by the firm, due to hardware replacement, if the hardware cannot be recovered after the cyber event. The worst economic loss is taken to be the sum of the balance sheet value of the digital assets in the Company's IT infrastructure.
   Variable name: **HW_Loss.**
2. **Data recovery costs:** monetary loss recorded by the firm, due to all actions required to recover the disrupted information. The loss is obtained by multiplying the number of records times the average cost per record breached, which we assume to be 150€ (the average cost for record according toIBM [31]. .
   Variable name: **Data_Recov_Loss.**
3. **Infrastructure recovery costs:** this cost is obtained as the sum of three variables:
   Pers_HW_Loss: monetary loss recorded by the firm, due to personnel cost, internal only, needed for hardware replacement.
   Out_Loss: monetary loss recorded by the firm, due to external personnel costs potentially involved with the disruption.
   Pers_SW_Loss: monetary loss recorded by the firm, due to personnel cost, internal only, needed for software replacement and installation.
4. **Costs for informatics experts:** cost for informatics (forensic) experts. Variable name: **For_Loss.**
5. **Losses from multimedia activities:** the cost for multimedia violation. Variable name: **MA_losses**

11.1.1.3 Losses related to Data breaches

In its operations every firm collects, elaborates and stores data of different nature. Relevant information for assessing economic impact include:

- Personal data (the information needed to identify, get in contact or locate with a physical person, like name, birth date, address, etc.);
- Sensible personal data (the information that gives clear indications about political, sexual, ethnical and religious orientations of a physical person);
- Lifestyle personal data (information that gives clear indications about the lifestyle, the spending capacity and shopping experience of a physical person, information regarding typical behaviour and/or data having the possibility to effect negatively customer experience);
- Payment system data (data used to activate payment systems such as credit card data, login and password for ebanking, etc.);
- Corporate sensible data (information giving clear indications regarding firm's clients, clients' profitability, market positioning, Merger and Acquisitions operations);
- IP data, i.e. data reporting intellectual property of the firm.

The potential loss related to a data breach depends on many factors, including the kind of data subject to breach, legislation, recovery costs.

Modelling variables.

1. **Contractual penalties:** monetary loss recorded by the firm in case of data unauthorized publication. Includes the costs for each contract for claims and compensation, mandatory disclosure costs for clients/users, excluding fines. Variable name: **Corp_Sens_Loss**

2. **Payment Card Industry Data Security Standard (PCI-DSS) costs**: standard costs for:
   a. credit card re-issuing;
   b. communication to clients;
   c. credit card network fines.

   Variable name: **Pay_Loss**

3. **Fines from regulatory authority:** cost of fines from regulatory authority (based on specific sectors, if any). Variable name: **Fine_Typ_Loss.**

4. **Reputational damage:** reputational loss related to the data breach**.** Variable name: R**ep_DB_Typ**
5. **Crisis handling costs, including brand reputation:**
   a. direct cost for extra persons that will be engaged to manage the crisis;
   b. cost of marketing campaign (for about two months).
   c. Variable name: **Rep_DB_Cat**
6. **Extortion:** cost for data extortion by ransomware. It could be evaluated from the average cost of 280€ for each PC / server infected (from Symantec ISTR Special Report on Ransomware 2016). Variable name: **Ext_pay.**
7. **Fines for privacy breach:** cost of fines for privacy breach by regulatory authority. Based on single country, from May 2018 it could be estimated as 2% of the turnover. Variable name: **Fine_Typ_Loss_Privacy.**

Business drive	Loss type	Loss related variables
Business Interruption	Loss revenues	Rev_loss
Business Interruption	Costs for notification to authority	Fine_Typ_Loss_Privacy_com
Business Interruption	Legal costs	Lg_costs
Business Interruption	Third Party damages	Third_party
Digital asset disruption	Digital assets disruption	HW_Loss
Digital asset disruption	Data recovery costs	Data_Recov_Loss
Digital asset disruption	Infrastructure recovery costs	Pers_HW_Loss , Out_Loss , Pers_SW_Loss
Digital asset disruption	Costs for informatics experts	For_Loss
Digital asset disruption	Losses from multimedia activities	MA_losses
Data breaches	Contractual penalties	Corp_Sens_Loss
Data breaches	PCI-DSS costs	Pay_Loss
Data breaches	Fines from regulatory authority	Fine_Typ_Loss
Data breaches	Reputational damage	Rep_DB_Typ
Data breaches	Crisis handling costs (including brand reputation)	Rep_DB_Cat
Data breaches	Extortion (e.g. Ransomware)	Ext_pay
Data breaches	Fines for privacy breach	Fine_Typ_Loss_Privacy

Table 12. Loss types and Loss related variables considered for assessment of economic impact of cyber risk.

11.1.2 Estimating consequence parameters

To exemplify the proposed procedure, we consider 10 types of cyber attacks identified in Deliverable D3.1. Table 13 lists the loss types involved in each attack. For example, a Denial of Service attack is assumed to involve the following loss types: Business interruption, Contractual penalties, Reputational damage, Crisis handling, Third party damages and Legal costs. Section 11.1.1 indicates the procedure to quantify each type of loss, by interviews with the company's management and internal experts. Assuming that the cost for each loss type have been quantified, for each attack type we proceed as follows:

- the worst case impact is obtained as the sum of the impacts quantified for the loss types involved in the attack as per Table 13;
- the typical case impact is obtained either by historical loss data recorded by the Company, or by publicly available datasets and surveys.

Loss type	Denial of Service	Invalidated redirects and forwards	Bypass login by brute force or DNS	Compromise security via Trojan-malware	Client-server protocol manipulation	Session hijacking	Cross site request forgery	SQL injection	Buffer overflow	Relative path traversal
Loss revenues	X			X		X	X	X	X	X
Costs for notification to authority		X	X	X	X	X	X	X	X	X
Legal costs	X	X	X	X	X	X	X	X	X	X
Third Party damages	X	X		X	X	X	X	X	X	X
Digital assets disruption				X						
Data recovery costs		X		X	X	X	X	X	X	X
Infrastructure recovery costs				X			X	X	X	X
Costs for informatics experts		X	X	X	X	X	X	X	X	X
Losses from multimedia activities				X						
Contractual penalties	X	X	X	X	X	X	X	X	X	X
PCI-DSS costs		X	X	X	X	X	X	X	X	X
Fines from regulatory authority		X	X	X	X	X	X	X	X	X
Reputational damage	X	X	X	X	X	X				
Crisis handling costs	X	X		X	X	X				
Extortion				X						
Fines for privacy breach		X	X	X	X	X	X	X	X	X

Table 13. Main types of cyber attacks (column labels) and corresponding loss types triggered by each attack (row labels).

11.2 Insurance policy guidelines

In this section, we provide a general overview of the main ingredients involved in structuring a cyber insurance policy as a risk transfer option.

Before considering risk transfer, cyber risk should be mitigated by the introduction of countermeasures such as new security technologies, process improvements, and training of personnel. For these operational aspects, the WISER platform already supports risk identification and it proposes the best solutions to implement according to a cost/benefit analysis, as further described in D5.1 (section 5.3) and D5.2 (section 5.1.3).

Risk treatment is the activity of selecting and implementing appropriate control measures to deal with the risk (ISO31000). Risk treatment includes:

- Risk control (or mitigations).
- Risk avoidance.

- Risk transfer (or risk sharing).
- Risk financing.

Transfer of risk is the underlying tenet behind insurance transactions. The purpose of this action is to take a specific risk, which is detailed in the insurance contract, and pass it from one party who does not wish to have this risk (the insured) to a party who is willing to take on the risk for a fee or premium (the insurer). The insurance market for cyber risks is relatively young: cyber risk insurance is not yet widely adopted, although this type of policies is increasingly recognised by companies as a valid tool to protect their assets. Given the complexity of the information systems supporting business processes, the lack of reliable databases to support modelling the policy premiums and the awareness that cyber risk cannot be completely eliminated, the insurance policies currently available for cyber risk do not offer standard covers which are applicable regardless of the different business sectors and the different exposures of individual companies. Therefore, the main difficulty that companies face is to identify which risks can effectively be transferred and what is a reasonable price according to a cost/benefit analysis [32].

11.2.1 Cyber risk policies

Risk transfer can be achieved through an Insurance Policy: the insured keeps part of the risk (deductible) at a premium price and the insurer covers the so-called “catastrophic risks” (as revealed by the company risk assessment, including economic impact assessment). Risk transfer through an insurance policy requires defining a coverage range to obtain maximum benefits at a reasonable premium price. A company should therefore understand which is the most appropriate threshold for transferring the residual risk to the insurance market and, at the same time, evaluate the best trade-off between the insurance coverage price and level of exposure to residual risk. Quantifying cyber risk is important to evaluate risk transfer solutions including adequate policy conditions, limitations, and exclusions at an actuarially fair rate.

The insurance market follows two approaches:

1. First Party Damages: the damage suffered by the company affected by a cyber event;
2. Third Party Damages: the insured company responsibility for violations of third party data that the insured company holds.

These two approaches involve two different indemnity methods. In the first case, the insurer indemnifies the costs needed to address the emergency crisis, meant as: costs borne by specialized IT companies for IT security, lost, encrypted or destroyed data recovery, legal costs for investigation by control authorities, profit loss linked to insured company activity interruption and, furthermore, in case, IT fraud suffered by the company and damage caused to state holder third parties.

The second approach is symmetrical and basically indemnifies damage claim of third parties because of the violation of third party data owned by the company, by adding the further costs for data recovery, damage to the insured company image, legal costs to address a damage claim or investigation, in case of effective loss of third party data outflow. In this case, it will not be indemnified, except for the profit loss suffered by the insured company, according to the relevant agreement.

The difference between the two indemnity approaches finds confirmation in two different coverage activation stages; in the first party case, the factor that triggers the coverage is the insured company damage detection, whether it is about a tangible, intangible or asset damage. In the Third party case, coverage is triggered by the damage indemnity claim by third parties as a consequence of the violation of third party data held by the insured company, for which the insured entity is responsible.

11.2.2 How to determine an Insurance policy

The main aspects of an insurance policy are:

- **Policy sections**: these are the different sections describing what type of incidents are covered by the policy.
- **Limits**: these specify the maximum amounts that will be covered for specific voices, with respect to the total amount that a policy can cover.
- **Deductibles**: this is the amount of money an individual pays for expenses before his insurance coverage starts to pay out.

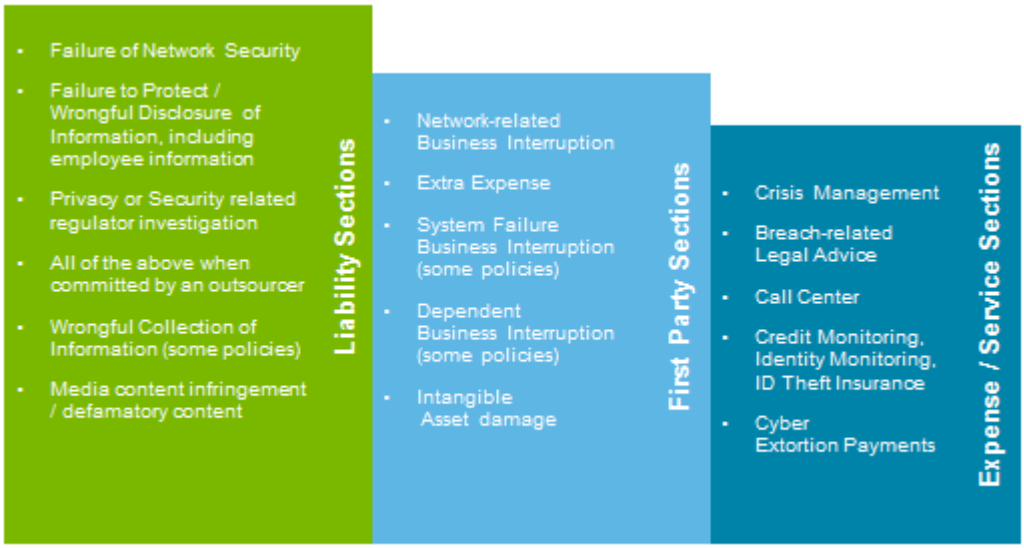
To define an insurance policy best fitting an organization's needs, the organization should answer the following questions:

- What are the Cyber Policy sections needed by the organization?
- What kind of limits they need for the different impacts?
- What kind of deductibles will be fit for the organization?

The main sections that can be offered by a Cyber insurance policy are:

- Liability Section: includes the Defense cost, the direct damage and the regulator fines (where applicable).
- First Party Section: includes the insured's loss.
- Expense / Service Section: includes the expense paid to Vendors.

For each section there are several voices that could be covered by a Cyber Policy:



The risk assessment will help the organization understand what are the right Sections of the insurance policy that could fit with the company's major risks. Based on this assumption, Table 14 helps the organization understand how WISER could help to define what type of section could fit the organization's needs.

Cyber Policy section	WRP-1	WRP-2	WRP-3	WRP-4	WRP-5	WRP-6	WRP-7	WRP-8	WRP-9	WRP-10
Network compromise	X			X						X
Breach of Data Privacy Liability		X	X	X	X	X	X	X	X	X
Digital Publication Breach			X	X	X	X	X	X	X	X
Privacy Notification Cost		X	X	X	X	X	X	X	X	X
Crisis Management Cost		X	X	X	X	X	X	X	X	X
Regulatory Actions		X	X	X	X	X	X	X	X	X
Business Interruption / Loss of Data	X	X	X	X	X	X	X	X	X	X
Cyber Extortion				X						

Table 14. Mapping of WISER risk patterns onto relevant cyber policy section.

Deductibles determine directly the premium price and they can be chosen by the organization. Based on the organization's risk appetite this value will be directly proportional to the cyber policy premium: the lower deductibles are, the higher the premium price is, and vice versa.

A cyber insurance policy will not cover the next items:

- Direct authorities fines (some countries).
- Body injuries.
- Brand reputation (instead of loss of market) [32][33].

12 Societal impact assessment

In this section, we present the WISER approach for societal impact assessment. We identify the factors to be taken into consideration, and show how these factors are structured and aggregated. This calculation is in essence qualitative, and can be done in two ways: by using a DEXi model with its corresponding qualitative decision rules, as previously presented in the document, or a Python module where numerical weights are employed, and translated back into qualitative values. Support for assessment of societal impact is offered by the DSS (Decision Support System). The model for the societal impact evaluation is offered by default in CyberWISER-Essential, and can be tailored and fine-tuned in order to better describe the company´s specific circumstances, with the help of the consulting team, in the case of CyberWISER-Light.

12.1 Preliminary considerations

The process of evaluating the societal impact of a cyber-risk event comprises a certain number of steps. It is necessary to differentiate between the calculation using a Python module and that using DEXi model:

When using a Python module, the steps are the following:

- **Step 1:** Identify qualitative criteria to evaluate the societal impact of cyber risk events, and organize them in categories.
- **Step 2:** Provide the weights to the criteria and the categories. This is necessary because not all the criteria have the same importance.
- **Step 3:** For each criterion, generate a utility function, mapping qualitative and quantitative values → (low, medium, high) to (0-10) and vice versa.
- **Step 4:** Assign qualitative values to all the criteria and generate the numerical values to the utility functions.
- **Step 5:** Generate an inverse utility function to transform the numerical result obtained back into a qualitative one.
- **Step 6:** Compare the risks in order to evaluate their effect per criteria and category.

