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

D3.2 - CYBER RISK MODELLING LANGUAGE AND GUIDELINES, PRELIMINARY 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 Colgiago (AON), Antonio Alvarez (ATOS), Paolo Lombardi (Trust-IT), Roberto Mannella (REXEL)
Due Date	31.05.2016
Date	01.06.2016
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	03.03.2016	Gencer Erdogan (SINTEF)	Initial version, document outline
0.2	18.04.2016	Atle Refsdal (SINTEF)	Modification of the structure, contents in Section 1.1, notes about planned contents in other sections
0.3	21.04.2016	Atle Refsdal (SINTEF)	Minor change to structure, additions/changes in Section 1, some initial contents in sections 2 and 5.
0.4	25.04.2016	Atle Refsdal (SINTEF)	Completed Section 1, additions/modifications in Section 2.
0.5	29.04.2016	Atle Refsdal (SINTEF)	Section 5, other minor modifications.
0.6	03.05.2016	Atle Refsdal (SINTEF)	Section 7.1
0.7	05.05.2016	Atle Refsdal (SINTEF)	Section 8.1 (incomplete)
0.8	06.05.2016	Atle Refsdal (SINTEF)	Completion of Section 8.1, other minor changes
0.9	09.05.2016	Gencer Erdogan, Atle Refsdal (SINTEF)	Section 4.1
0.10	09.05.2016	Atle Refsdal (SINTEF)	Restructuring and initial content to Section 3
0.11	13.05.2016	Atle Refsdal (SINTEF)	Sections 3, 5.3, 9, executive summary.
0.12	13.05.2016	Giorgio Aprile (AON), Sara Poidomani (AON)	Sections 4.2, 4.3
0.13	14.05.2016	Atle Refsdal (SINTEF)	Minor corrections, Section 6 (incomplete)
0.14	15.05.2016	Atle Refsdal (SINTEF), Giorgio Aprile (AON), Romina Colgiago (AON), Antonio Álvarez (ATOS), Paolo Lombardi (Trust-IT)	Completed Section 6, restructuring with separete sections for economic and societal impact (9, 10), inserted contributions for sections 9 and 10, minor corrections
0.15	19.05.2016	Atle Refsdal (SINTEF)	Minor corrections after comments
0.16	22.05.2016	Atle Refsdal (SINTEF)	Minor corrections after

			comments, in particular in Section 9
0.17	23.05.2016	Antonio Álvarez (ATOS)	Review
0.18	25.05.2016	Roberto Mannella (REXEL)	Review
0.19	27.05.2016	Atle Refsdal (SINTEF)	Updates after review, included mitigations in Section 4.1 and added Section 6.7.
0.20	30.05.2016	Antonio Álvarez (ATOS)	Updates in Section 10
0.21	30.05.2016	Atle Refsdal (SINTEF)	Minor additions to executive summary and conclusion
0.22	30.05.2016	Giorgio Aprile (AON)	Addressing issues at section 9
0.23	30.05.2016	Atle Refsdal (SINTEF)	Added appendices with examples of guideline application and made corresponding updates in executive summary, introduction and conclusion
0.24	01.06.2016	Atle Refsdal (SINTEF)	Minor corrections of language, punctuation etc.
1.0	01.06.2016	Antonio Álvarez (ATOS)	General Assembly approval and delivery to the EC

Disclaimer

Table of Contents

List of Tables

List of Figures

Executive Summary

This document reports on the work carried out during the first year of the project 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.

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 has been an important part of the work undertaken within WP3 during the first part of the project. A description of the role of cyber risk modelling as established during the first year 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.

As the title suggests, this document presents a preliminary version of the WISER modelling languages and guidelines. The final version will be presented in D3.4. Together, D3.2 and D3.4 will document the majority of the results from the WP3 tasks. For this preliminary version, the focus is primarily on the modelling languages and guidelines for cyber risk, threat and vulnerability modelling, as well as structured approaches for economic and societal impact modelling. The reason is that this constitutes the basic core required to fulfil the role of the modelling in the WISER framework. We motivate our choice of modelling languages based on an evaluation of a set of desirable properties, such as comprehensibility, expressiveness and editor support.

We also 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 [13]. The purpose of Step 2 is to define an algorithm for continuous risk monitoring based on dynamic indicators, using either DEXi [6] for qualitative assessments or **R** [21] for quantitative assessments. In addition, the Decision Support System will offer dedicated support for societal impact assessment that combines qualitative assessments with the use of numerical weights, as further explained in Section 10.

We offer guidelines for how to develop CORAS diagrams in Step 1, as well as for defining assessment algorithms in Step 2 based on CORAS diagrams, using either DEXi for qualitative assessments or **R** for quantitative assessments. Here we take a modular approach, explaining how to represent fragments of a CORAS diagram by corresponding fragments of a DEXi model or **R** script. We also offer examples of a DEXi model and an **R** script developed based on a complete CORAS diagram according to the guidelines.

The CORAS diagrams are primarily an aid in the identification of threats, vulnerabilities, potential incidents and assets, and also offers support for assessing the likelihood of incidents. For assessment of the economical or societal impact of risk, we offer guidelines that do not depend on 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.

The final version of this report is due in March 2017 (M22). We then expect to present more refined cyber risk modelling guidelines based on the experiences we gather in the coming months, in particular in the context of the full scale pilots, as well as guidelines for target modelling.

1 Introduction

1.1 Purpose and Scope

This document reports on the work carried out during the first year of the project 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 documented in D2.3 in M9 (February 2016). This design is again based on the requirements and initial version of the design documented in D2.2 in M6 (November 2015). 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 has been an important part of the work undertaken within WP3 during the first part of the project. A description of the role of cyber risk modelling as established during the first year is therefore included in this report.

As the title suggests, this document presents a preliminary version of the WISER modelling languages and guidelines, to be submitted at M12 (May 2016). The final version, D3.4, is due in M22 (March 2017). Together, D3.2 and D3.4 will document the majority of the results from the WP3 tasks. More specifically, this includes results from the tasks on cyber target modelling (T3.1), cyber risk, threat and vulnerability modelling (T3.2), societal impact modelling (T3.4), economic impact modelling (T3.5) and cyber risk treatment selection (T3.6).

For this preliminary version, the focus is primarily on the modelling languages and guidelines for cyber risk, threat and vulnerability modelling, as well as economic and societal impact modelling. The reason is that this constitutes the basic core required to fulfil the role of the modelling in the WISER framework. We motivate our choice of modelling languages, 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. In D3.2, the contributions from the tasks are as follows: T3.2 and T3.6 has primarily contributed to sections 5-8, T3.4 has primarily contributed to Section 10, T3.5 has primarily contributed to Section 9. All the tasks have contributed to sections 2-4.

In order to support different types of user needs and service delivery models, we address 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.

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. Having thus provided the context, 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 the overall method for risk cyber risk modelling in WISER. This overall method is the same whether one chooses to use qualitative or quantitative assessment. Section 6 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 7, we give guidelines for defining qualitative assessment algorithms based on a CORAS diagram using DEXi, while Section 8 offer similar guidelines for quantitative algorithms using **R**. Although simple support for impact assessment is included in Section 7 and Section 8, these sections are primarily dedicated to likelihood assessment. In Section 9, we present a more in-depth approach to economic impact assessment, while societal impact assessment is addressed in Section 10. We then conclude in Section 11.

This document also contains five appendices. Appendix I defines what it means for a function on intervals to be monotonically increasing, as this concept plays a role in the guidelines offered in Section 8. Appendix II presents the naming conventions we use in order to ensure clear links between the elements of a CORAS diagram and a corresponding DEXi model or **R** script. The next three appendices illustrate results of applying the guidelines from Section 7 and Section 8 on a CORAS diagram. First, we present the CORAS diagram in Appendix III. Then we show a corresponding DEXi model in Appendix IV, and finally a corresponding **R** script in Appendix V.

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 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. We try out, modify, refine and validate the modelling approach described here in the context of the pilots performed in WP6. Based on the experiences from the pilots, the final version of the modelling languages and guidelines (D3.4) will be submitted in March 2017 (M22), thereby completing the report and ensuring that all relevant WP3 tasks are covered.

This 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 DEXi or **R** based on those patterns. Moreover, it is related to D8.7 (Exploitation plan & business models, first version), which explains the role of the modelling approach in the WISER exploitation plan and business model.

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 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.