This means that the process needs three kinds of inputs: the weights of the criteria and categories (step 2), the definition of the utility functions per criterion (step 3), the qualitative values of the criteria (step 4) and the utility function to obtain the final qualitative value (step 5). These inputs have to be introduced, and the aforementioned module will take care of performing the computation of the societal impact (a practical example is presented in Section 12.2). This risk societal impact evaluation, once calculated, will be shown to the user in the Decision Support System interface.

As commented above, in the case of CyberWISER Essential, everything will be configured by default and the user will be just shown the result of the risk societal impact evaluation in the Decision Support System interface.

As for CyberWISER Plus, the configuration can be customized with the assistance of the WISER expert team. A specific interface will be provided to introduce the inputs, once their values have been worked out.

On the other hand, when using a DEXi module, the steps to follow are slightly different:

- **Step 1:** Identify of qualitative criteria to evaluate the societal impact of cyber risk events, and organize them in categories.
- **Step 2:** Provide the weights to the criteria and the categories. This is necessary because not all the criteria have the same importance.
- **Step 3:** Establish the decision rules, and organize them into nodes forming a decision tree. These decision rules take into consideration the weights established in the previous step.
- **Step 4:** Assign qualitative values to the leaves of the tree, which correspond to the criteria
- **Step 5:** Compare the risks in order to evaluate their effect per criteria and category.

This means that inputs are provided in step 3, when setting the decision rules throughout the tree, and step 4, when assigning qualitative values to the criteria.

Again, CyberWISER-Essential offers a configuration by default and CyberWISER-Plus allows to tailor the configuration, with the help of WISER experts.

12.2 Societal impact classification

Table 15 reports the proposed classification for:

- Impacted societal asset,
- The social criteria to be considered;
- The suggested question that WISER will propose to the final user in order to explain and assess the societal impact.

Category	Criterion Id	Criterion	Question
Society	C1	Social cohesion	Does the risk entail social tensions?
	C2	Trust in fellow citizens	Does the risk harm the trust in fellow citizens?
	C3	Emotions	Does the risk provoke fear frustration, anger, etc?
	C4	Social alertness	Does the risk produce social alertness?
	C5	Job quality and labour market	Does this risk affect job quality? Does it increase the demand of jobs or the loss of jobs?
	C6	Education	Do people know about this risk?
	C7	Reputation	Does the risk influence the internal and external reputation of the victim?
	C8	Interplay with media	How will the media react to this risk?
	C9	Consumption	Does the risk influence consumption behavior?
	C10	Market & trade relations	Does the risk influence the capacity of the company to compete in the market?
Individual	C11	Motivation	How does the risk influence motivation to work, commitment, etc?
	C12	Quality of life / comfort	Does the risk affect quality of life or comfort?
Law	C13	Accountability	What is the impact of the risk from the point of view of company´s accountability?

Rights and ethics	C14	Privacy, personal data and liberty	Does the risk impact on privacy, family life, personal data protection, liberty and security of individuals?
	C15	Freedoms of thought, conscience, religion, expression, information, movement	Does the risk compromise the freedoms of thought, conscience, religion, expression, information and movement?
Politics	C16	Culture of control / authority	Could this risk entail promoting and perpetuating a 'culture of control', where authorities have an overview / power over people´s movements, bodies and actions?
	C17	Trust	Does the risk affect trust in politics?
Environment	C18	Hidden effects	Does the risk involve any chance of hidden environmental effects?
	C19	Organization	Should reactions of national and international environmental organizations be expected or considered?

Table 15. Classification of societal assets with criteria and questions

Societal Assets are categorised as follows:

- Society → This is the largest category, where the most important societal effects linked to the risk in question are recorded. WISER will consider Social cohesion, Trust in fellow citizens, Emotions, Social alertness, Job quality and labour market, Education, Reputation, Interplay with media, Consumption, Market & trade relations, as drivers capable of affecting the quality of the societal environment and society in its entirety;
- Individual → In this category WISER will record the effect the risk has on the individual as part of the society, considering Motivation and Quality of life / comfort, as the main drivers;
- Law → Company accountability is considered as the main driver for this category, where the impact on the judicial system is recorded;
- Rights and ethics → This category encompasses values considered by the European Union as fundamental for European society, Privacy, personal data and liberty, Freedoms of thought, conscience, religion, expression, information, movement;
- Politics → In this category we will consider the potential effect on politics, considering the Culture of control / authority and Trust;
- Environment→ In this category we will consider the potential effect of the risk in question on the environment, considered as a societal asset, evaluating potential environmental effects and the reaction on environmental organizations.

12.2.1 Calculation by means of Python module

As an initial weighting mechanism, in order to provide aggregated measures of the societal impact, WISER will consider the following.

Values are the results of an estimation process performed by the WISER partners, based on their experience and could potentially be revised during the course of the project. Table 16 show assignment of weights to categories and criteria.

Category	**Criterion**
Society: 50%	Social cohesion: 5%
	Trust in fellow citizens: 2%
	Emotions: 20%
	Social alertness: 2%
	Job quality and labour market: 15%
	Education: 3%
	Reputation: 20%
	Interplay with media: 10%
	Consumption: 3%
	Market & trade relations: 20%
Individual: 25%	Motivation: 60%
	Quality of life / comfort: 40%
Law: 3%	Accountability: 100%
Rights and ethics: 10%	Privacy, personal data and liberty: 60%
	Freedoms of thought, conscience, religion, expression, information, movement: 40%
Politics: 2%	Culture of control / authority: 90%
	Trust: 10%
Environment: 10%	Hidden effects: 75%
	Organizations: 25%

Table 16. Weights

In order to consider both qualitative and quali-quantitative models, utility functions are provided, as shown in Table 17.

Category	**Criterion**	**Question**	**Utility function**
Society: 50%	Social cohesion: 5%	Does the risk entail social tensions?	[L,M-L,M,M-H,H] [1,4,7,9,10]
	Trust in fellow citizens:	Does the risk harm the	[L,M-L,M,M-H,H]

	2%	trust in fellow citizens?	[1,4,7,9,10]
	Emotions: 20%	Does the risk provoke fear frustration, anger, etc?	[L,M-L,M,M-H,H] [1,3,6,8,10]
	Social alertness: 2%	Does the risk produce social alertness?	[L,M-L,M,M-H,H] [1,3,5,7,9]
	Job quality and labour market: 15%	Does this risk affect job quality? Does it increase the demand of jobs or the loss of jobs?	[L,M-L,M,M-H,H] [3,4,5,6,7]
	Education: 3%	Do people know about this risk?	[L,M-L,M,M-H,H] [0,2,6,8,10]
	Reputation: 20%	Does the risk influence the internal and external reputation of the victim?	[L,M-L,M,M-H,H] [0,2,6,8,10]
	Interplay with media: 10%	How will the media react to this risk?	[L,M-L,M,M-H,H] [0,2,6,8,10]
	Consumption: 3%	Does the risk influence consumption behavior?	[L,M-L,M,M-H,H] [2,4,5,6,8]
	Market & trade relations: 20%	Does the risk influence the capacity of the company to compete in the market?	[L,M-L,M,M-H,H] [3,5,7,9,10]
Individual: 25%	Motivation: 60%	How does the risk influence motivation to work, commitment, etc?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Quality of life / comfort: 40%	Does the risk affect quality of life or comfort?	[L,M-L,M,M-H,H] [3,5,6,8,10]
Law: 3%	Accountability: 100%	What is the impact of the risk from the point of view of company´s accountability?	[L,M-L,M,M-H,H] [3,5,7,9,10]
Rights and ethics: 10%	Privacy, personal data and liberty: 60%	Does the risk impact on privacy, family life, personal data protection, liberty and security of individuals?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Freedoms of thought, conscience, religion, expression, information, movement: 40%	Does risk compromise the freedoms of thought, conscience, religion, expression, information and movement?	[L,M-L,M,M-H,H] [3,5,7,9,10]
Politics: 2%	Culture of control / authority: 90%	Could this risk entail promoting and perpetuating a ‘culture of control’, where authorities have an	[L,M-L,M,M-H,H] [3,5,7,9,10]

		overview / power over people´s movements, bodies and actions?	
	Trust: 10%	Does the risk affect trust in politics?	[L,M-L,M,M-H,H] [2,4,6,7,10]
Environment: 10%	Hidden effects: 75%	Does the risk involve any chance of hidden environmental effects?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Organization: 25%	Should reactions of national and international environmental organizations be expected or considered?	[L,M-L,M,M-H,H] [2,4,6,7,10]

Table 17. Utility functions

Considering the structure reported above, the final user will be able to introduce its estimation of the societal impact, starting from default values.

These default values, for both the weights and the utility functions, are adopted actually in CyberWISER-Essential and, following the WISER Portfolio definition, tailoring is not considered. In the case of CyberWISER-Plus, the user will be assisted by the consulting team to fine-tune these values in order to better reflect the company´s specific circumstances.

Table 18 reports an example of evaluation of user answers, considering the application of the utility functions expressed in Table 17.

Category	**Criterion**	**Question**	**Utility function**	**Value**
Society: 50%	Social cohesion: 5%	Does the risk entail social tensions?	[L,M-L,M,M-H,H] [1,4,7,9,10]	M-L: 4
	Trust in fellow citizens: 2%	Does the risk harm the trust in fellow citizens?	[L,M-L,M,M-H,H] [1,4,7,9,10]	M: 6
	Emotions: 20%	Does the risk provoke fear frustration, anger, etc?	[L,M-L,M,M-H,H] [1,3,6,8,10]	H: 10
	Social alertness: 2%	Does the risk produce social alertness?	[L,M-L,M,M-H,H] [1,3,5,7,9]	L: 1
	Job quality and labour market: 15%	Does this risk affect job quality? Does it increase the demand of jobs or the loss of jobs?	[L,M-L,M,M-H,H] [3,4,5,6,7]	M: 5
	Education: 3%	Do people know about this risk?	[L,M-L,M,M-H,H] [0,2,6,8,10]	H: 10

	Reputation: 20%	Does the risk influence the internal and external reputation of the victim?	[L,M-L,M,M-H,H] [0,2,6,8,10]	M-H: 8
	Interplay with media: 10%	How will the media react to this risk?	[L,M-L,M,M-H,H] [0,2,6,8,10]	H: 10
	Consumption: 3%	Does the risk influence consumption behavior?	[L,M-L,M,M-H,H] [2,4,5,6,8]	M: 5
	Market & trade relations: 20%	Does the risk influence the capacity of the company to compete in the market?	[L,M-L,M,M-H,H] [3,5,7,9,10]	M: 7
Individual: 25%	Motivation: 60%	How does the risk influence motivation to work, commitment, etc?	[L,M-L,M,M-H,H] [3,5,7,9,10]	M-H: 9
	Quality of life / comfort: 40%	Does the risk affect quality of life or comfort?	[L,M-L,M,M-H,H] [3,5,6,8,10]	M-L: 5
Law: 3%	Accountability: 100%	What is the impact of the risk from the point of view of company´s accountability?	[L,M-L,M,M-H,H] [3,5,7,9,10]	M-H: 9
Rights and ethics: 10%	Privacy, personal data and liberty: 60%	Does the risk impact in privacy, family life, personal data protection, liberty and security of individuals?	[L,M-L,M,M-H,H] [3,5,7,9,10]	L: 3
	Freedoms of thought, conscience, religion, expression, information, movement: 40%	Does risk compromise the freedoms of thought, conscience, religion, expression, information and movement?	[L,M-L,M,M-H,H] [3,5,7,9,10]	L: 3
Politics: 2%	Culture of control / authority: 90%	Could this risk entail promoting and perpetuating a ‘culture of control’, where authorities have an overview / power over people´s movements, bodies	[L,M-L,M,M-H,H] [3,5,7,9,10]	L: 3

		and actions?		
	Trust: 10%	Does the risk affect trust in politics?	[L,M-L,M,M-H,H] [2,4,6,7,10]	M-L: 4
Environment: 10%	Hidden effects: 75%	Does the risk involve any chance of hidden environmental effects?	[L,M-L,M,M-H,H] [3,5,7,9,10]	L: 3
	Organization: 25%	Should reactions of national and international environmental organizations be expected or considered?	[L,M-L,M,M-H,H] [2,4,6,7,10]	L: 2

Table 18. An example of user answers and application of utility function

If we do not consider any potential override, and we chose the proposed utility function (expressed in the fourth column in Table 18), then the obtained assessment per category is the following:

- Society: 7,54
- Individual: 7,4
- Law: 9
- Rights and ethics: 3
- Politics: 3,1
- Environment: 2,75

If the weights per category are, in this example, 50%, 25%, 3% 10%, 2%, 10% respectively, then the overall impact is 6,527. Then the following inverse utility function is utilized to translate this numeric result back into a qualitative value, to be shown in the DSS dashboard.

Quantitative value	**Qualitative value**
[0,2]	Very Low
[2,4]	Low
[4,6]	Medium
[6,8]	High
[8,10]	Very High

Table 19. Inverse utility function

This will give the possibility to compare different risks as done in Table 20.

	Risk 1	Risk 2
Society	High	High
Individual	High	Very High
Law	Very High	Very High
Rights and ethics	Low	Medium
Politics	Low	Medium
Environment	Low	Low

Table 20. Comparing risks

12.2.2 Calculation by means of a DEXi model

A DEXi model was defined, based on the information provided by Table 16 and according to the following principles:

1) Whenever possible intermediate nodes with 2 descendants were constructed, with a maximum of 3 descendants.
2) It was attempted to assemble nodes which are balanced according to the distribution of the weights in the descendants (that is, the weights for categories and criteria in Table 16).
3) The lowest levels (the leaves) contain the questions and each question is connected exactly to one criterion, as specified in Table 15

The root node has two descendants:

1) A node corresponding to category “Society”; this category has weight 50% (Table 16); the subtree attached to this node was constructed balancing the weights of the 10 questions of this category (which sum up to 100%).
2) A node “Overall Impact 2”, which aggregates the other 5 categories of Table 16, which account for the remaining 50% weight; the subtree attached to this node was constructed balancing the weights of the 5 categories.

Subtree “Society” has been constructed as follows:

- Questions C2 and C4 are placed together, since they have the same weight (2%); this yields parent node “Trust and Alertness” with an overall combined weight of 4%.
- Similarly questions C6 and C9, with a weight of 3% are placed together, yielding parent node “Education and consumption” with an overall weight of 6%.
- Intermediate nodes “Trust and Alertness” and “Education and consumption” are placed together with question C1 (weight 5%), yielding node “Society 1”, with an overall weight of 15%.
- Node “Society 1” and question C5 (weight 15%) yield node “Society 2” (weight 30%).
- Questions C3 and C8 (weights 20% and 10%) yield node “Emotions and Media” (overall weight of 30%).
- Nodes “Society 2” and “Emotions and Media” yield “Society 3” (weight 60%).
- “Market and reputational” (weight 40%, from questions C10 and C7) is placed together with “Society 3”, yielding node “Society”.