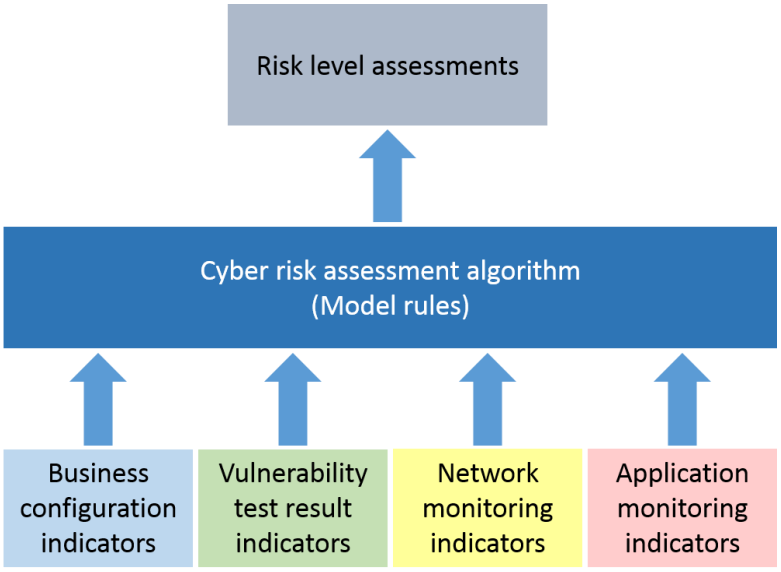


Figure 1 Definition of aggregation from indicators to risk level assessments

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. 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 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. The overall method for cyber risk modelling in WISER therefore consists of the two main steps illustrated in Figure 2.

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

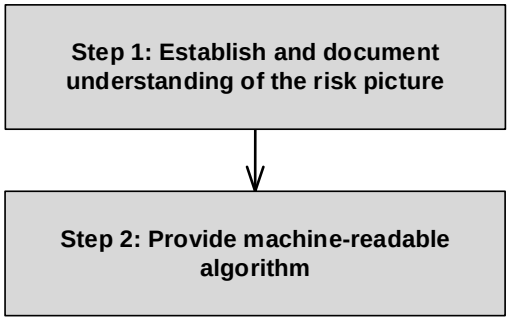


Figure 2 Outline of overall method for cyber risk modelling

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.

The first step is performed using CORAS [13] as the modelling language. For the machine-readable algorithms to be developed in the second step of the method, WISER offers different modelling 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 1, 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 [7].
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 [13], as well as a number of papers [1], [3], [7], [20].
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 7 and 8 show how to exploit the structure of the CORAS diagrams to develop assessment algorithms

Table 1 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. Developing the RAE should not take too much of the available WISER efforts. We therefore include the following requirement:

> LR2c: *The language should not require extensive effort to implement 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 2 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 allow 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 2 Evaluation of DEXi fulfilment of requirements to Step 2

3.2.3 R fulfilment of the requirements for quantitative modelling

Table 2 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 3 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 5. 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.

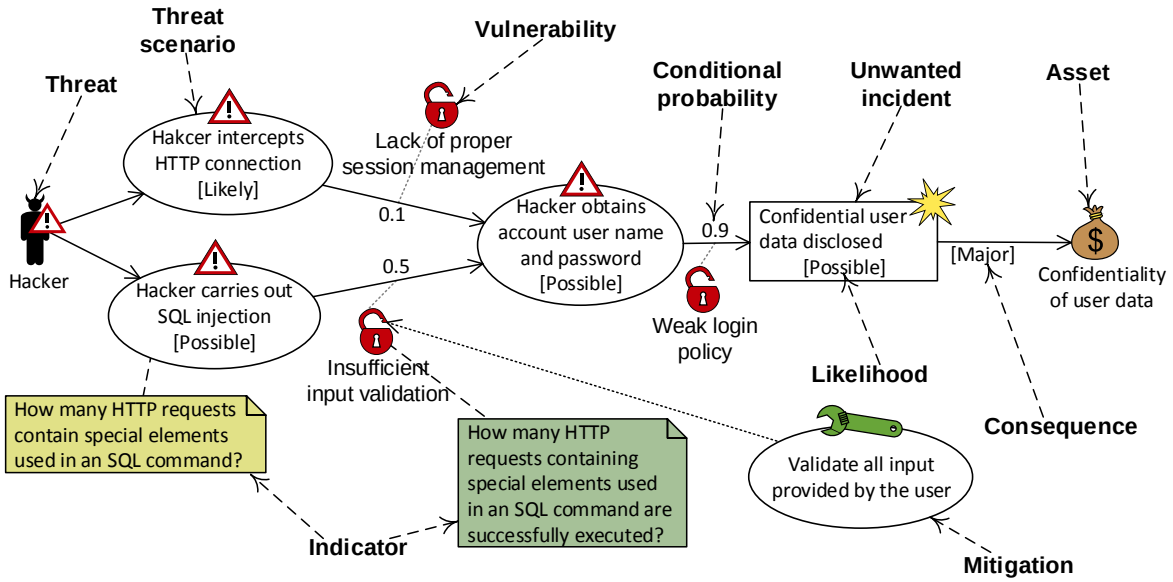


Figure 3 Example of a CORAS risk model

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. [13] 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 [6] 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 provides an example of utility function for an aggregate attribute “CAR” with two descendants “PRICE”, “TECH.CHAR”: the utility function is a table associating a value for the aggregate node (CAR) to each combination of values for the basic nodes PRICE, TECH.CHAR; the scales of the attributes are here:

PRICE: high, medium, low

TECH.CHAR: bad, acceptable, good, excellent

CAR: unacc, acc, good, exc

	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 4 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 4, 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 [18] [21] 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 [16], with a rich set of learning and reference resources for both the beginners and expert **R** users. We refer to the **R** documentation [17] for a thorough description and to D3.3 for details on the **R** tool.

5 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 5 shows the overall method for cyber risk modelling, considering these aspects.

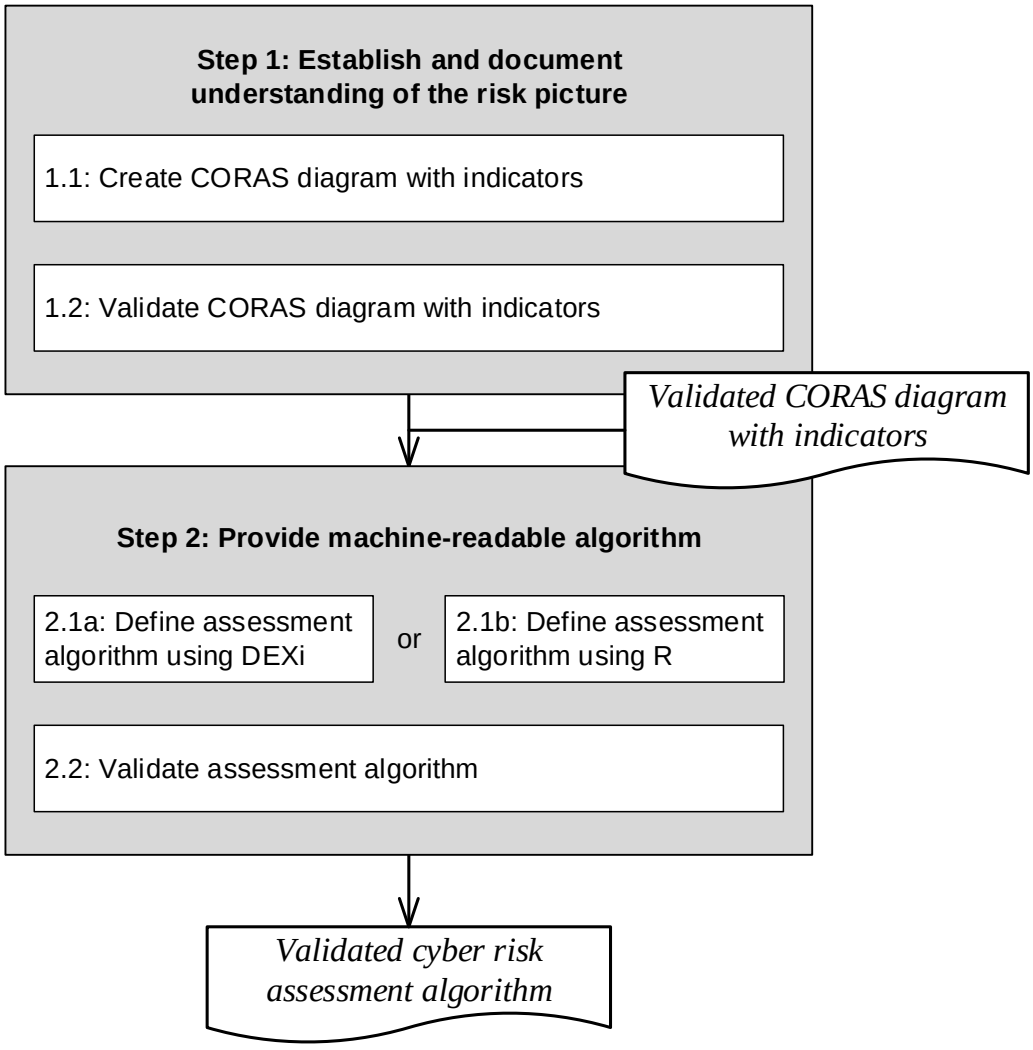


Figure 5 Overall method for cyber risk modelling

The outcome of the first step is a validated CORAS diagram with indicators. This diagram captures the relevant 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 6, 7 and 8 then provide specific modelling guidelines for each of the three modelling languages CORAS, DEXi and **R**.