Sub-tree “Overall impact 2” has been constructed as follows:

- Categories “Law” (weight 3%) and “Politics” (2%) are placed together in node “Law and Politics” (weight 5%);

- Categories “Rights and Ethics” (weight 10%) and “Environment” (10%) are placed together in node “Environment Rights and Ethics” (weight 20%);
- Node “Overall Impact 1” combines “Law and Politics” with “Environment Rights and Ethics” and has therefore an overall weight of 25%;
- Category “Individual” (weight 25%) is combined with node “Overall Impact 1” into “Overall Impact 2”, which therefore accounts for 50% category weight.

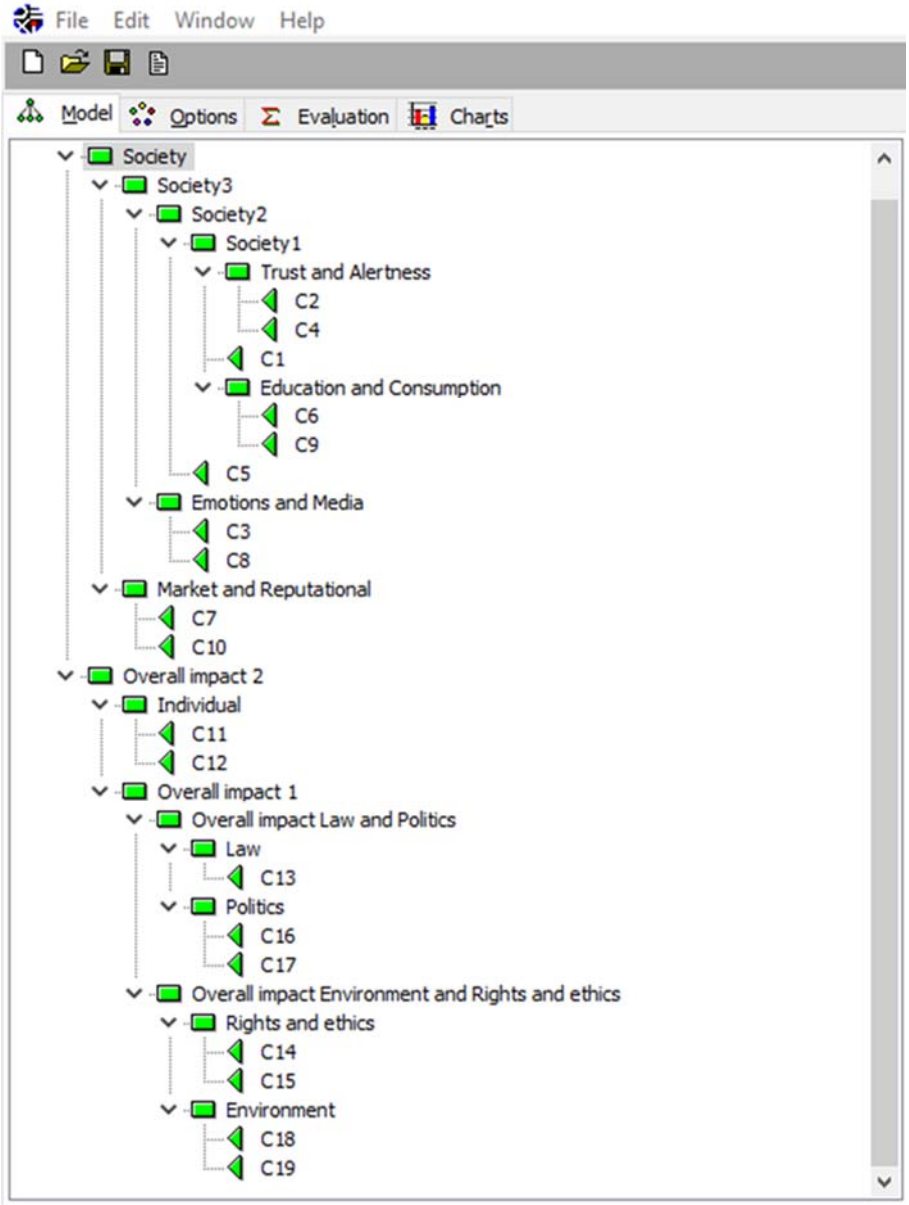
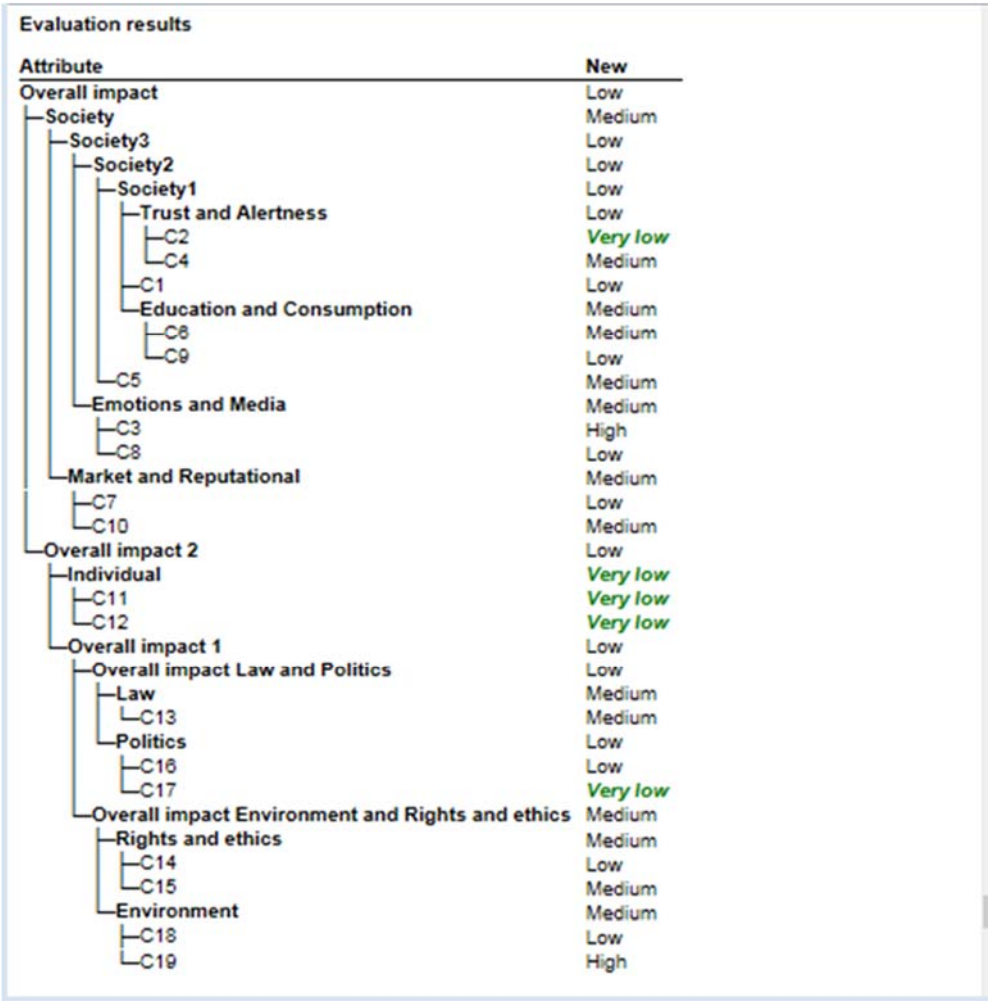


Figure 32. DEXi tree for societal impact.

A concrete example of the evaluation using the DEXi model is shown below:



Evaluation results

Attribute	New
Overall impact	Low
Society	Medium
Society3	Low
Society2	Low
Society1	Low
Trust and Alertness	Low
C2	*Very low*
C4	Medium
C1	Low
Education and Consumption	Medium
C6	Medium
C9	Low
C5	Medium
Emotions and Media	Medium
C3	High
C8	Low
Market and Reputational	Medium
C7	Low
C10	Medium
Overall impact 2	Low
Individual	*Very low*
C11	*Very low*
C12	*Very low*
Overall impact 1	Low
Overall impact Law and Politics	Low
Law	Medium
C13	Medium
Politics	Low
C16	Low
C17	*Very low*
Overall impact Environment and Rights and ethics	Medium
Rights and ethics	Medium
C14	Low
C15	Medium
Environment	Medium
C18	Low
C19	High

Figure 33. Societal impact evaluation example.

13 Conclusion

In this report, we have presented the final version of the WISER cyber risk modelling language and guidelines. We started by explaining the role of the risk modelling in the overall WISER framework. The goal of the modelling is primarily to arrive at executable algorithms to be used for continuous monitoring of cyber risk. We also motivated the choice of modelling languages. The three different languages, CORAS, DEXi and **R**, were selected and briefly presented. Each language has different qualities and serve a different purpose.

The modelling method consists of two main steps. In Step 1, an understanding of the ways in which cyber risks may materialize is established and documented. This is done using CORAS diagrams. In Step 2, an algorithm for continuous risk monitoring based on dynamic indicators is defined using either DEXi for qualitative assessments or **R** for quantitative assessments.

We have offered guidelines for developing a CORAS diagram in Step 1, as well as for defining assessment algorithms using either DEXi or **R** in Step 2 based on the CORAS diagram developed in Step 1. This is done in a modular fashion, by showing how to represent a fragment of a CORAS

diagram as a fragment of a DEXi model or an **R** script. Examples of results from following the guidelines (for **R** as well as DEXi) have been provided, as well as a description of the integration of the algorithms in the Risk Assessment Engine.

Notice that from a purely technical point of view, use of CORAS is not mandatory, as the purpose of the CORAS diagrams is to help establish and document an understanding of cyber-risk by human actors to support the definition of machine-readable algorithms. This allows users to use other approaches if they prefer, although WISER does not then provide specific guidelines.

The guidelines based on CORAS focus primarily on likelihood assessment. In addition to those guidelines, we have presented structured approaches for detailed assessment of economic and societal impact of incidents. Together with the guidelines for economic impact assessment, we offered cyber insurance policy guidelines, as cyber insurance is a general mitigation option that is closely related to economic impact considerations. The societal impact assessment combines qualitative assessments with the use of numerical weights, and is supported by the Decision Support System.

For users who wish to employ the ready-made WISER risk patterns, the target can be captured simply by selecting characteristics from a predefined list. For those who wish to develop their own risk models and algorithms for a specific target system, it is important to establish a good understanding of this target. For this purpose, we have provided simple target modelling guidelines. These guidelines are purely methodological, do not prescribe any particular modelling language, and have no technical dependencies to the WISER tools. This allows modellers to employ their own preferred target modelling language, which will typically depend on their competence and background, as well as the intended abstraction level of the cyber risk analysis.

We have applied the guidelines for risk modelling and assessment algorithm development to create quantitative and qualitative assessment algorithms for each of the 10 risk patterns documented in D3.1, which demonstrates the feasibility and utility of the approach. In order to validate these algorithms, we defined a validation method specifically tailored to the context of the WISER project. This method, which can be viewed as a refinement of the overall approach described in Section 6, is described in detail in an appendix.

14 References

[1] K. Beckers, M. Heisel, B. Solhaug, K. Stølen. ISMS-CORAS: A structured method for establishing an ISO 27001 compliant information security management system. In Engineering Secure Future Internet Services, LNCS 8431, 2014.

[2] M. Bohanec: DEXi: Program for Multi-Attribute Decision Making. User's Manual. IJS DP-11897, 2015.

[3] F. den Braber, I. Hogganvik, M. S. Lund, K. Stølen, F. Vraalsen. Model-based security analysis in seven steps – a guided tour to the CORAS method. In BT Technology Journal, Springer, 2007.

[4] The Comprehensive R Archive Network. Online: https://cran.r-project.org/ (accessed 13/5-2016).

[5] HydeNet package. Online: https://cran.r-project.org/package=HydeNet (accessed 19/02/2017).

[6] CORAS downloads. Online: http://coras.sourceforge.net/downloads.html (accessed 5/5-2016).

[7] DEXi: A Program for Multi-Attribute Decision Making. Online: http://kt.ijs.si/MarkoBohanec/dexi.html (accessed 2/5-2016).

[8] M. Neil, N. Fenton and M. Tailor. Using Bayesian Networks to model expected and unexpected operational losses.

[9] Hogganvik, I.: A Graphical Approach to Security Risk Analysis. PhD Thesis, University of Oslo, 2007.

[10] International Organization for Standardization: ISO 31000 – Risk management – Principles and Guidelines, 2009.

[11] International Organization for Standardization: ISO 27001 – Information technology – Security techniques – Information security management systems – Requirements, 2013.

[12] International Organization for Standardization: ISO 27005 – Information technology – Security techniques – Information security risk management, 2011.

[13] International Organization for Standardization: ISO 27032 – Information technology – Security techniques – Guidelines for cybersecurity, 2005.

[14] A. Lenstra and T. Voss: Information Security Risk Management, Aggregation and Mitigation. In ACISP 2004, LNCS 3108, Springer, 2004.

[15] M. S. Lund, B. Solhaug and K. Stølen: Model-Driven Risk Analysis. The CORAS Approach. Springer, 2011.

[16] MITRE: Common attack pattern enumeration and classification (CAPEC): Online: https://capec.mitre.org/ (accessed 2/5-2016).

[17] OWASP: OWASP to 10 – The ten most critical web application security risks, 2013.

[18] **R**-bloggers. Online: http://www.r-bloggers.com/ (accessed 13/5-2016).

[19] **R** documentation. Online: https://www.r-project.org/other-docs.html (accessed 13/5-2016).

[20] The **R** Project for Statistical Computing. Online: https://www.r-project.org/ (accessed 4/5-2016).

[21] A. Refsdal, B. Solhaug, K. Stølen. Cyber-Risk Management. Springer, 2015.

[22] A. Refsdal, B. Solhaug, K. Stølen. Security risk analysis of system changes exemplified within the oil and gas domain. In International Journal on Software Tools for Technology Transfer, Volume 17, Issue 3, 2015.

[23] W. N. Veneables, D.M. Smith and the **R** Core Team: An Introduction to **R**, v. 3.2.2, 2015.

[24] Global Report on the Cost of Cyber Crime, Ponemon Institute, 2015. Online: https://ssl.www8.hp.com/us/en/ssl/leadgen/secure_document.html?objid=4AA5-5207ENW (accessed 4/3-2017). Springer, 2016.

[25] The cost of incidents affecting CIIs, ENISA. Online: https://www.enisa.europa.eu/publications/the-cost-of-incidents-affecting-ciis/ (accessed 4/3-2017).

[26] Hassani, Bertrand. Scenario Analysis in Risk Management, Theory and Practice in Finance. Springer.

[27] Biener, C. and Eling, M. and Wirfs, J. H. Insurability of Cyber Risk – An Empirical Analysis. The Geneva Papers on Risk and Insurance – Issues and Practice. Vol. 40,1, pages. 131-158, 2015.

[28] ACM Computing Classification System. Online: http://www.acm.org/about/class (accessed 10/3-2017).