5.1 Establish and document understanding of the risk picture

5.1.1 Create CORAS diagram with indicators

As illustrated by Figure 5, 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.

5.1.2 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 [9], ISO 27005 [10], ISO 27032 [11], CAPEC [14] and OWASP [15]. 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.

5.2 Provide machine-readable algorithm

5.2.1 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 7 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**.

5.2.2 Validate assessment algorithm

The point regarding the impossibility of establishing objective correctness and completeness of a CORAS diagram discussed in Section 5.1.2 applies, of course, also for the corresponding assessment algorithm. Lenstra and Voss [12] (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 7 and Section 8 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.

5.3 A comment on the flexibility of the method

As explained earlier, the method illustrated by Figure 5 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.

6 Guidelines for 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], [13]. 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 [14].

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 [5] 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 [13]. 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 [19].

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.

6.1 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.

6.1.1 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.

6.1.2 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.

6.1.3 Example diagram

Figure 6 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.

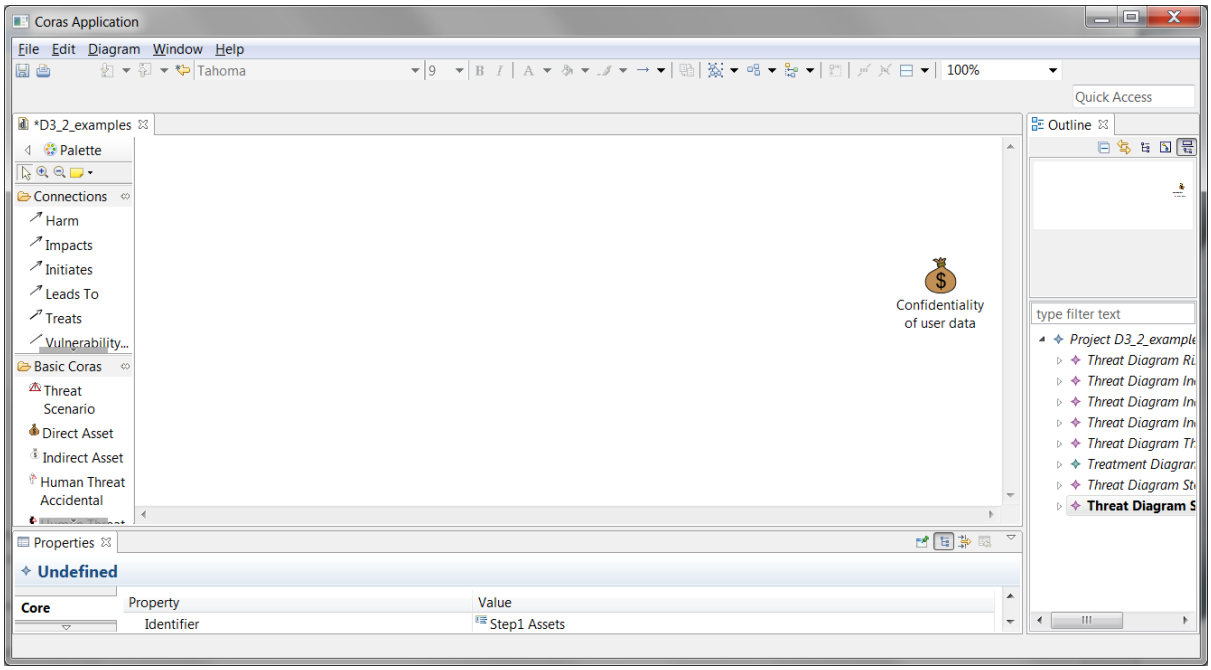


Figure 6 Adding assets to a CORAS diagram

6.2 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.

6.2.1 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?

6.2.2 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.

6.2.3 Example diagram

Figure 7 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.

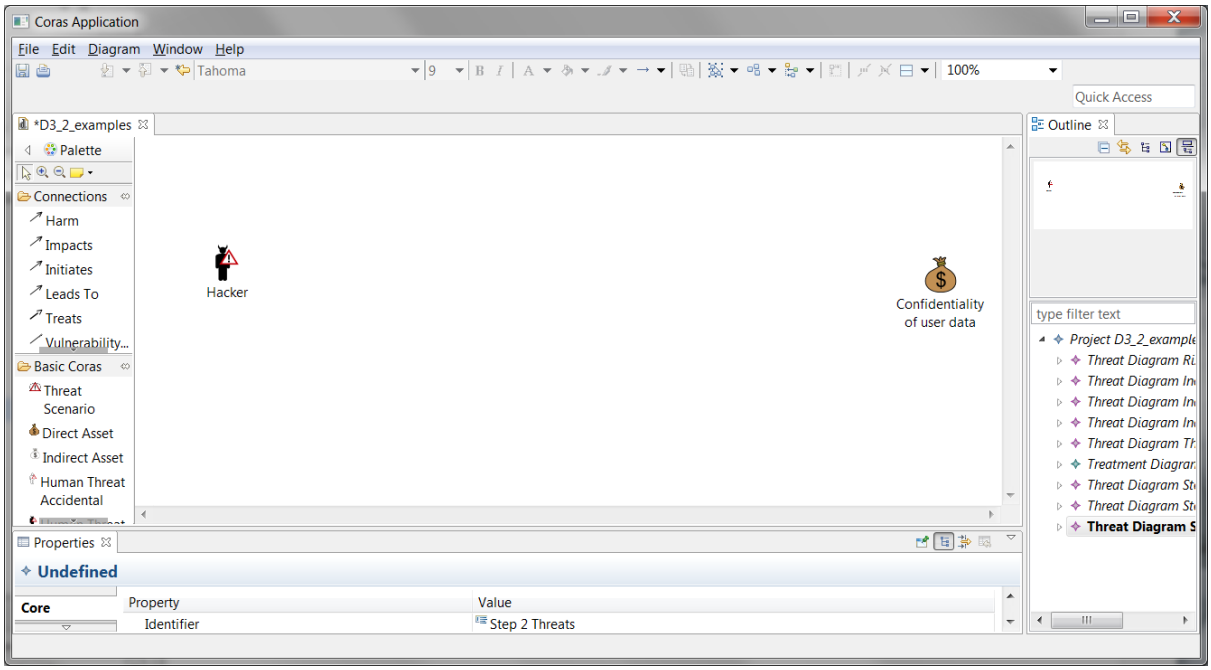


Figure 7 Adding threats to a CORAS diagram

6.3 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.

6.3.1 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?

6.3.2 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.

6.3.3 Example diagram

Figure 8 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.

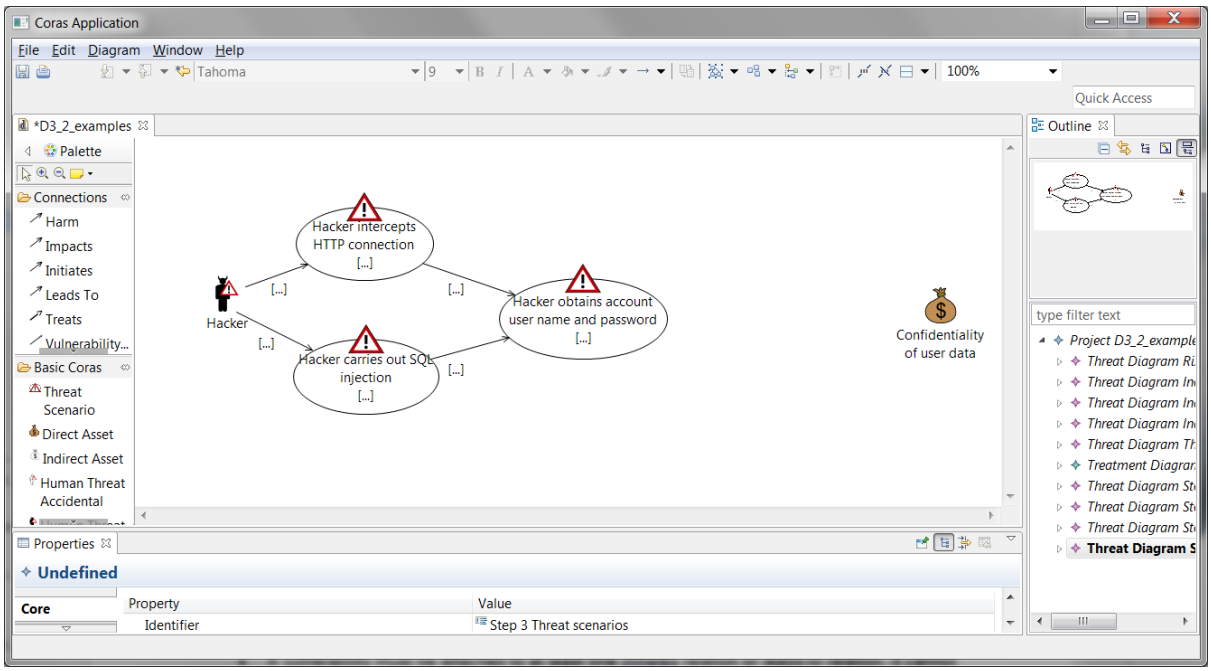


Figure 8 Adding threat scenarios

6.4 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.

6.4.1 Guiding questions

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

6.4.2 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.

6.4.3 Example diagram

Figure 9 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.

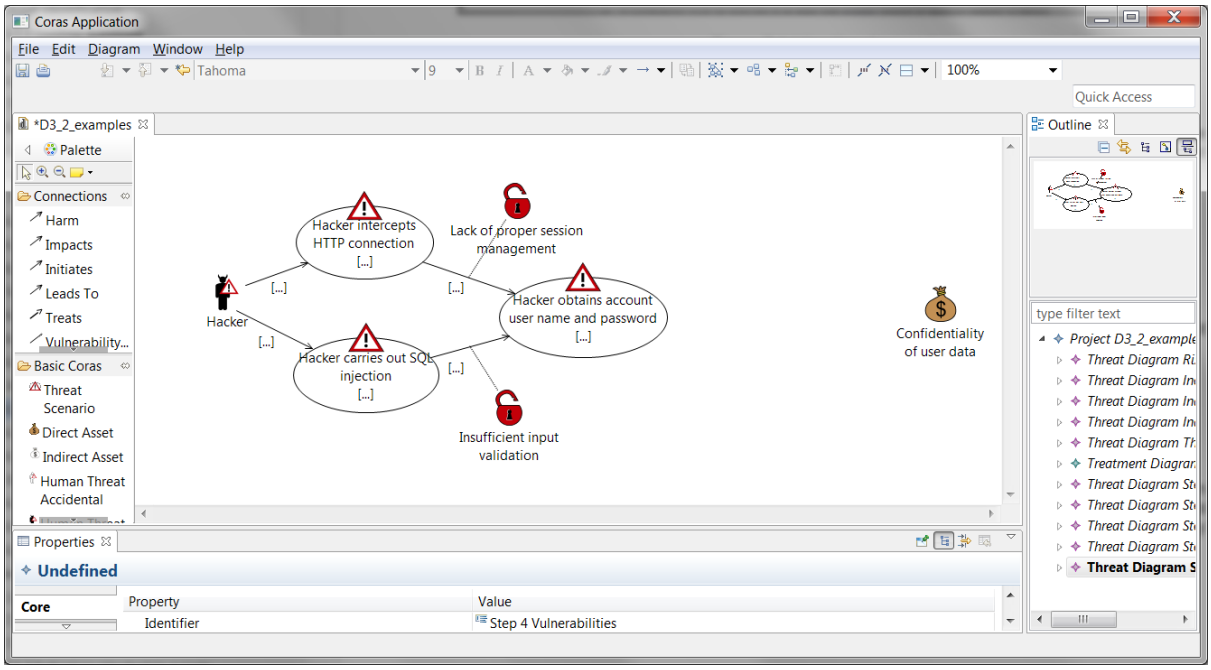


Figure 9 Adding vulnerabilities

6.5 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.

6.5.1 Guiding questions

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

6.5.2 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.

6.5.3 Example diagram

Figure 10 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.

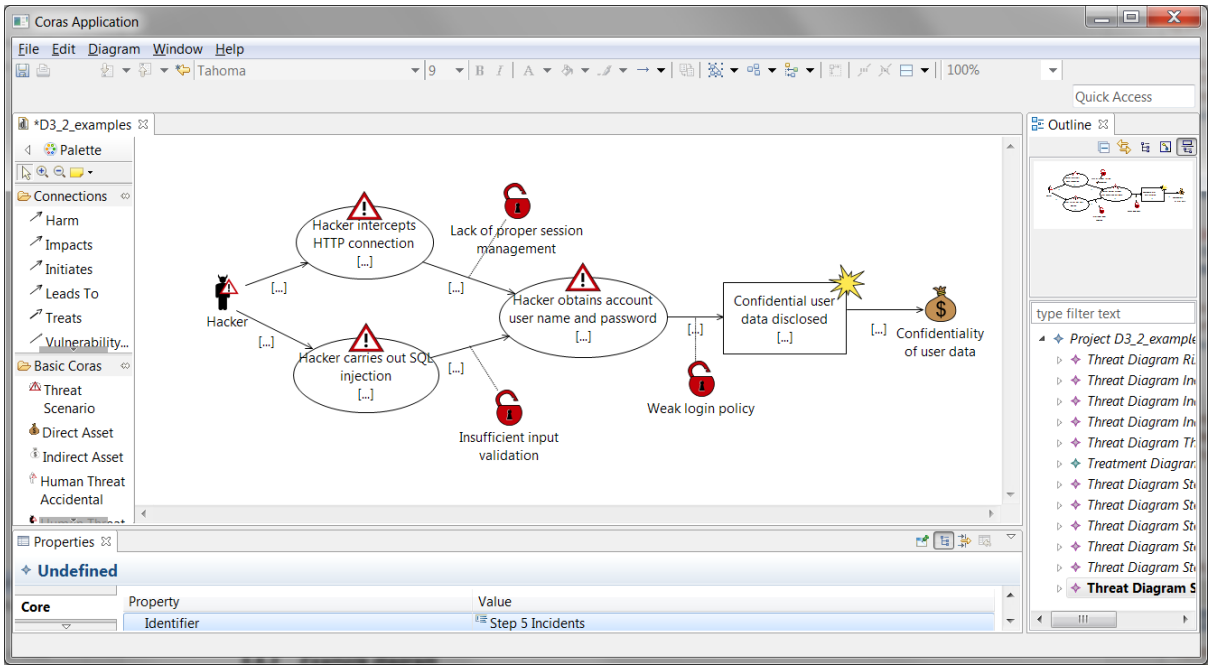


Figure 10 Adding incidents

6.6 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.

6.6.1 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.

6.6.2 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.

6.6.3 Example diagram

Figure 11 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.

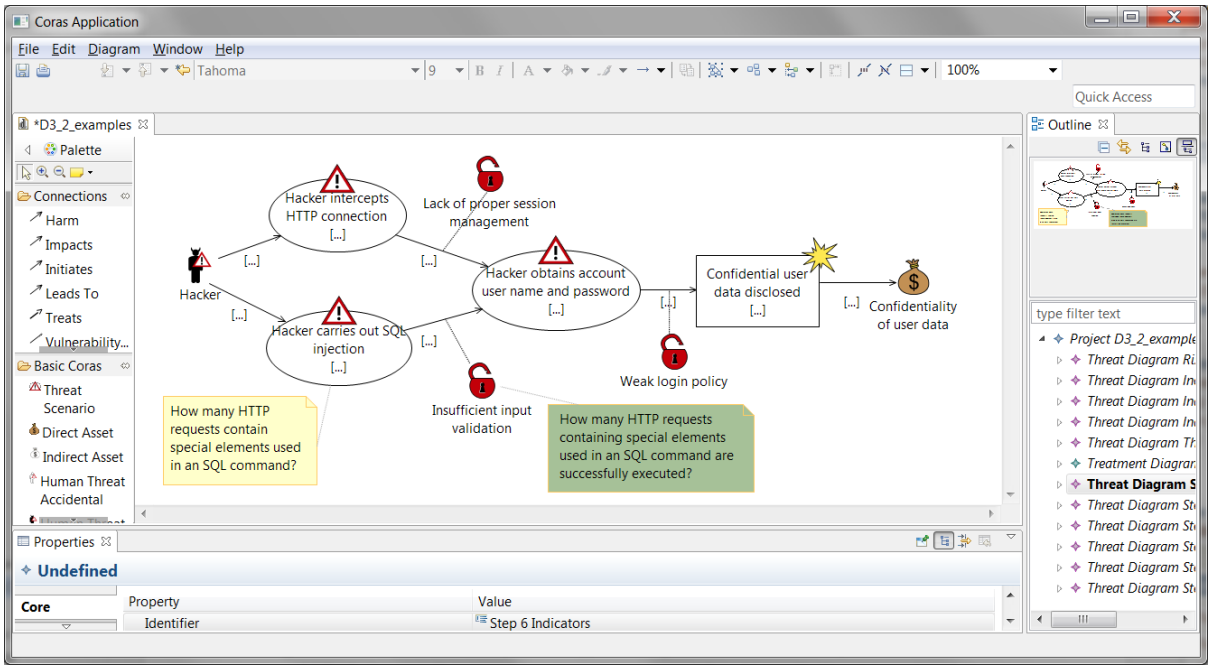


Figure 11 Adding indicators

6.7 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.