[29] The EU General Data Protection Regulation (GDPR). Online: http://data.consilium.europa.eu/doc/document/ST-5419-2016-REV-1/en/pdf (accessed 21/03/2017).

[30] The Information Commissioner's Office. How much does it cost? Online: **https://ico.org.uk/for-organisations/register/cost/ (accessed 21/3/2017).**

[31] http://www-03.ibm.com/security/infographics/data-breach (accessed 21/3/2017).

[32] 2015 Italian Cyber Security Report: A National Cyber Security Framework. Research Center of Cyber Intelligence and Information Security, Sapienza Università di Roma and CINI Cyber Security National Laboratory – Cyber Risk Policy. Online: http://www.cybersecurityframework.it/sites/default/files/CSR2015_ENG.pdf (accessed 21/3/2017).

[33] Cyber - The fast moving target. AON Plc.

Appendix I Naming conventions for CORAS, DEXi and R model elements

Name	Meaning
Ax, where x is an integer	Asset x
Rx, where x is an integer	Risk x
Ux, where x is an integer	Incident x ('U' stands for unwanted incident)
l_Ux, where x is an integer	Likelihood of Ux
c_Ux_Ay where x and y are integers	Consequence of Ux for Ay
cl_Sx_to_Sy	Conditional likelihood of Sx leading to Sy
cl_Sx_to_Uy	Conditional likelihood of Sx leading to Uy
l_Sx_to_Sy	Likelihood contribution from Sx to Sy
l_Sx_to_Uy	Likelihood contribution from Sx to Uy
Mx	Mitigation x

Table 21. Naming conventions for CORAS and DEXi elements

Appendix II Bayesian networks for risk modelling

As argued in [8], BNs provide an attractive solution to the problem of modelling risk:

- BNs enable an analyst to combine quantitative and qualitative information: available historical data, qualitative data and subjective judgments about the loss-generating processes.
- BNs provide a coherent framework to measure both the operational risk exposure and the effectiveness of the operational processes of a company.

Using BNs a risk manager can:

- combine proactive loss indicators, related to the business process, with reactive outcome measures such as near miss and loss data;
- incorporate expert judgments about the contribution that qualitative estimates provide to the overall risk assessment;
- enter incomplete evidence (observations) and still obtain predictions;
- perform ``*what-if?*'' scenario analysis and test sensitivity of conclusions;
- obtain a visual reasoning tool and a major documentation aid;
- obtain output in the form of verifiable predictions against actual performance measures and loss event rates.

A Bayesian network is a joint probability distribution for a set of random variables for which the set of conditional independencies can be represented using a directed acyclic graph. Each node represents a random variable, and each link represents the direction of influence between two nodes. The probability distribution for any given node is defined as conditional (only) on its parent nodes, and every node is conditionally independent from all of its non-descendants (given its parent nodes). We focus on the construction of Bayesian networks by means expert knowledge, where subject matter experts specify the structure of the network, the distributions at each node and the parameter values of the distributions.

Below we present the simplest example of Bayesian Network to model risk of losses in an operational context. The network is constructed using three nodes: the two ``parent'' nodes represent the yearly number of losses (frequency F) and the monetary amounts of each loss (severity S), whereas the '*descendant*' node represents the yearly aggregate number of losses (A).

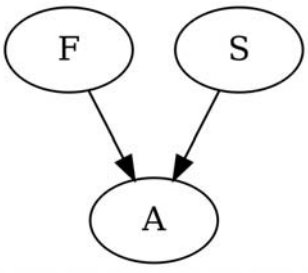


Figure 34. Basic BN model.

We choose the following distributions for the parent nodes: a uniform distribution for the number of events and a lognormal distribution for the loss amounts. The distribution of the aggregate losses is obtained by multiplying the values of the parent nodes: therefore, the node of the aggregate losses is called a deterministic node, since no further randomness is involved, beyond that of the parent nodes.

The line in the R code below illustrates the definition of the conditionality structure in the network, with node A depending on both F and S:

```
net <- HydeNetwork(~ A | F*S)
```

The next three lines define the distribution of the nodes:

- a uniform distribution in the interval $(10, 100)$ for the frequency;
- a Lognormal distribution with parameters $\mu = 10$ and $\sigma = 1.2$ for the severity (note that HydeNet uses the parametrization with the inverse of the variance, the same as the package '*rjags*');
- a distribution for the aggregate losses, obtained as a product of the frequency and the severity (the corresponding node is deterministic).

```
net <- setNode(net, F, nodeType="dunif", a=10, b=100)
net <- setNode(net, S, nodeType="dlnorm", mu=10, tau=1/1.2)
net <- setNode(net, A, nodeType="determ",
               define=fromFormula(),
               nodeFormula = A ~ F * S)
```

Table 22. **R** script for assignation of probability distributions to BN in Figure 34.

HydeNet uses the 'rjags' package to randomly simulate values from the joint distribution of the network. The next three lines launch the simulation, after specifying which variables should be monitored for the outcomes:

```
trackedVars <- c("A", "F", "S")
compNet <- compileJagsModel(net)
Posterior <- HydePosterior(compNet, variable.names = trackedVars, n.iter = 1e5)
```

Table 23. Tracked variables for the BN model in Figure 34

At the end of the simulation, the variable '`Posterior`' is a dataframe containing three columns, one for the simulated values of F, one for the simulated values of S and one for the corresponding values of A, each obtained as the product of the corresponding values of F and S.

The structure just described should be regarded as the basic building block of more complex models, where both the frequency and the severity might be influenced by other factors, through changes in the parameter values.

Appendix III Numerical results for the worked example in Section 8.3

In the example considered here, we have two numerical indicators I1 and I2, assuming integer values (Boolean indicators might occur in other risk patterns). By selecting a threshold, we reduce to four scenarios (combinations of indicator values), see Table 24.

Scenario	Value for indicator I1	Value for indicator I2
scenario 1	0	0
scenario 2	0	10
scenario 3	20	0
scenario 4	20	10

Table 24. Scenarios for BN in Figure 17.

In the CORAS model in Figure 15, indicator I1 affects the threat scenario S2; this is implemented in the model by letting the parameters of the distribution of S2 depend on the value of I1, see Table 25. The threshold of 20 is chosen here just for illustration purposes: such values are also chosen on the grounds of expert judgment and, when available, historical data. In a similar way, indicator I2 affects the parameters of the *leads-to* node cl_S2_to_S3.

For each scenario, a Monte Carlo simulation with 10^5 iterates is executed where all nodes of the network of Figure 17 assume a value: for each iteration of the simulation, the value of the initiating nodes is generated randomly, whereas the values of the deterministic nodes are calculated consequentially. Figure 35 illustrates the simulated distribution for the frequency of the unwanted incident, whereas Figure 36 illustrates the distribution of the annual aggregate loss distribution of the risk pattern under Scenario 1.

```
##parameters for the frequency of S2, based on I1
if(I1>20){
    par_l_S2 <- list(a=0.4,b=0.8)
  }else{
    par_l_S2 <- list(a=0.2,b=0.3)
  }
```

Table 25. Parameters of threat scenario node S2, depending on the values of indicator I1.

```
##parameters of cl_S2_to_S3 based on I2
  if(I2>10){
    cl_S2_to_S3 <- list(a=0.01, b=0.05)
  }else{
    cl_S2_to_S3 <- list(a=0.0, b=0.02)
  }
```

Table 26. Parameters of threat scenario node cl_S2_to_S3, depending on the values of indicator I2.

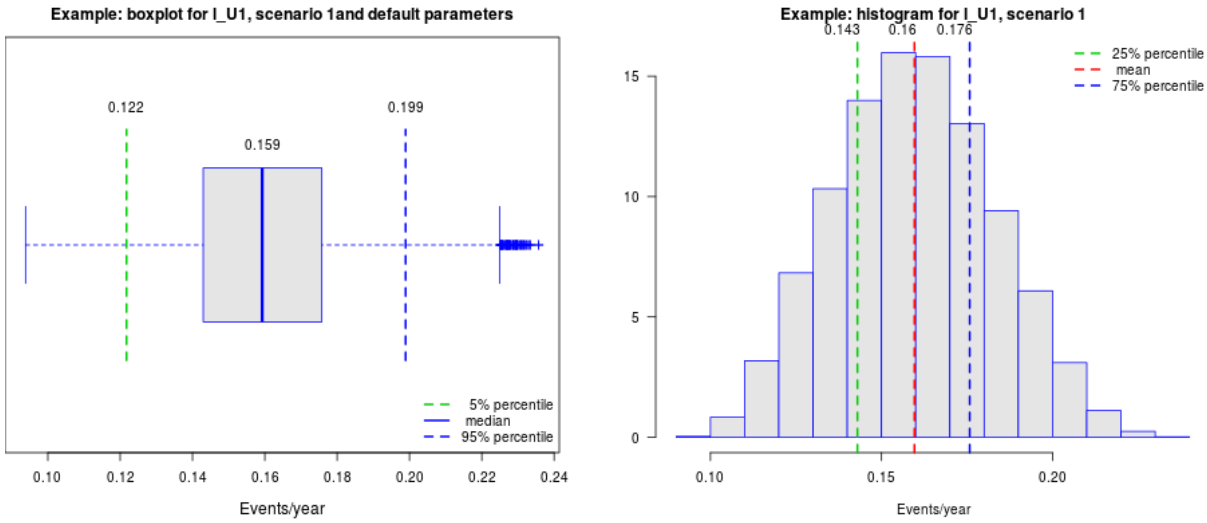


Figure 35. Boxplot and histogram for the simulated distribution of the unwanted incident U1 under Scenario 1.

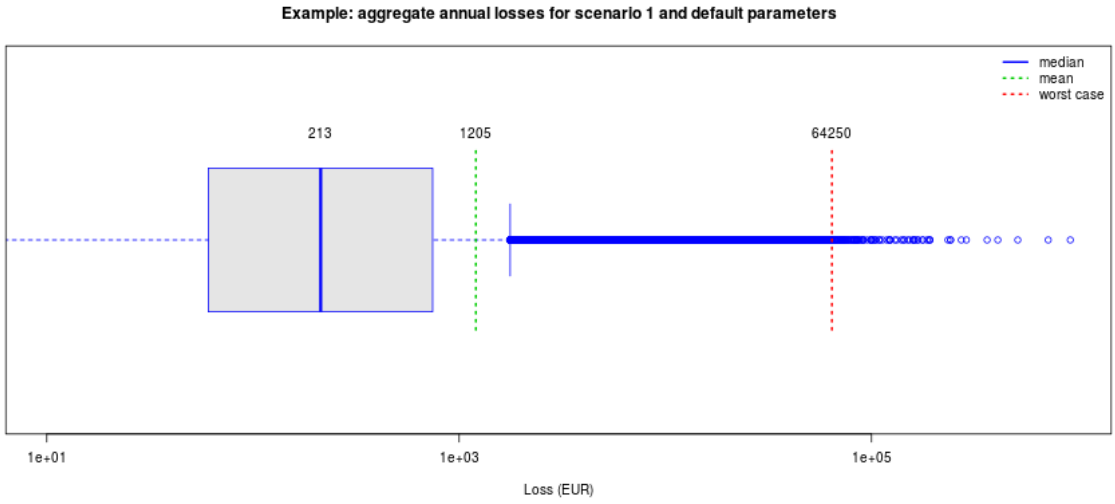


Figure 36. Boxplot of the annual aggregate loss distribution under Scenario 1.

Appendix IV Guidelines application example: DEXi model

We now present an example of a DEXi model obtained by following the guidelines provided in Section 9. The purpose here is only to illustrate the relation between a CORAS diagram and a corresponding DEXi model. The model is not intended to be used for real-life assessments, and has not been validated for this purpose.

Figure 37 shows a DEXi model based on the CORAS diagram in Figure 15. The root node R1 refers to the risk that U1 will harm A1. Notice that the label R1 does not occur in the CORAS diagram. As explained in Section 9.1.1, the risk is represented by the incident, together with its relation to the asset in the CORAS diagram.

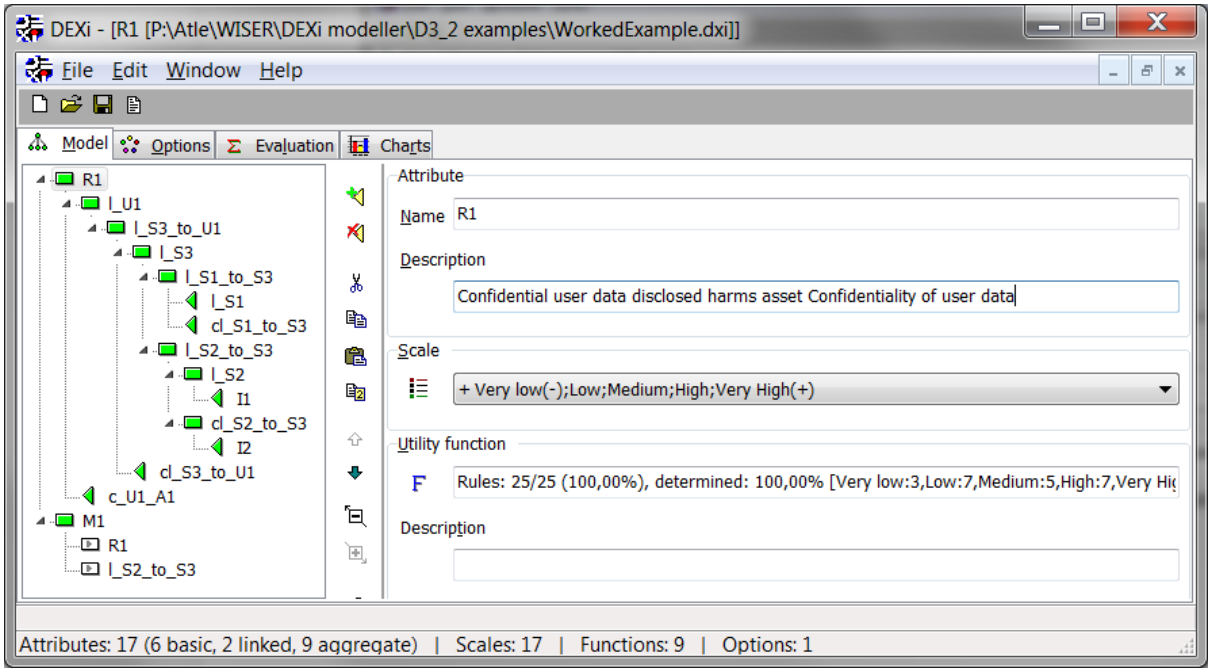


Figure 37. DEXi model obtained from following the guidelines in Section 9

As is always the case for indicators, I1 and I2 occur as leaf nodes. In addition, in this particular model, the following elements are leaf nodes: I_S1, cl_S1_to_S3, cl_S3_to_U1, and c_U1_A1. The reason is that they have no attached indicators and cannot be further broken down. This means that, unlike the indicators, their values will be fixed when instantiating the model, instead of being updated before each execution of the model.

Risk level of R1

Figure 38 shows the definition of the utility function used to determine the risk level of R1 from I_U1 and c_U1_A1. We have chosen to define it exactly like the example illustrated by Figure 20.

	I_U1	c_U1_A1	R1
1	Very low	Very low	**Very low**
2	Very low	Low	**Very low**
3	Very low	Medium	**Low**
4	Very low	High	**Low**
5	Very low	Very High	**Medium**
6	Low	Very low	**Very low**
7	Low	Low	**Low**
8	Low	Medium	**Low**
9	Low	High	**Medium**
10	Low	Very High	**High**
11	Medium	Very low	**Low**
12	Medium	Low	**Low**
13	Medium	Medium	**Medium**
14	Medium	High	**High**
15	Medium	Very High	**High**
16	High	Very low	**Low**
17	High	Low	**Medium**
18	High	Medium	**High**
19	High	High	**High**
20	High	Very High	**Very High**
21	Very High	Very low	**Medium**
22	Very High	Low	**High**
23	Very High	Medium	**High**
24	Very High	High	**Very High**
25	Very High	Very High	**Very High**

Figure 38. Utility function for R1 risk level

Since there is only one incoming path to U1, its likelihood is the same as the likelihood contribution from S3. Hence, its utility function is defined as shown in Figure 39. Of course, in this particular case, distinguishing between I_S3_to_U1 and I_U1 is not strictly necessary, and these nodes could have been merged. However, we prefer to follow consistently the structure recommended by the guidelines.

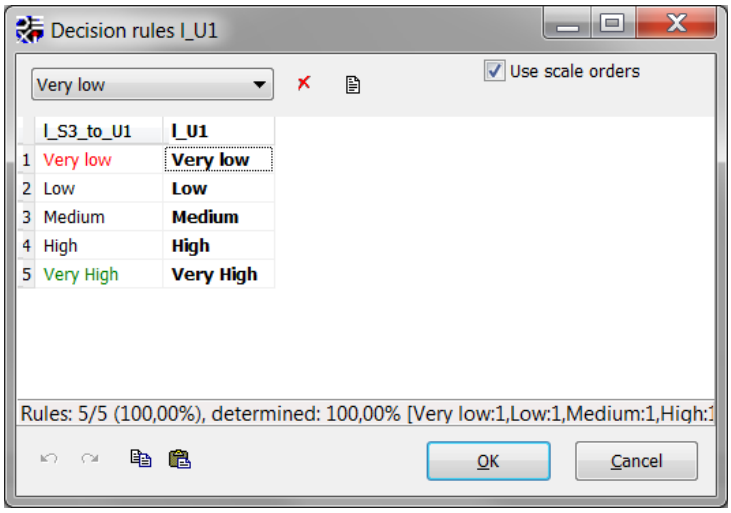


Figure 39. Utility function for I_U1

Likelihood of S3

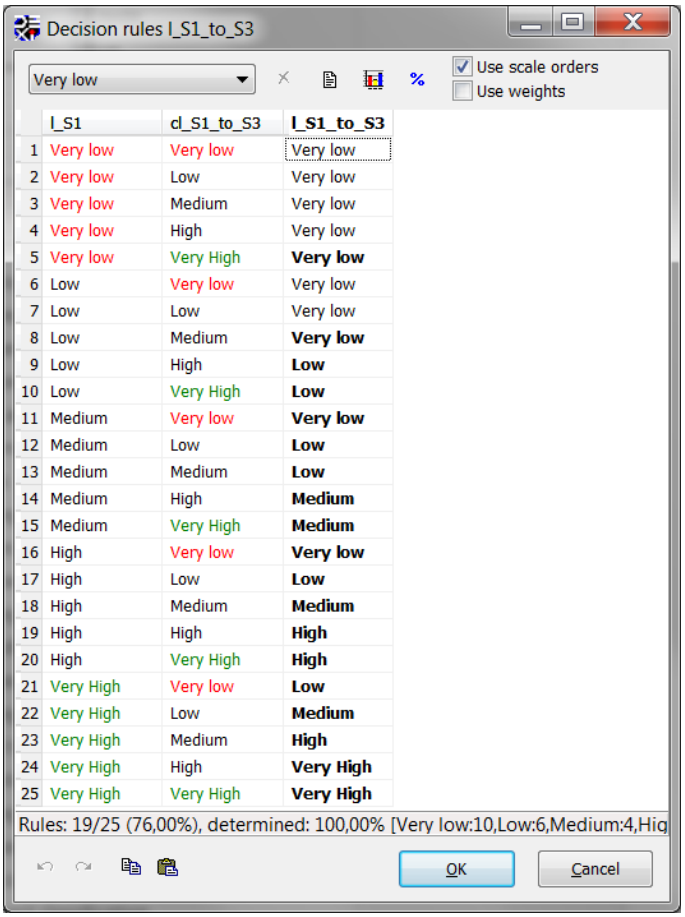
Figure 40 shows the definition of the utility function for I_S3, which depends on the combined likelihood contributions from S1 and S2. The utility function can therefore be viewed as a kind of "qualitative addition", ensuring that I_S3 is always at least as high as the highest of I_S1_to_S3 and I_S2_to_S3. The definition is also monotonically increasing in both arguments.

	I_S1_to_S3	I_S2_to_S3	I_S3
1	Very low	Very low	**Very low**
2	Very low	Low	**Low**
3	Very low	Medium	**Medium**
4	Very low	High	**High**
5	Very low	Very High	**Very High**
6	Low	Very low	**Low**
7	Low	Low	**Medium**
8	Low	Medium	**High**
9	Low	High	**Very High**
10	Low	Very High	**Very High**
11	Medium	Very low	**Medium**
12	Medium	Low	**High**
13	Medium	Medium	**High**
14	Medium	High	**Very High**
15	Medium	Very High	**Very High**
16	High	Very low	**High**
17	High	Low	**Very High**
18	High	Medium	**Very High**
19	High	High	**Very High**
20	High	Very High	**Very High**
21	Very High	Very low	**Very High**
22	Very High	Low	**Very High**
23	Very High	Medium	**Very High**
24	Very High	High	**Very High**
25	Very High	Very High	**Very High**

Figure 40. Utility function for I_S3

Likelihood contribution from S1 to S3

Figure 41 shows the definition of the utility function for I_S1_to_S3, which depends on I_S1 and cl_S1_to_S3. Since cl_S1_to_S3 represents the conditional likelihood that an occurrence of S1 will lead to an occurrence of S3, the definition ensures that I_S1_to_S3 is never higher than I_S1. The definition is also monotonically increasing in both arguments.



	I_S1	cl_S1_to_S3	I_S1_to_S3
1	Very low	Very low	Very low
2	Very low	Low	Very low
3	Very low	Medium	Very low
4	Very low	High	Very low
5	Very low	Very High	**Very low**
6	Low	Very low	Very low
7	Low	Low	Very low
8	Low	Medium	**Very low**
9	Low	High	**Low**
10	Low	Very High	**Low**
11	Medium	Very low	**Very low**
12	Medium	Low	**Low**
13	Medium	Medium	**Low**
14	Medium	High	**Medium**
15	Medium	Very High	**Medium**
16	High	Very low	**Very low**
17	High	Low	**Low**
18	High	Medium	**Medium**
19	High	High	**High**
20	High	Very High	**High**
21	Very High	Very low	**Low**
22	Very High	Low	**Medium**
23	Very High	Medium	**High**
24	Very High	High	**Very High**
25	Very High	Very High	**Very High**

Figure 41. Utility function for I_S1_to_S3

As explained above, neither I_S1 nor cl_S1_to_S3 will be further broken down, as no indicators are attached to these elements. They will therefore be assigned a constant value when instantiating the model.

Likelihood contribution from S2 to S3

The utility function of I_S2_to_S3 is identical to the one for I_S1_to_S3, so we do not include it here.

The likelihood contribution I_S2_to_S3 depends on I_S2 and cl_S2_to_S3, which depend on the indicators I1 and I2, respectively. As suggested by the indicator texts, which both starts with "How many ...etc.", they are basically integers. In order to use them in the DEXi model, we need to map their possible values to an appropriate scale. For both indicators, we have chosen to distinguish between the following three cases: a) The indicator value is 0; b) The indicator value is 1 or 2; c) The indicator value is 3 or more. Figure 42 shows the defined scale for I1. The scale is identical for I2.

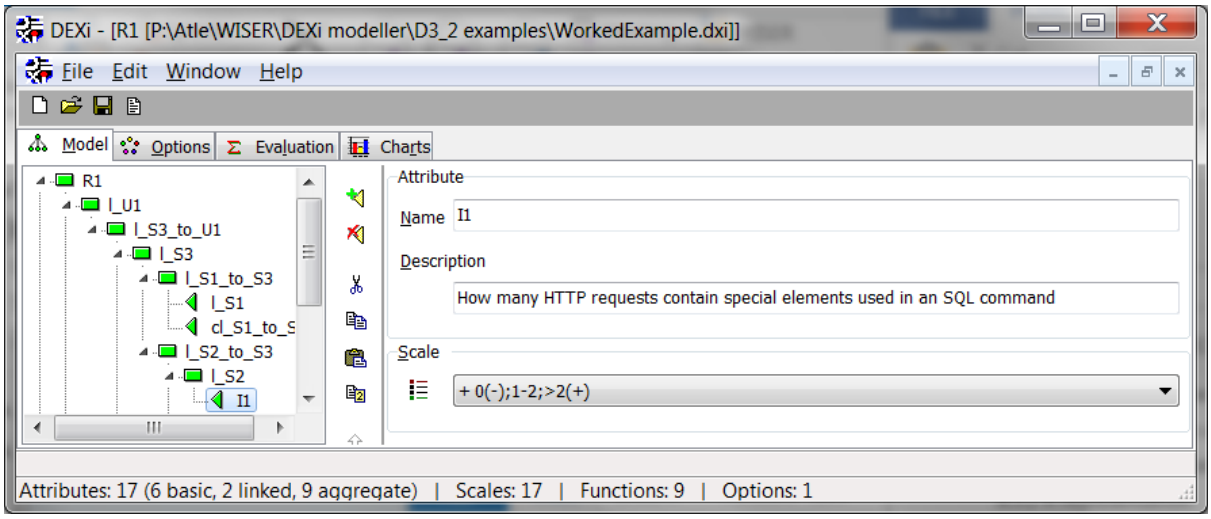


Figure 42. Definition of the scale for an indicator of type integer

Figure 43 shows the definition of the utility function for I_S2. The reasoning here is as follows: If no instances of HTTP requests with special elements used in an SQL command are observed, then we assume that the likelihood of S2 is Low, i.e. the second lowest value on our five-step likelihood scale. If one or two observations has been made, then the likelihood of S2 is High. Otherwise, if more than two such observations have been made, then the likelihood is Very high.

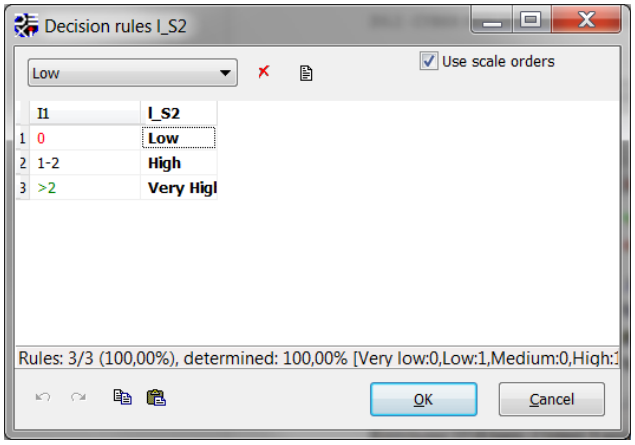


Figure 43. Utility function for I_S2

The definition of the utility function for cl_S2_to_S3 is identical to the one in Figure 43. We therefore do not include it here.

Mitigation triggering

The CORAS diagram in Figure 15 contains a single mitigation, M1, which is attached to the vulnerability on the *leads-to* relation from S2 to S3. Hence, this mitigation should be proposed if the risk R is above a given threshold and the contribution from I_S2_to_S3 is significant. As shown in Figure 37, M1 is represented by a root node in the DEXi model, with R1 and I_S2_to_S3 as sub-nodes. These sub-nodes are represented by white boxes with a small grey triangle, which means that they refer to the nodes with identical names and scales defined elsewhere in the model. Such references are called *linked attributes* in DEXi.

Figure 44 shows the definition of the utility function for M1, which is monotonically increasing in both arguments. The scale of M1 has only two steps, where Yes means that the mitigations should be proposed and No means that it should not.

	R1	I_S2_to_S3	M1
1	Very low	Very low	No
2	Very low	Low	No
3	Very low	Medium	No
4	Very low	High	No
5	Very low	Very High	No
6	Low	Very low	No
7	Low	Low	No
8	Low	Medium	No
9	Low	High	No
10	Low	Very High	No
11	Medium	Very low	No
12	Medium	Low	No
13	Medium	Medium	No
14	Medium	High	No
15	Medium	Very High	**No**
16	High	Very low	No
17	High	Low	**No**
18	High	Medium	**Yes**
19	High	High	Yes
20	High	Very High	Yes
21	Very High	Very low	**No**
22	Very High	Low	**No**
23	Very High	Medium	Yes
24	Very High	High	Yes
25	Very High	Very High	Yes

Figure 44. Utility function for M1

Appendix V Validation of the WISER risk pattern algorithms

This appendix explains the approach taken for validating the quantitative and qualitative algorithms for the ten risk patterns developed in WISER, including the expert assessments involved in defining the relation between the inputs and outputs of the algorithm. This procedure was designed specifically for the context of the WISER project, as explained in Section 6.2.2.2.

Procedure for assessment and validation

The steps of the assessment and validation process is outlined below. Step 1 is a general preparation step. Step 2 concerns the assessment and validation of the quantitative algorithms. This is the most time-consuming and demanding step, and includes identification and exploitation of empirical data sources. Notice that a separate validation document was created for performing and documenting step 2. Excerpts from the validation document for WRP6 (WISER Risk Pattern 6) are provided after the description of the process below in order to exemplify and demonstrate each sub-step. Finally, step 3 concerns the validation of the qualitative algorithms. This validation was done based on the quantitative algorithms, by comparing each qualitative likelihood assessment with its quantitative counterparts. The following describes the steps of the procedure:

1. Establish a validation team consisting of members of the WISER consortium with relevant background and competence in areas related to security and cyber risk. The role of this team is to check if the output of the algorithms is reasonable based on the available empirical data (identified in step 2a), which is done in step 2e. The validation team is documented in Table 27. Notice that not all team members took part in all validation meetings. The date and participants were documented in the validation document for each risk pattern, as illustrated by Table 28.
2. For each quantitative algorithm, do the following:
    a. Identify relevant empirical data sources. From these data sources, extract the facts of particular importance for the risk pattern in question. Table 29 gives an example of this step.
    b. Based on the above facts, provide default likelihood assessments (without considering indicator values) for all threat scenarios, unwanted incidents and leads-to relations in the risk model. We refer to these as *assessment elements*. Recall from Section 8 that all assessment elements are represented in the algorithm, following the naming convention given in Appendix I. These assessments are given in the form of frequency intervals for threat scenarios and unwanted incidents, and (conditional) probability intervals for leads-to relations. Also, assign a typical and worst case loss, in Euros per incident, to each consequence (severity) node. Table 30 gives an example of this step.
    c. For each assessment element represented as a parentless node in the BN skeleton (meaning that that its value cannot be computed from parent nodes), define a function for computing the value of the assessment element from the attached indicators. Exploit the default assessments provided in the previous step to guide this definition. Table 31, Table 32, Table 33 and Table 34 give examples of this step.
    d. Identify a set of validation scenarios for presentation to the validation team. A validation scenario is an indicator vector, i.e. an assignment of a value to each indicator. This is done because it would typically be unfeasible to validate all scenarios. For example, if we have 7 Boolean indicators, this would give 128 possible scenarios. The validation scenarios should be chosen so that they fulfil the following coverage criteria: 1) the scenarios include both the extreme cases, i.e. the indicator values indicating maximum and minimum risk; 2) the scenario include a selection of other cases that covers all paths in the CORAS diagram. By the latter, we mean that one or more indicators along the path *p* is triggered and the indicators along other paths are not triggered unless they also affect path *p*. An indicator is triggered if it has the value indicating increased risk. Table 35 to Table 39 show the definition of the validation scenarios for WRP6.
    e. For each validation scenario, run the algorithm with the indicator values defined by the scenario and provide plots of the resulting likelihood and aggregate annual loss of

the relevant risk(s) for presentation to the validation team. Make necessary modifications until the validation team agrees on the outputs for all validation scenarios. Figure 46 to Figure 55 show the plots presented during the validation of WRP6.