6.7.1 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?

6.7.2 Syntactical constraints

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

6.7.3 Example diagram

Figure 12 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.

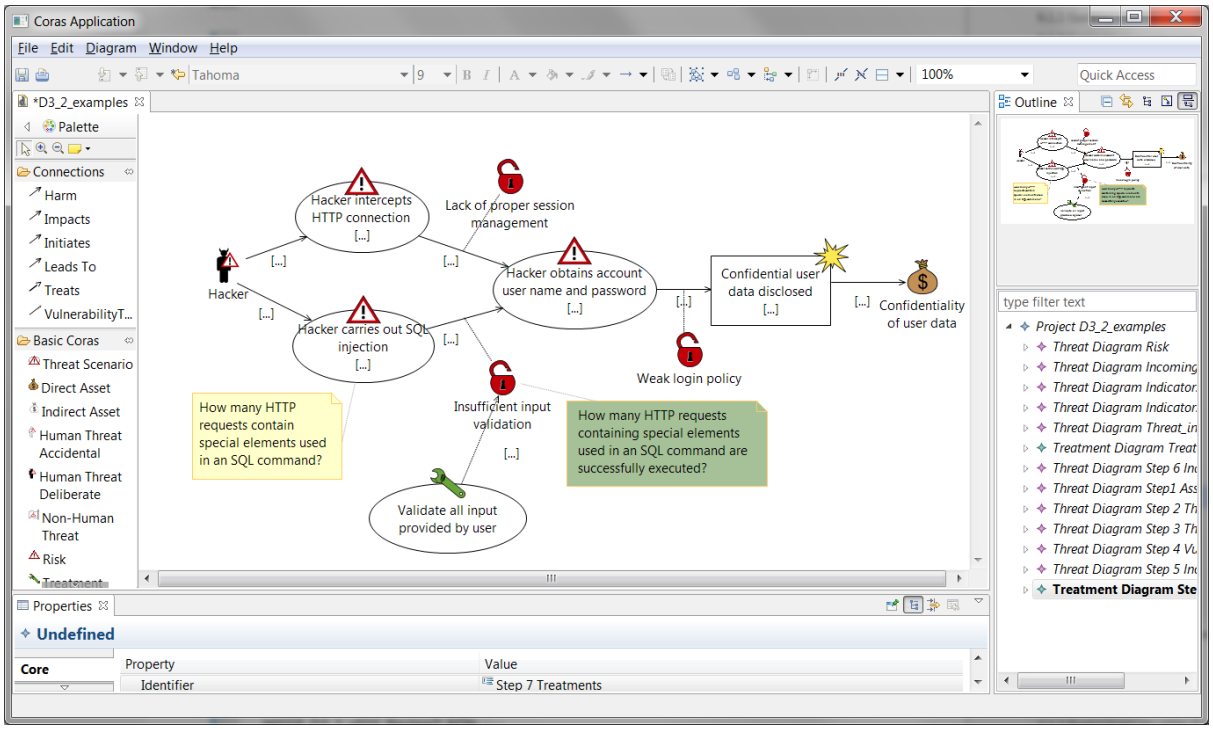


Figure 12 Adding mitigations

7 Guidelines for 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 is provided in Appendix IV.

CORAS defines a number of different diagram types [13]. 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.

7.1 Risk level

7.1.1 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 13 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 Appendix I. 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.

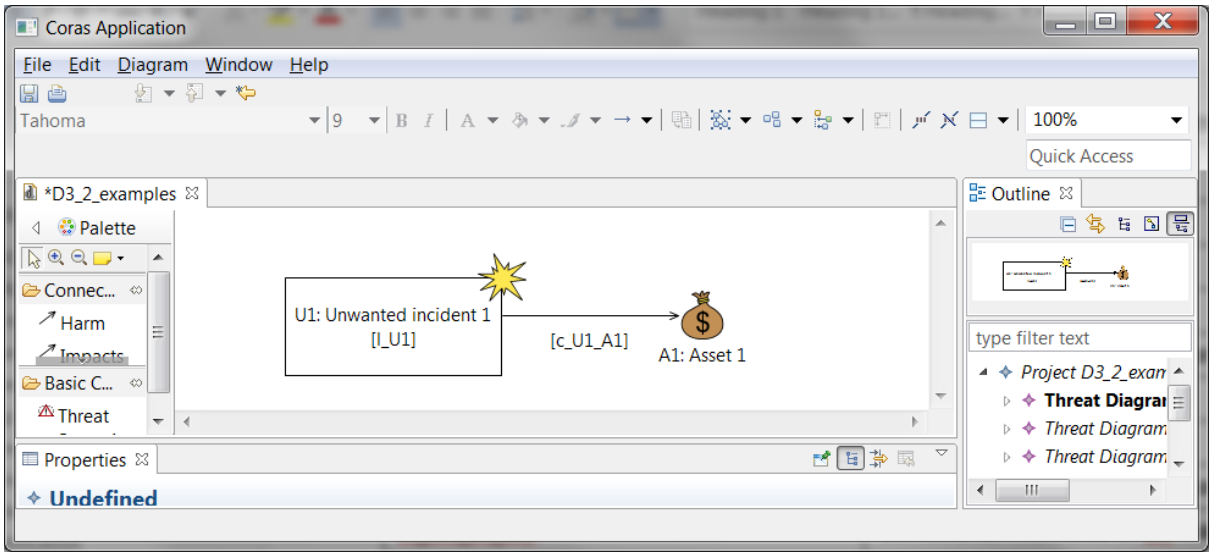


Figure 13 CORAS fragment representing a risk

7.1.2 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 14 shows the DEXi-representation of the CORAS fragment shown in Figure 13, where R1 represents the risk, l_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.

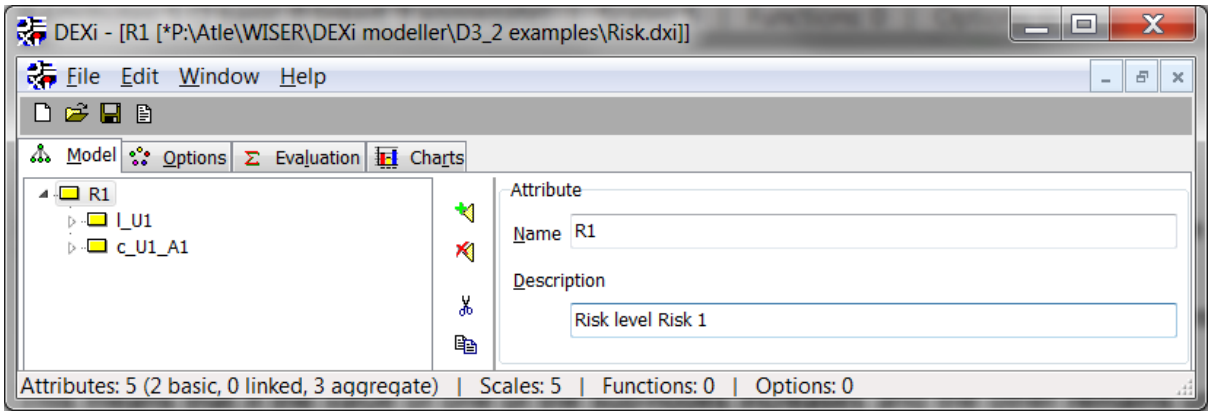


Figure 14 DEXi fragment representing a risk

7.1.3 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 14, R1 should be monotonically increasing in l_U1 and c_U1_A1.