3. In order to validate the qualitative algorithms, do the following:
   a. Create a complete overview of all validation scenarios for all patterns. Each row in the overview should contain the following: Risk pattern number; incident; validation scenario number; median likelihood assigned to the incident by the quantitative algorithm; qualitative likelihood assigned by the quantitative algorithm. Order the overview according to the quantitative likelihood, so that the entries with highest median likelihood are placed at the top of the list. Table 40 shows this overview (after performing step 3b).
   b. Check if the ordering of the quantitative values correspond to the quantitative ordering. It this is the case, the entries on top should now have the qualitative value "Very high", followed by "High" etc. In situations where this is not the case, adjust the qualitative algorithm (typically by modifying utility functions) as needed.

Validation team member #1	
Name	Gencer Erdogan
Organization	SINTEF
Position	Research Scientist
Background, experience and main fields of expertise	Background and experience: security tester, security risk analyst, software developer, model-based testing. Main field of expertise: phd in risk-driven security testing.
Validation team member #2	
Name	Aleš Černivec
Organization	XLAB
Position	Researcher
Background, experience and main fields of expertise	Background and experience: software development, security services development, networks administration Main field of expertise: network and application security
Validation team member #3	
Name	Anže Žitnik
Organization	XLAB
Position	Researcher
Background, experience and main fields of expertise	Background and experience: software development, security services development, networks administration Main field of expertise: network and application security
Validation team member #4	
Name	Fredrik Seehusen
Organization	SINTEF
Position	Senior Researcher
Background, experience and main fields of expertise	Background and experience: PhD in computer science (2009). Researcher at SINTEF since 2009, mainly addressing cyber

	security and risk assessment. Main areas for expertise:risk assessment, security testing, and (semi) formal specification techniques and languages
Validation team member #5	
Name	Antonio Álvarez
Organization	ATOS
Position	Project manager
Background, experience and main fields of expertise	Focused on management of projects but with capacity to contribute to technical tasks: requirements elicitation, design, architecture, integration, etc. Interested in algorithms and solution implementation, with high-level knowledge on cyber security.
Validation team member #6	
Name	Paolo Lombardi
Organization	Trust-IT Services
Position	Director
Background, experience and main fields of expertise	Project management and experience in cyber risk management.
Validation team member #7	
Name	Michele Nannipieri
Organization	Trust-IT Services
Position	Project Coordination Director
Background, experience and main fields of expertise	Project management and experience in cyber risk management.
Validation team member #8	
Name	Roberto Cascella
Organization	Trust-IT Services
Position	Innovation and Research Project Manager
Background, experience and main fields of expertise	Background and experience: PhD in computer science (2007). Areas of expertise: security and privacy
Validation team member #9	
Name	Romina Colciago
Organization	AON
Position	Head of Aon AGRC (Aon Global Risk Consulting) Italy
Background, experience and main fields of expertise	Romina has extensive knowledge of Industrial and Healthcare Organizations, managing several projects concerning identification, evaluation, quantification, mitigation and definition of risk strategy to manage Cyber Risk and Operational Risks.
Validation team member #10	
Name	Alberto Luca Biasibetti

Organization	AON
Position	Senior consultant
Background, experience and main fields of expertise	Background in Information Security Management System(ISMS): supporting the definition and implementation of an information security management system in accordance with international standard ISO/IEC 27001 and ISO27002, and ICT Risk Assessment. Certifications: COBIT5 (ISACA), CSX Cybersecurity (ISACA), eJPT (eLearnSecurity)

Table 27. Algorithm validation team

Example of validation document for a quantitative algorithm

The following excerpts are taken from the validation document for WRP6 in order to illustrate the approach. The CORAS diagram the pattern is included to support the considerations and discussions (Figure 45).

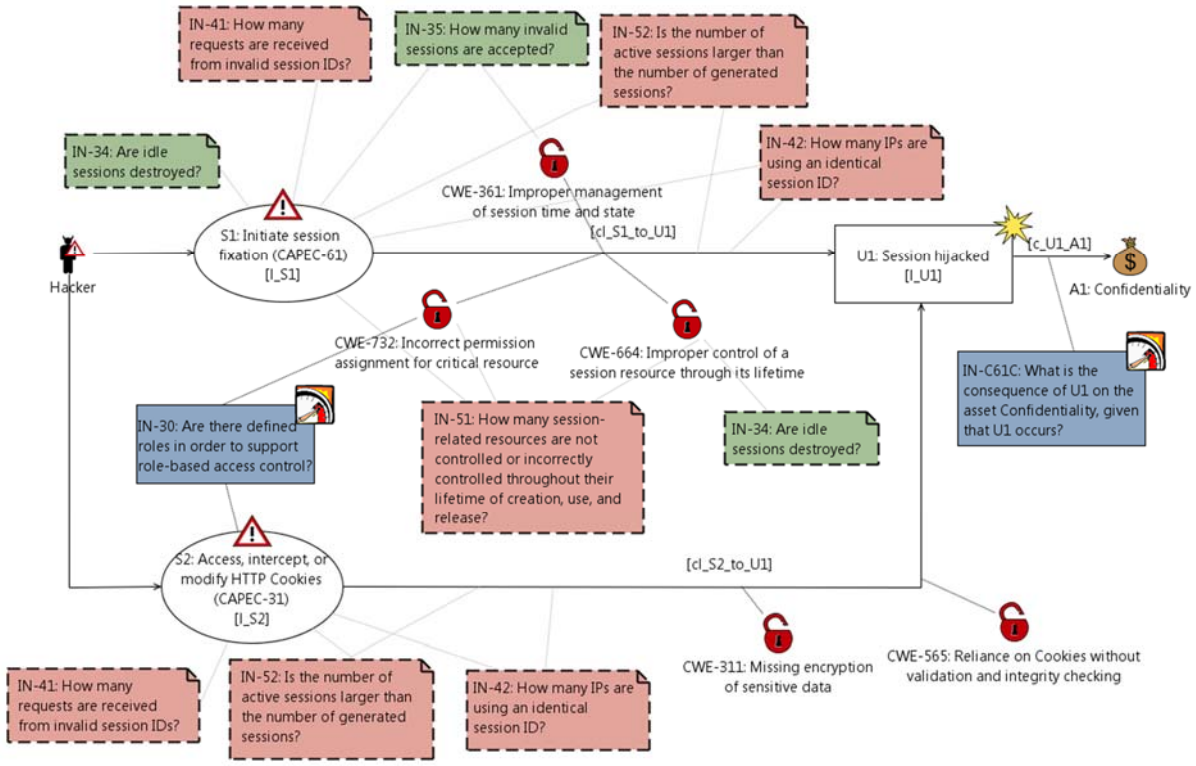


Figure 45. CORAS diagram for WRP6

The participants of the validation of WRP6 are documented in Table 28.

Date	Validation team
25.01.2017	Antonio Alvarez, Ales Cernivec, Gencer Erdogan, Alejandra Gonzalez, Fredrik Seehusen, Niccolò Zazzeri

Table 28. Participants in the validation of the quantitative algorithm for WRP6

Historical data

The historical data sources and extracted facts (relevant data) used to guide the algorithm for WRP6 are documented in Table 29.

ID	Source	Source basis	Relevant data	Refined explanation of the relevant data
F20	Symantec Internet Security Threat Report[3]	Analysis report. Symantec collects data through its Global Intelligence Network (more than 63.8 million attack sensors which monitors threat activity in over 157 countries)	Over the past three years, more than three quarters of websites scanned contained unpatched vulnerabilities, one in seven (15 percent) of which were deemed critical in 2015 [p. 26]	In 2013, 2014, and 2015, more than 75% of scanned websites contained unpatched vulnerabilities. 15% of these were deemed critical in 2015.
F21	UK Cyber Security Breaches Survey 2016[4]	Statistics collected from 1008 companies/organizations	A quarter (24%) of all businesses detected one or more cyber security breaches in the last 12 months. This is substantially higher among medium firms (51%) and large firms (65%). Large firms are also more frequently targeted, with 25 per cent of those that experienced breaches having been breached at least once a month. [p. 4]	A quarter (24%) of all businesses in the UK detected one or more cyber security breaches in 2014-2015. Medium and large firms are particularly exposed.
F26	UK Cyber Security Breaches Survey 2016[5]	Statistics collected from 1008 companies/organizations	Among companies/organizations who had any breach or attack (428 out of 1008) 13% of the attacks were related to "Access to computers, networks or services without permission and 8% of the attacks were relate to "Personal information stolen" [p. 35]	Among companies/organizations who had any breach or attack (428 out of 1008) 13% of the attacks were related to "Access to computers, networks or services without permission and 8% of the attacks were relate to "Personal information stolen"
F28	OWASP Top 10[6]	N/A	Table on page 4	According to OWASP, then ten most critical web application risks in 2013 were: A1 – Injection, A2 –

[3] Symantec Internet Security Threat Report, Volume 21. April, 2016.
[4] Rebecca Klahr, Sophie Amili, Jayesh Navin Shah. Ipsos MORI Social Research Institute. Mark Button, Victoria Wang, Institute for Criminal Justice Studies, University of Portsmouth. Cyber Security Breaches Survey 2016 Main report. 2016.
[5] Same as footnote 4.
[6] The Open Web Application Security Project, OWAS Top 10, The Ten Most Critical Web Application Security Risks, 2013

				Broken Authentication and Session Management, A3 – Cross-Site Scripting (XSS), A4 – Insecure Direct Object References, A5 – Security Misconfiguration, A6 – Sensitive Data Exposure, A7 – Missing Function Level Access Control, A8 – Cross-Site Request Forgery (CRSF), A9 – Using Known Vulnerable Components, A10 – Unvalidated Redirects and Forwards
C1	UK Cyber Security Breaches Survey 2016[7]	Statistics collected from 428 companies/organizations in the UK that experienced a cyber breach or attack in 2014-2015	Table 5.3 on page 41.	The cost of all cyber security breaches experienced during 12 months in 2014-2015 for UK organizations that experienced cyber breaches was as follows: Micro/small businesses (base 107): £3.100 (mean), £200 (median). Medium businesses (base 173): £1860 (mean), £180 (median). Large businesses (base 126): £36500 (mean), £1300 (median).
C6	Kaspersky lab[8]	Statistics collected from 5500 companies in 26 countries	Graphs on page 5.	The cost of recovering from cyber incidents related to Malware/Viruses is on average (in dollars): Large businesses: ~4700000 (direct cost), ~600000 (indirect cost). SMBs: ~25.000 (direct cost), ~7.000 (indirect cost)

Table 29. Historical data sources used to guide the algorithm for WRP6

[7] Same as footnote 4.
[8] Kaspersky Lab, Damage Control: The cost of security breaches IT security risks special report, 2015.

Reasoning behind the quantitative algorithm

In Table 30, we estimate the likelihood values of the risk model without considering the indicator values. Note that the frequency values (for the nodes) are given as intervals of occurrences per year, whereas the conditional likelihoods are given as probability intervals.

Name	Value	Description	Rationale
l_S1	[0, 25]	The likelihood that session fixation attack will be initiated	- Due to F28, we know that Broken Authentication and Session Management is the rated as the second most critical web-application risk by OWASP. Session fixation is a common threat scenario w.r.t. to this risk. The attack is well known. We do now know whether the attack itself can be automated, but there are automated tools for checking whether a web-application is vulnerable to this kind of attack. We therefore believe that the attack may be fairly common against web-pages that may seem vulnerable to the attack, but less common otherwise. However, we believe that the attack is not extremely common, since it likely that the attack has to be tailored to the web-application (not completely automated).
l_S2	[12, 52]	The likelihood that "Access, intercept, or modify HTTP Cookies" will be initiated	- Due to F28, we know that Broken Authentication and Session Management is the rated as the second most critical web-application risk by OWASP. Session fixation is a common threat scenario w.r.t. to this risk. The attack is well known, but its only relevant in this pattern the web-application used cookies to manage sessions and authentication. Like the session fixation attack, there are automated tools for checking whether a web-application is vulnerable to this kind of attack. However, we assume that this attack is easier to automate. We therefore believe that the attack frequency likes about 1 time each month to 1 time each week.
cl_S1_to_U1	[0.0, 0.0001]	The conditional likelihood that session fixation will lead to Session hijacking	We don't have any data sources on this, but we believe that the likelihood should be low since the session hijacking attack is well known, and it's reasonable to assume that most systems are protected against it. The maximum conditional likelihood must be around 0.0001 to ensure consistency with the estimate for l_S2 and l_U1.
cl_S2_to_U1	[0.0, 0.0001]	The likelihood that "Access, intercept, or modify HTTP Cookies" will lead to Session hijacking	We don't have any data sources on this, but we believe that the likelihood should be low since the attack described by threat scenario S2 is well known, and it's reasonable to assume that most systems are protected against it. The maximum conditional likelihood must be around 0.0001 to ensure consistency with the estimate for l_S2 and l_U1.
l_U1	[0, 0.005]	The frequency of the	Due to F26, we know that of the organizations

		unwanted incident "Session hijacked"	in the UK that had a cyber breach or attack: a) 13% of the cases were related to "Access to computers, networks or services without permission" and b) 8% of the cases were related to "Personal information stolen" In addition, we know that 242 organizations were breached. Assuming that the ratios are the same for "breached or attacked" and "attacked", then, we have 242 *(0.13 + 0.8) = 51 of breached is related to a) and b). This is an upper bound on the number of breaches per year due a) and b). However, not all of these breaches as caused by session hijacking. We assume that at most 10% of the 51 breaches is related to session hijacking, i.e. 51 * 0.1 = 5. We therefore estimate [0, 5/1008] = [0, 0.005]
c_U1_A1	Typical case (median): 725EUR per incident Worst case: 217500 EUR per incident	Default consequence that the unwanted incident U1 has on the asset Confidentiality	We believe that this consequence is similar to the consequence of the unwanted incident "hacker gains privileges/assumes identity" described in pattern WRP-3. There the typical case was estimated to 725 EUR per incident, and the worst case was estimated to 217500 EUR per incident.