Figure 15 shows an example of how a utility function fulfilling this restriction might be defined.

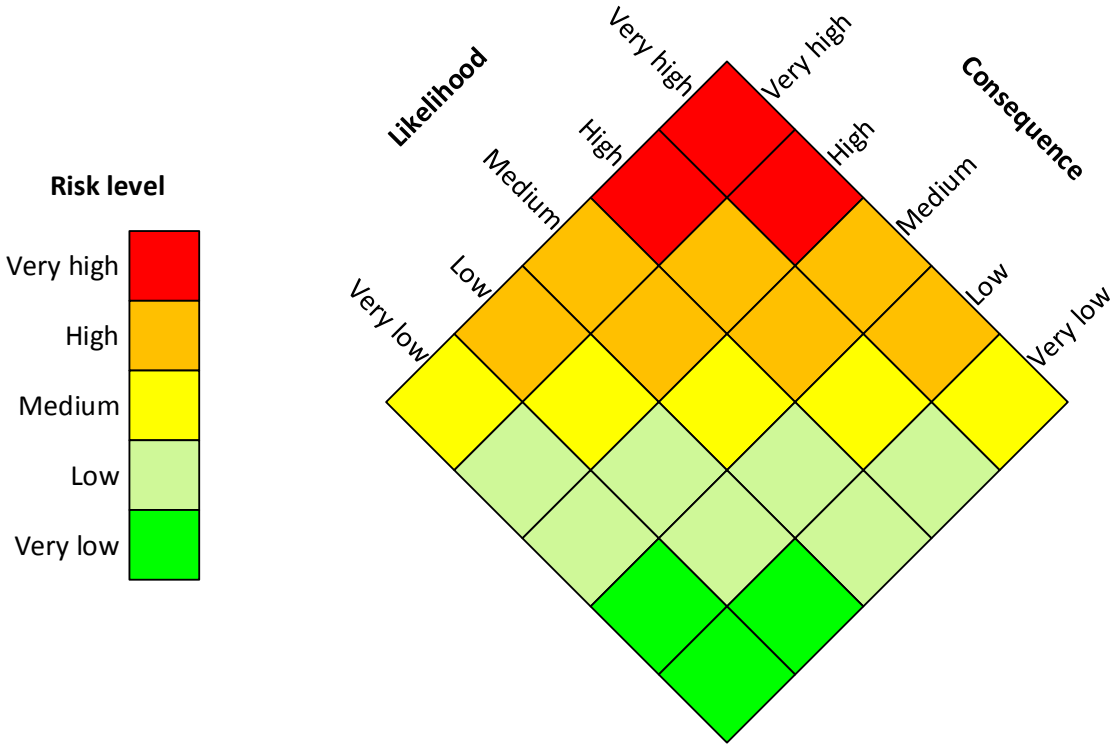


Figure 15 Example of utility function defining risk level as a function of likelihood and consequence

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.

7.2 Incoming 'leads-to' relations to a node

7.2.1 CORAS representation

Figure 16 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.

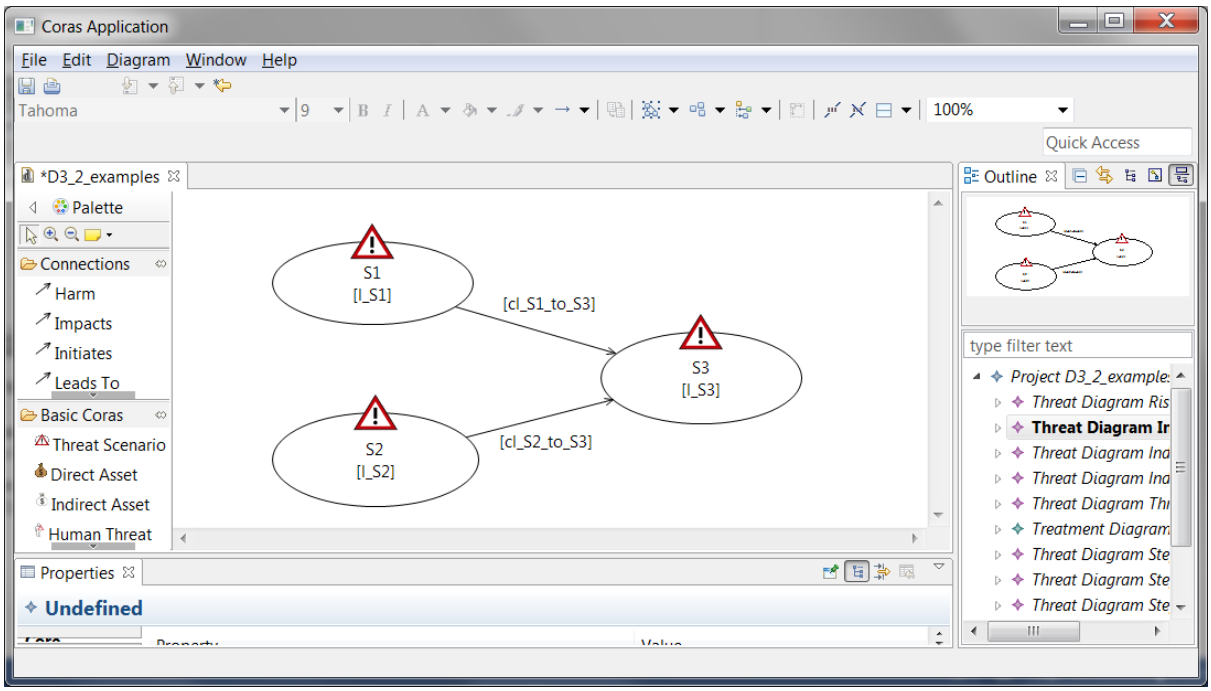


Figure 16 CORAS fragment representing incoming 'leads-to' relations

7.2.2 DEXi representation

Figure 17 shows a DEXi fragment corresponding to the CORAS fragment in Figure 16.

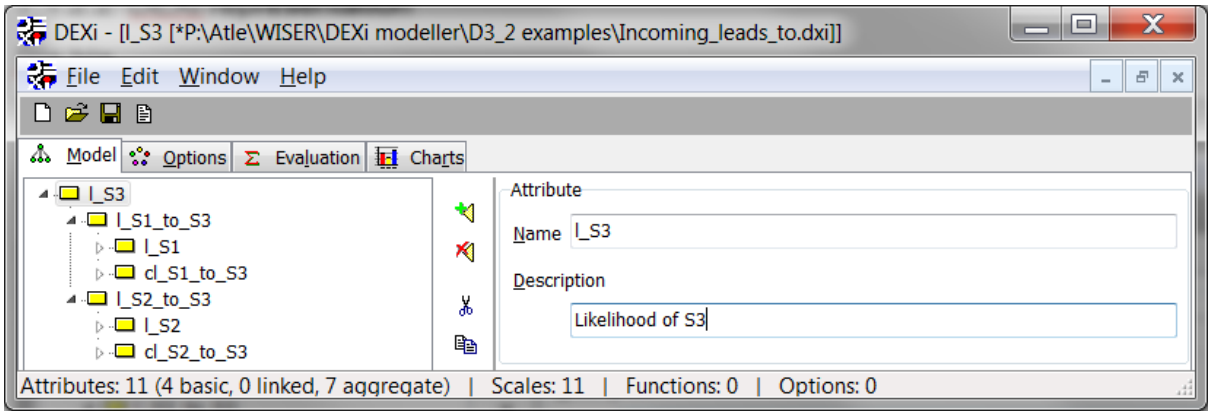


Figure 17 DEXi fragment representing incoming 'leads-to' relations

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 17 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 17. As these may again depend on incoming branches, the model allows any finite number of levels in the DEXi tree.

7.2.3 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 (I_S3), the aggregation of the likelihood contributions from all incoming branches (I_S1_to_S3 and I_s2_to_S3). The second defines, at the level of each incoming branch (either I_S1_to_S3 or I_S2_to_S3), the aggregation of the likelihood of the source-node (either I_S1 or I_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 (I_S3 in Figure 17) is monotonically increasing in all its direct sub-nodes (I_S1_to_S3 and I_S1_to_S3 in Figure 17). It does not have to be *strictly* increasing.

Example: In Figure 17, I_S3 should be monotonically increasing in I_S1_to_S3 as well as I_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 17, 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.

7.3 Indicators attached to a node

7.3.1 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 18 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.