Table 30. Estimates of likelihood values without considering indicators

The calculation functions for estimating likelihood values of the risk model taking also the indicator values into account are documented in Table 31 (l_S1), Table 32 (l_S2), Table 33 (cl_S1_to_U1) and Table 34 (cl_S2_to_U1). All values are either true (T) or false (F). In all tables below, T for the IN-35, IN-41, IN-42, or IN-51 means that the indicator is greater than zero, and F means that it is zero. The wildcard (*) stands for either T or F. In the cases where there is and overlap between the set of indicator value vectors represented by two rows (like for instance the first and second row of Table 31), the uppermost row takes precedence.

ID	IN-34	IN-41	IN-35	IN-51	IN-52	IN-42	l_S1	Rationale
I0	F	T	T	T	T	T	[15,30]	In this case, there is a very strong possibility that a successful attack has occurred or is occurring.
I1	*	*	*	*	*	T	[12,25]	In this case, there is a strong possibility that a successful attack has occurred or is occurring.
I2	*	*	*	*	T	*	[12,25]	In this case, there is a strong possibility that a successful attack has occurred or is

								occurring.
I3	*	*	T	*	*	*	[12,25]	IN-35 is true means that there is a strong possibility than a successful attack is occurring or has occurred.
I4	*	T	*	*	*	*	[5,25]	If IN-41 is true, then there is a fairly strong possibility that an attack has occurred.
I5	*	*	*	T	*	*	[4,20]	If IN-51 is true, then this may indicate that a successful attack has occurred.
I7	F	*	*	*	*	*	[4,20]	If IN-34 is false, then this may increase the chance than an attack will occur.
I6	T	F	F	F	F	F	[0,15]	If all indicators are false, it is unlikely that an attack is occurring or has occurred within the timespan observed by monitoring. However, this does not mean that the attack will not happen in the future.

Table 31. Calculation function for I_S1

In Table 32, IN-30 is ignored since it is unclear how it will affect the frequency.

ID	IN-30	IN-41	IN-42	IN-52	I_S2	Rationale
I1	*	*	*	T	[12,52]	In this case, there is a strong possibility that a successful attack has occurred or is occurring.
I2	*	*	T	*	[12,52]	In this case, there is a strong possibility that a successful attack has occurred or is occurring.
I3	*	T	*	*	[5,52]	If IN-41 is true, then there is a fairly strong possibility that an attack has occurred.
I4	F	F	F	F	[0, 35]	In this case, it may be unlikely that an attack is occurring, but it's still possible that an attack will occur in the future since the attack described by S1 is fairly common.

Table 32. Calculation function for I_S2

ID	IN-34	IN-30	IN-51	IN-35	IN-42	IN-52	cl_s1_to_U1	Rationale
I0	F	F	T	T	T	T	[0.01, 0.05]	In this case, there is a very strong possibility than a successful attack is occurring or has

								occurred.
I1	*	*	*	*	*	T	[0.01, 0.04]	In this case, there is a strong possibility than a successful attack is occurring or has occurred.
I2	*	*	*	*	T	*	[0.01, 0.04]	In this case, there is a strong possibility than a successful attack is occurring or has occurred.
I3	*	*	*	T	*	*	[0.01, 0.04]	IN-35 is true means that there is a strong possibility than a successful attack is occurring or has occurred.
I4	*	*	T	*	*	*	[0.005, 0.02]	If IN-51 is true, then this may indicate that a successful attack has occurred.
I5	F	*	*	*	*	*	[0.0, 0.02]	If IN-34 is triggered, then this may make the application more vulnerable to session fixation attacks.
I7	*	F	*	*	*	*	[0.0, 0.0001]	If IN-30 is triggered, then there is no role-based access control, making it slightly more likely that the attack will succeed then if IN-30 is true.
I6	T	T	F	F	F	F	[0.0, 0.00002]	If no indicators are triggered, then it is unlikely that a session fixation attack will succeed, since it is a well-known attack and most web-applications has protection against it.

Table 33. Calculation function for cl_S1_to_U1

ID	IN-42	IN-52	cl_s2_to_U1	Rationale
I1	*	T	[0.01, 0.05]	In this case, there is a strong possibility than a successful attack is occurring or has occurred.
I2	T	*	[0.01, 0.05]	In this case, there is a strong possibility than a successful attack is occurring or has occurred.
I3	F	F	[0.0, 0.00002]	If all indicators are false, then it is unlikely that a cookie manipulation attack will succeed, since it is a well-known attack and most web-applications has protection against it.

Table 34. Calculation function for cl_S2_to_U1

Validation scenarios

In the following, we show the definition of the validation scenarios and corresponding algorithm outputs presented in the validation meeting for WRP6. Five different validation scenarios were used for this risk pattern.

Table 35 defines validation scenario 1, which represents the minimum risk scenario. The likelihood output for this scenario is shown in Figure 46, while the aggregate annual loss output is shown in Figure 47. Notice that for this scenario, the aggregate loss is so low that median, mean and 90% percentile values falls outside the diagram in Figure 47 (to the left), due to the low likelihood of the incident.

IN-30	IN-34	IN-52	IN-35	IN-41	IN-42	IN-51
T	T	F	F	F	F	F

Table 35. Definition of validation scenario 1

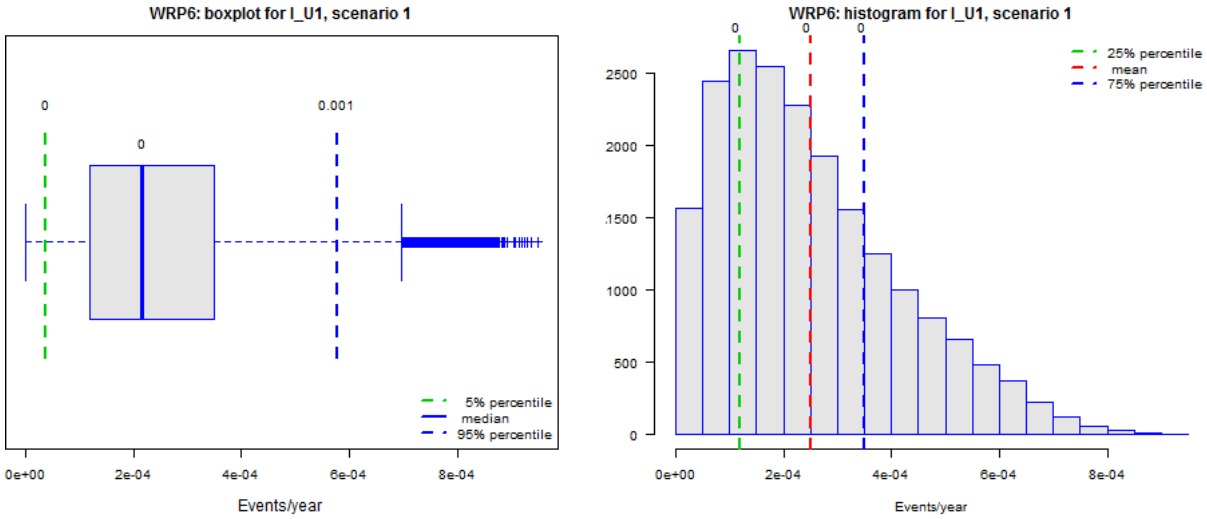


Figure 46. Likelihood output for validation scenario 1.

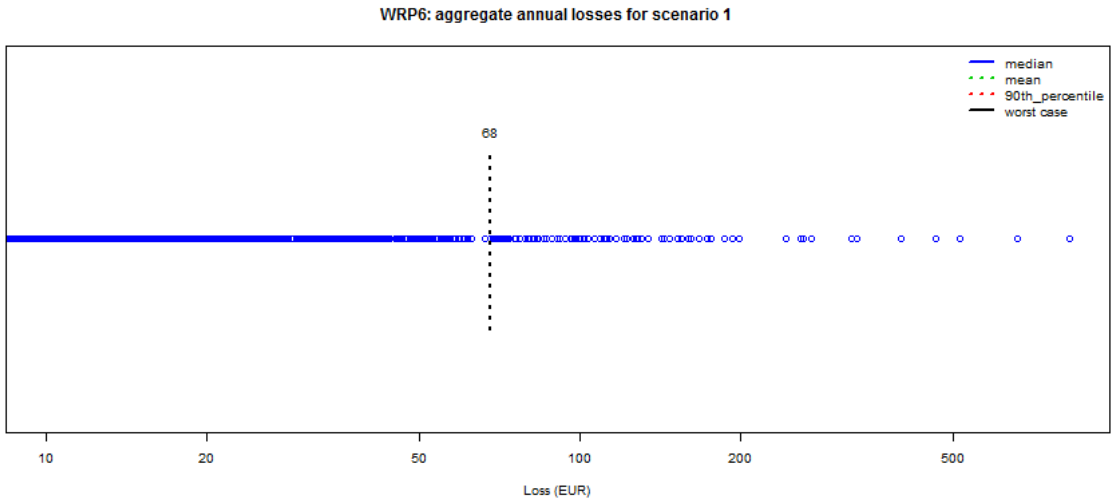


Figure 47. Aggregate annual loss output for validation scenario 1.

Table 36 defines validation scenario 2. Here, indicators on top path are triggered except for IN-42 and IN-52. The likelihood output for this scenario is shown in Figure 48, while the aggregate annual loss output is shown in Figure 49.

IN-30	IN-34	IN-52	IN-35	IN-41	IN-42	IN-51
F	F	F	T	T	F	T

Table 36. Definition of validation scenario 2

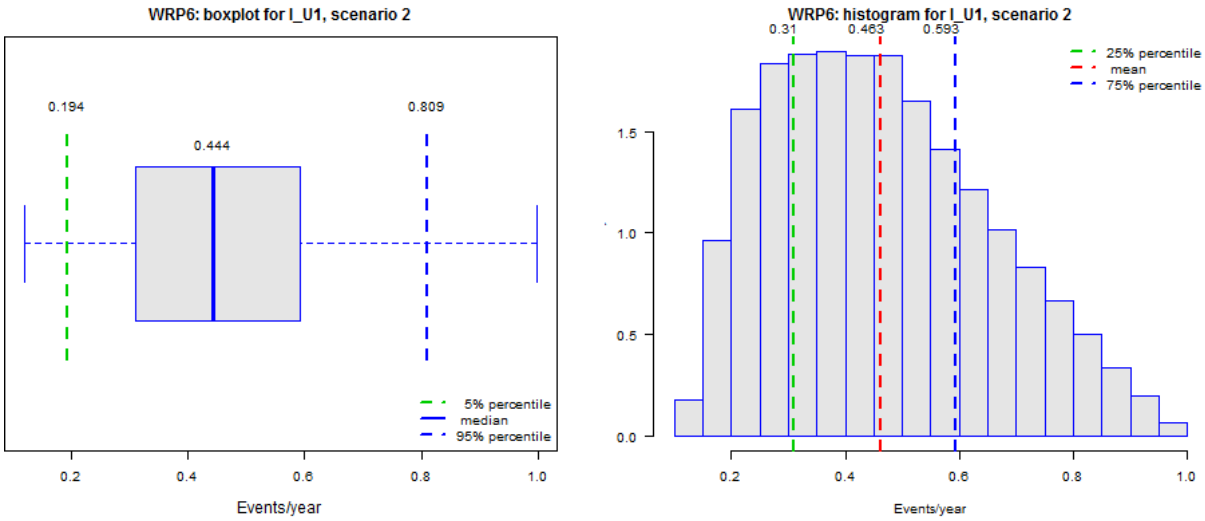


Figure 48. Likelihood output for validation scenario 2

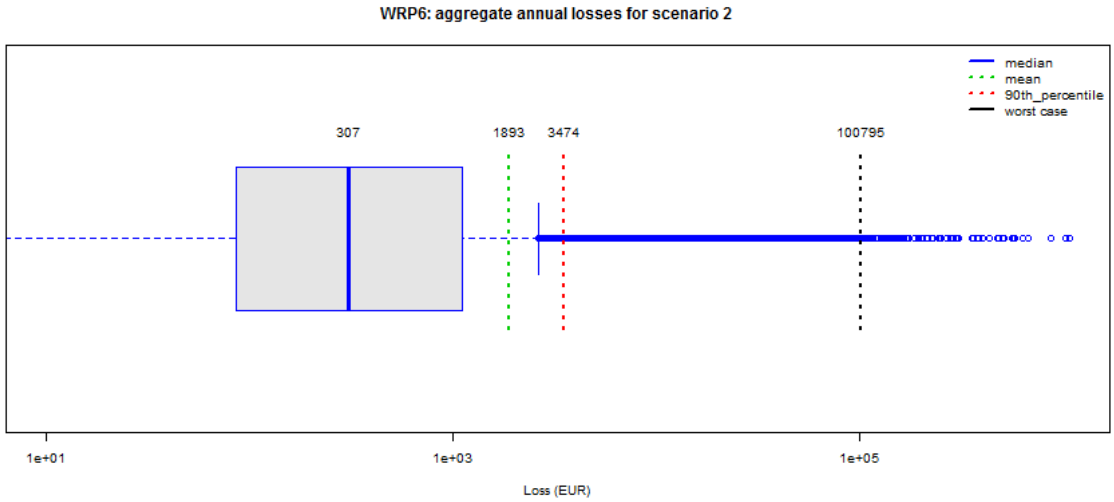


Figure 49. Aggregate annual loss output for validation scenario 2

Table 37 defines validation scenario 3. Here, indicators on bottom path are triggered except for IN-42 and IN-52. The likelihood output for this scenario is shown in Figure 50, while the aggregate annual loss output is shown in Figure 51.

IN-30	IN-34	IN-52	IN-35	IN-41	IN-42	IN-51
F	T	F	F	T	F	F

Table 37. Definition of validation scenario 3

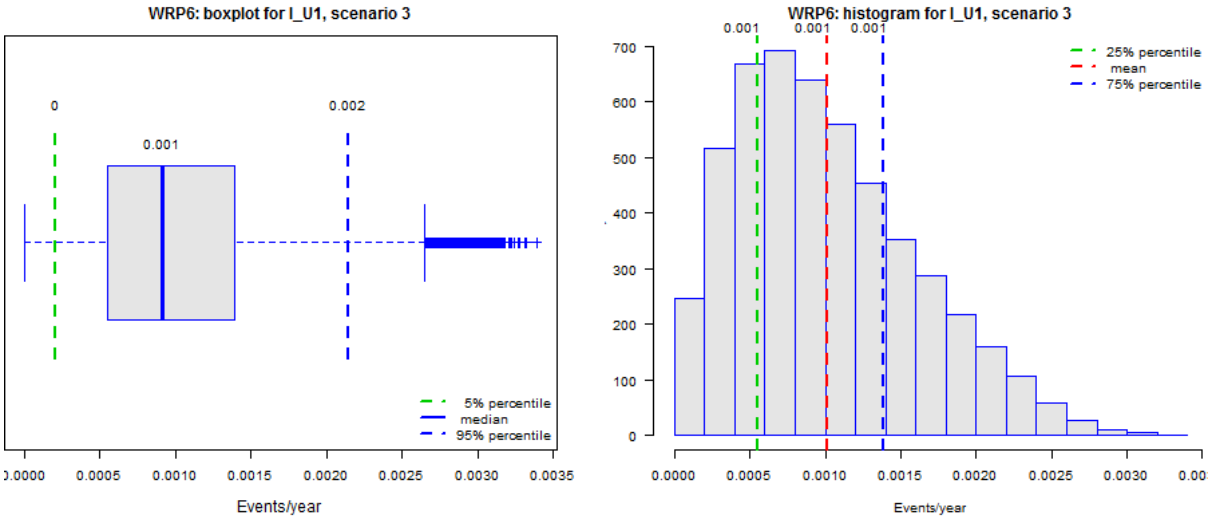


Figure 50. Likelihood output for validation scenario 3

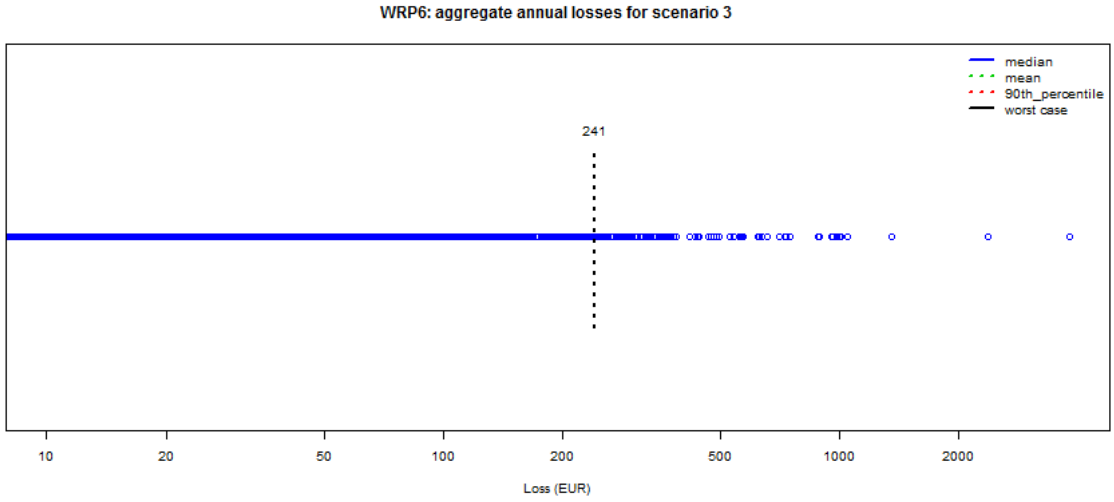


Figure 51 Aggregate annual loss output for validation scenario 3

Table 38 defines validation scenario 4. Here, indicators IN-42 and IN-52 are triggered. The likelihood output for this scenario is shown in Figure 52, while the aggregate annual loss output is shown in Figure 53.

IN-30	IN-34	IN-52	IN-35	IN-41	IN-42	IN-51
T	T	T	F	F	T	F

Table 38. Definition of validation scenario 4

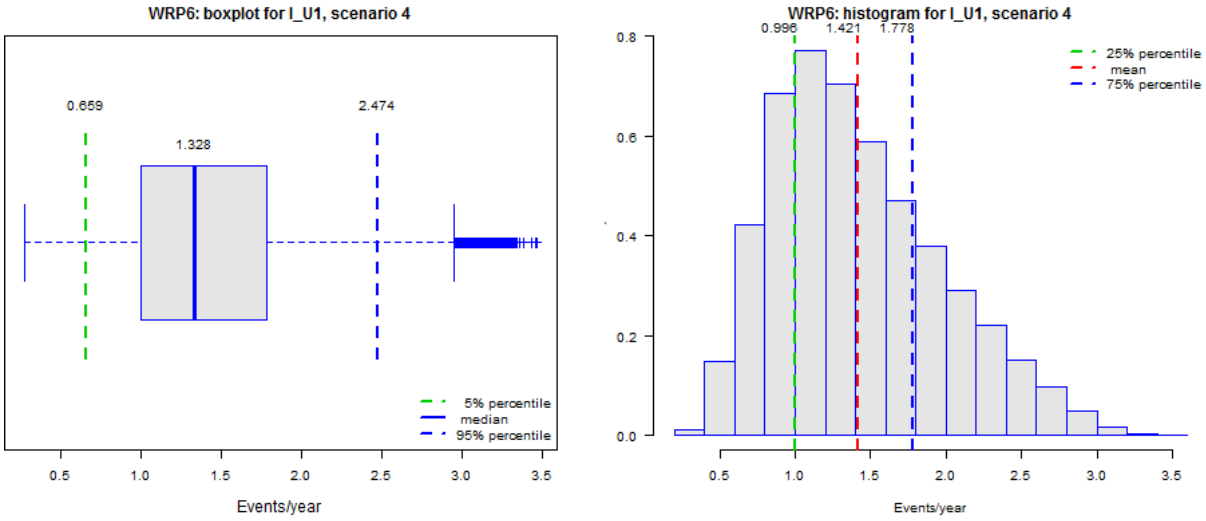


Figure 52. Likelihood output for validation scenario 4

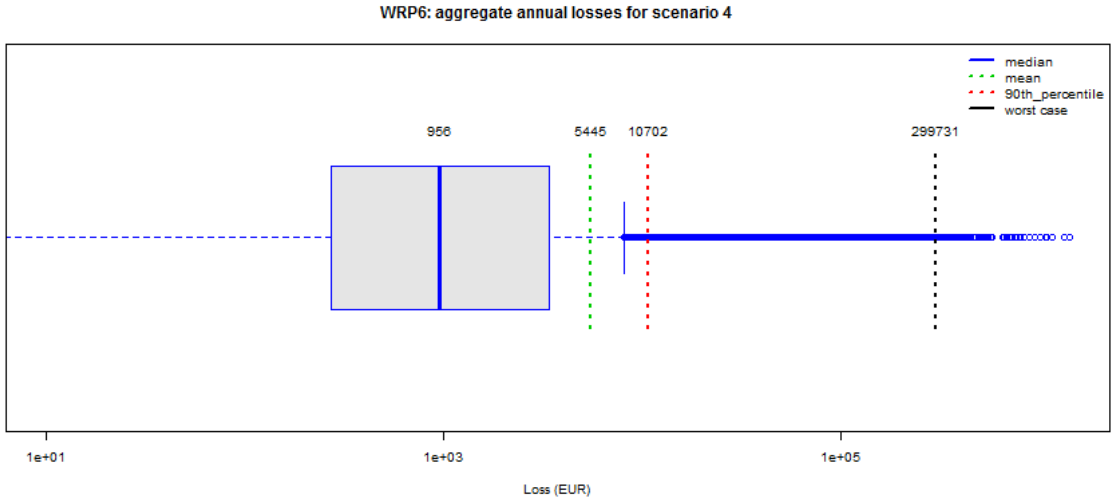


Figure 53 Aggregate annual loss output for validation scenario 4

Table 39 defines validation scenario 5. Here, all indicators are triggered, meaning that this is the maximum risk scenario. The likelihood output for this scenario is shown in Figure 54, while the aggregate annual loss output is shown in Figure 55.

IN-30	IN-34	IN-52	IN-35	IN-41	IN-42	IN-51
F	F	T	T	T	T	T

Table 39. Definition of validation scenario 5

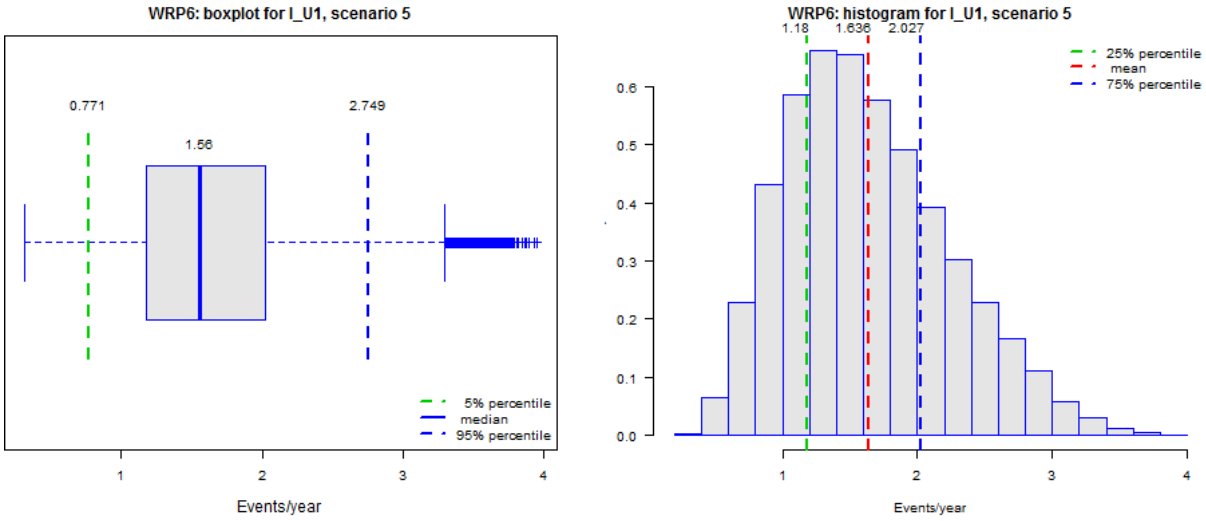


Figure 54. Likelihood output for validation scenario 5

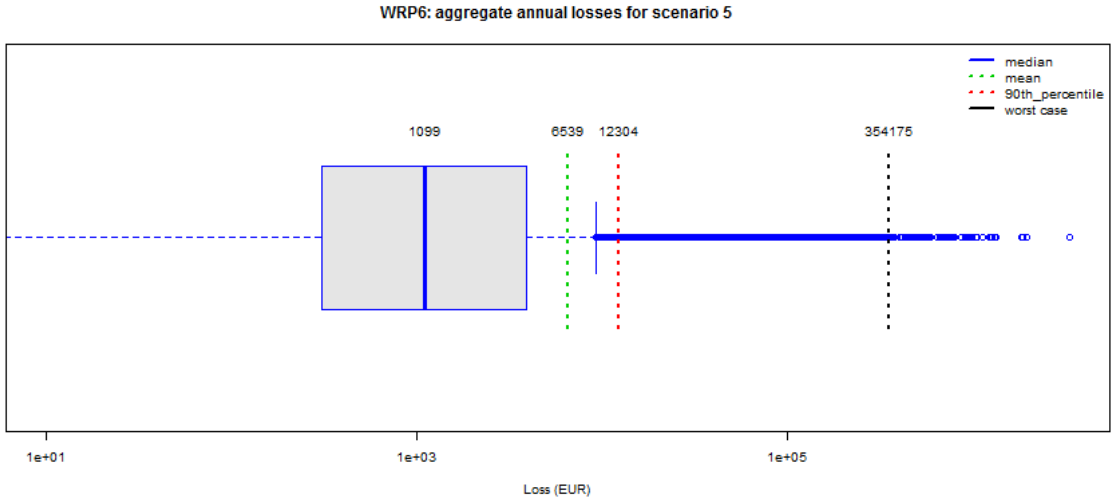


Figure 55. Aggregate annual loss output for validation scenario 5

Validation of the qualitative algorithms

Table 40 shows the overview of the mapping between the median likelihood values produced by the quantitative algorithms and the corresponding qualitative values, as explained in step 3 of the procedure presented at the start of this appendix. It shows the mapping after performing step 3b, that is after adjusting the quantitative algorithms. Notice that there are two instances where the ordering of qualitative likelihood do not correspond to the quantitative values, namely row 10 and row 46. For these two cases, we decided that accepting the slight deviation was preferable to modifying the quantitative algorithm, due to the much lower granularity of the qualitative scale.

Row	WRP no.	Incident	Validation scenario no.	Median likelihood	Qualitative likelihood
1	6	U1	5	1,56	Very high
2	6	U1	4	1,328	Very high
3	8	U1	4	1,2	Very high

4	5	U1	14	1,039	Very high
5	7	U1	5	0,902	Very high
6	4	U1	2	0,899	Very high
7	10	U1	6	0,898	Very high
8	2	U1	5	0,76	Very high
9	10	U1	5	0,676	Very high
10	2	U1	4	0,667	High
11	5	U1	7	0,612	Very high
12	1	U1	5	0,557	Very high
13	1	U1	2	0,545	Very high
14	5	U2	14	0,534	Very high
15	7	U1	3	0,503	Very high
16	5	U2	10	0,458	High
17	8	U1	3	0,446	High
18	6	U1	2	0,444	High
19	9	U1	4	0,409	High
20	5	U2	8	0,399	High
21	5	U1	4	0,36	High
22	9	U1	3	0,339	High
23	7	U1	4	0,238	High
24	5	U1	2	0,219	Medium
25	1	U1	3	0,14	Medium
26	10	U1	4	0,136	Medium
27	10	U1	2	0,135	Medium
28	7	U1	2	0,089	Medium
29	5	U2	13	0,078	Medium
30	3	U1	4	0,068	Medium
31	5	U1	5	0,063	Medium
32	5	U1	6	0,063	Medium
33	2	U1	2	0,037	Medium
34	3	U1	2	0,034	Low
35	2	U1	3	0,021	Low
36	5	U1	3	0,018	Low
37	4	U1	1	0,013	Low
38	5	U1	1	0,013	Very Low
39	5	U1	8	0,013	Very Low

40	5	U1	9	0,013	Very Low
41	5	U1	10	0,013	Very Low
42	5	U1	11	0,013	Very Low
43	5	U1	12	0,013	Very Low
44	5	U1	13	0,013	Very Low
45	2	U1	1	0,011	Very Low
46	8	U1	2	0,007	Low
47	10	U1	3	0,007	Very Low
48	1	U1	1	0,006	Very Low
49	1	U1	4	0,006	Very Low
50	5	U2	12	0,006	Very Low
51	5	U2	1	0,005	Very Low
52	5	U2	2	0,005	Very Low
53	5	U2	3	0,005	Very Low
54	5	U2	4	0,005	Very Low
55	5	U2	5	0,005	Very Low
56	5	U2	6	0,005	Very Low
57	5	U2	7	0,005	Very Low
58	5	U2	9	0,005	Very Low
59	5	U2	11	0,005	Very Low
60	10	U1	1	0,005	Very Low
61	8	U1	1	0,002	Very Low
62	6	U1	3	0,001	Very Low
63	7	U1	1	0,001	Very Low
64	9	U1	2	0,001	Very Low
65	3	U1	1	0	Very Low
66	3	U1	3	0	Very Low
67	6	U1	1	0	Very Low
68	9	U1	1	0	Very Low

Table 40. Overview of validation scenarios with quantitative (median) and qualitative likelihood values