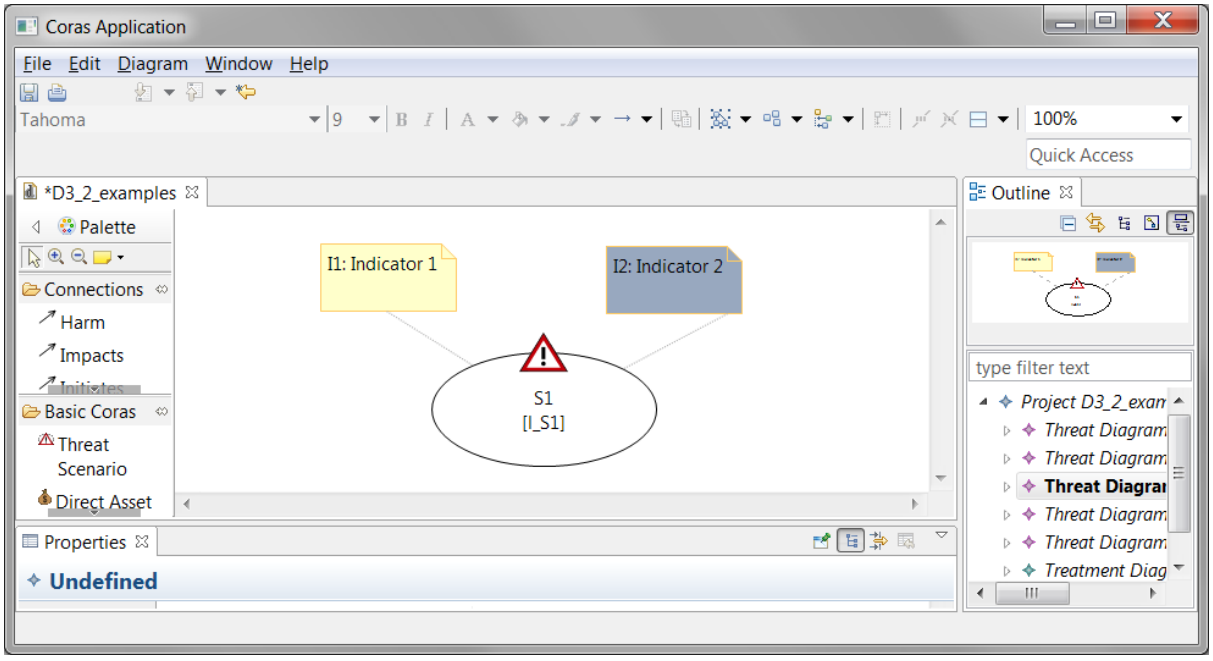


Figure 18 CORAS fragment representing a node with attached indicators

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.

7.3.2 DEXi representation

Figure 19 shows a DEXi fragment corresponding to the CORAS fragment in Figure 18.

[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.

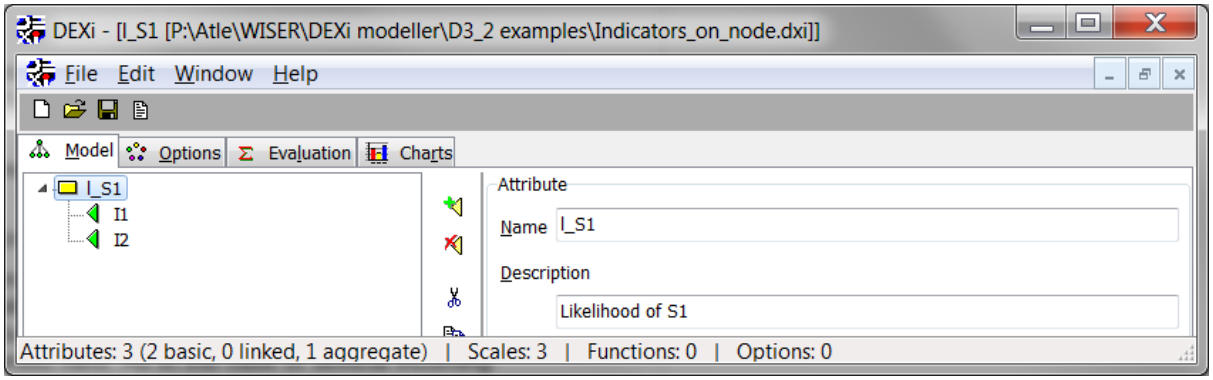


Figure 19 DEXi fragment representing a node with attached indicators

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.

7.3.3 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 19, I_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 7.2 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 7.2.3 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.

7.4 Indicators attached to a 'leads-to' relation

7.4.1 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 20 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.

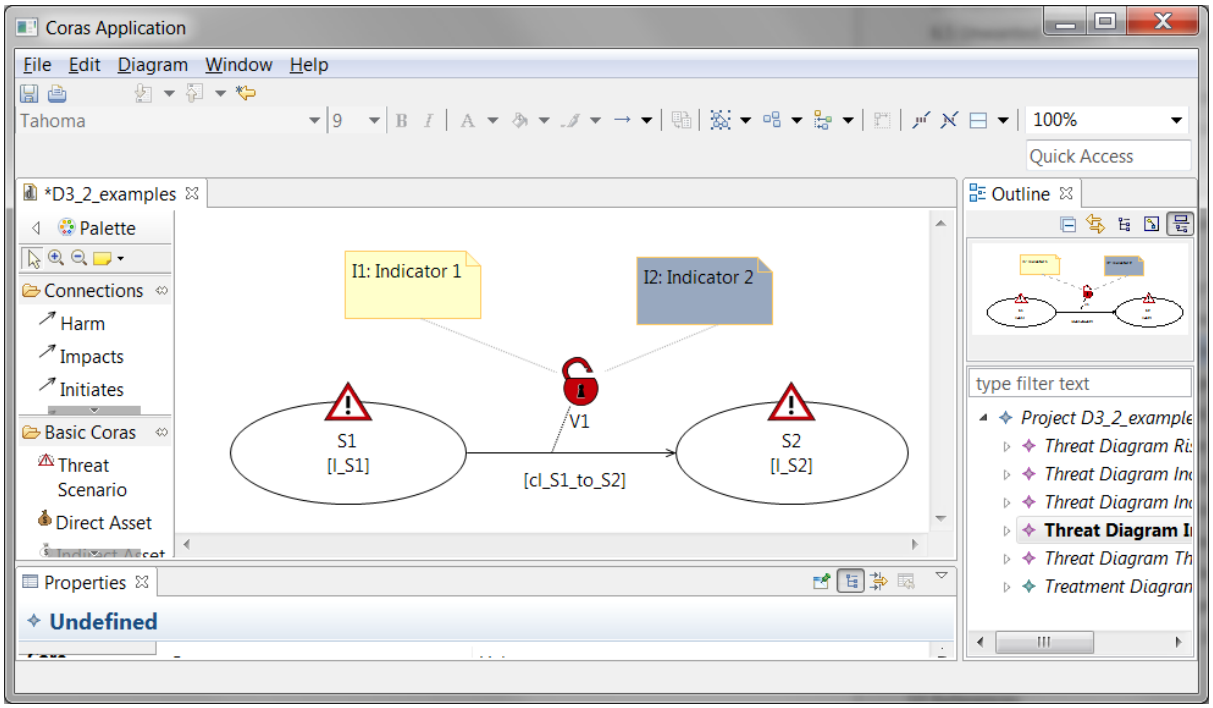


Figure 20 CORAS fragment representing 'leads-to' relation with indicators

7.4.2 DEXi representation

Figure 21 shows a DEXi fragment corresponding to the CORAS fragment in Figure 20.

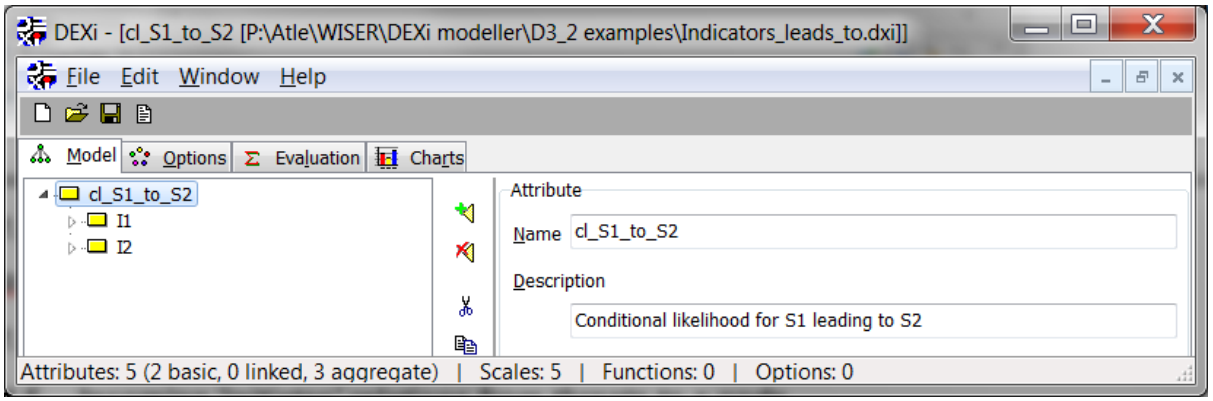


Figure 21 DEXi fragment representing 'leads-to' relation with indicators

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 7.3.

7.4.3 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 21, cl_S1_to_S2 should be monotonically increasing in I1 as well as I2.

7.5 Mitigation proposal triggering

7.5.1 CORAS representation

CORAS diagrams can be used to show risk mitigation options by attaching these to the different elements of a risk model. Figure 22 shows such a diagram.

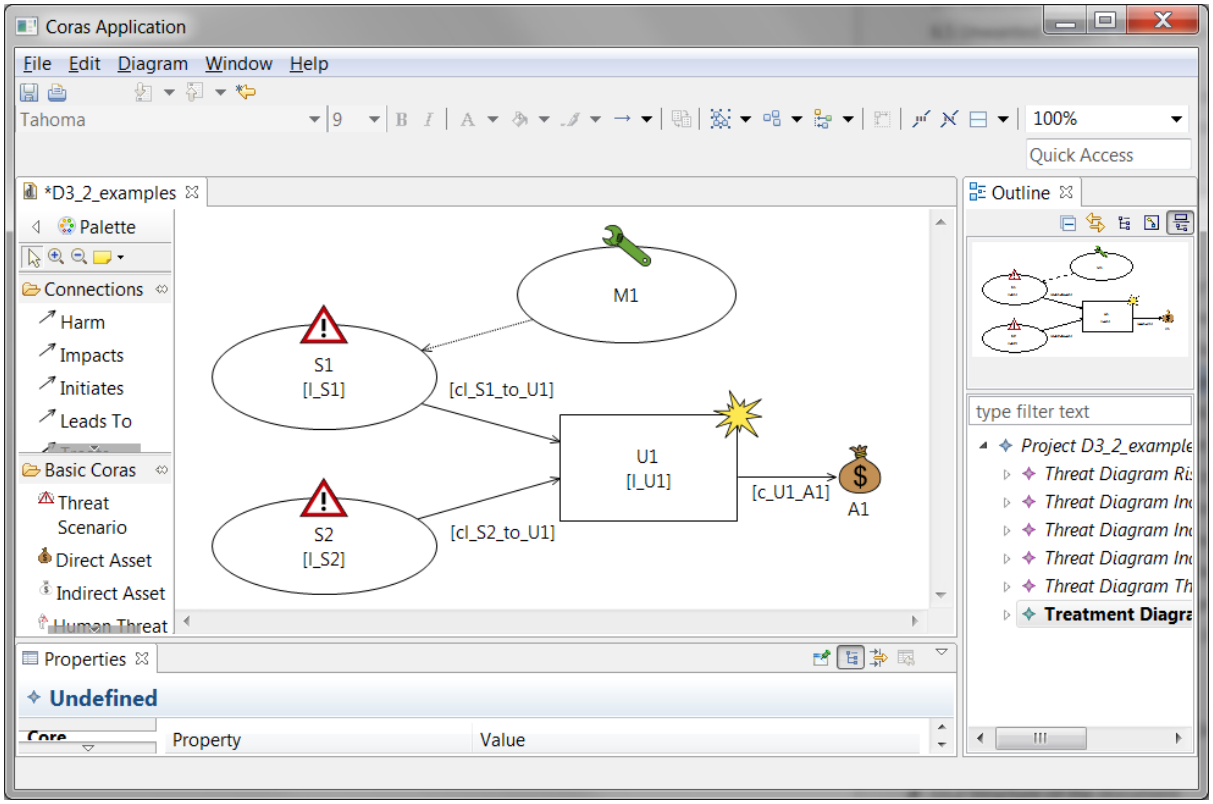


Figure 22 CORAS fragment associated with mitigation proposal

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.

7.5.2 DEXi representation

Figure 23 shows a DEXi fragment used to trigger a proposal for implementing mitigation M1.

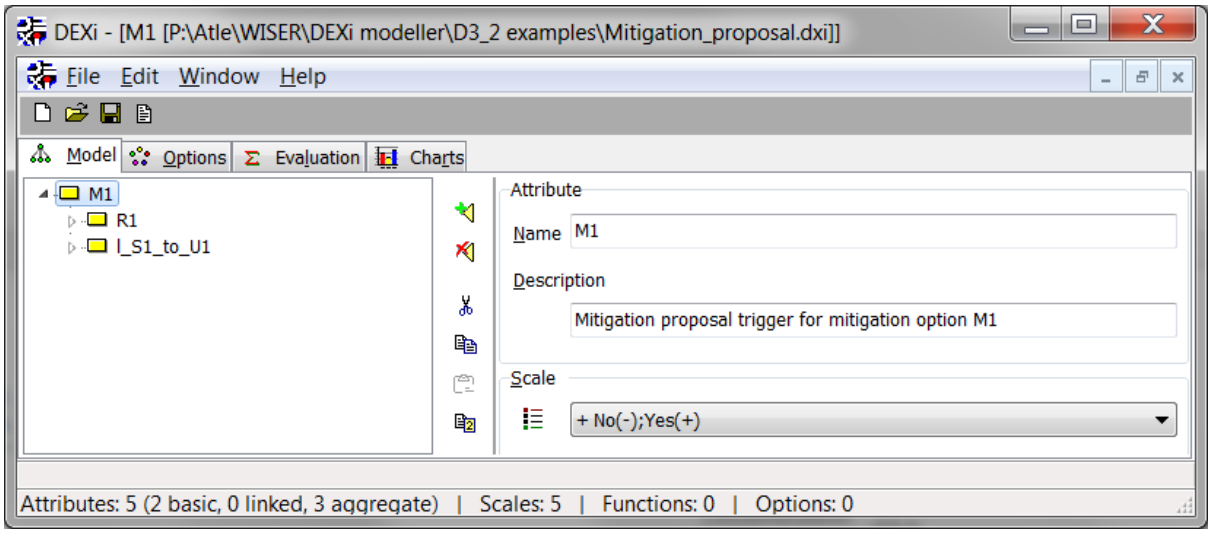


Figure 23 DEXi fragment for triggering a mitigation proposal

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 22, the only risk that M1 has the potential to reduce is the risk of incident U1 harming asset A1. In Figure 23, 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 I_S1_to_U1, representing the likelihood contribution from S1 to U1 (and hence the second condition above).

7.5.3 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 23, the value of M1 should be monotonically increasing in R1 and I_S1_to_U1.

8 Guidelines for defining quantitative assessment algorithms using R from CORAS diagrams

In this section, we provide guidelines for defining quantitative assessment algorithms using **R** based on a CORAS diagram. More specifically, we show how to exploit the structure of a CORAS diagram

and the CORAS calculus to help defining algorithms for likelihood assessment. Although the focus is on likelihood, we also briefly address, in Section 8.1, a simple form of impact/consequence assessment in a similar manner as we did in Section 7.1. A much more extensive presentation of economic impact assessment, which does not depend on CORAS diagrams, will be given in Section 9, while societal impact assessment is addressed in Section 10. In Section 8.5, we show how to trigger mitigation proposals.

The quantitative assessment algorithms are defined as **R** scripts, which are files on the form 'name.R'. An example of an **R** script resulting from following the guidelines is provided in Appendix V.

We assess the likelihood of nodes (i.e. threat scenarios and incidents) in terms of frequency intervals. Hence, for each node, the assessment algorithm should assign a frequency interval. For the conditional likelihoods assigned to 'leads-to' relations we use probability intervals.[3]

The use of intervals rather than exact values represents the uncertainty that is always associated with risk assessments. A frequency is given on the form 'x times per time unit', where x is a positive number and the time unit could be, for example, a month or a year. In the following, we will assume that the time unit is the same for all intervals and not explicitly represented. This means that in the assessment algorithms, a frequency interval is represented simply as an ordinary interval. Of course, when presenting assessment results to end-users, the time unit should be explicitly stated.

The CORAS calculus requires the use of interval arithmetic when reasoning about frequency intervals. To support this we use the 'sets' package[4] for **R**. Figure 24 shows how we can define an interval using this package.

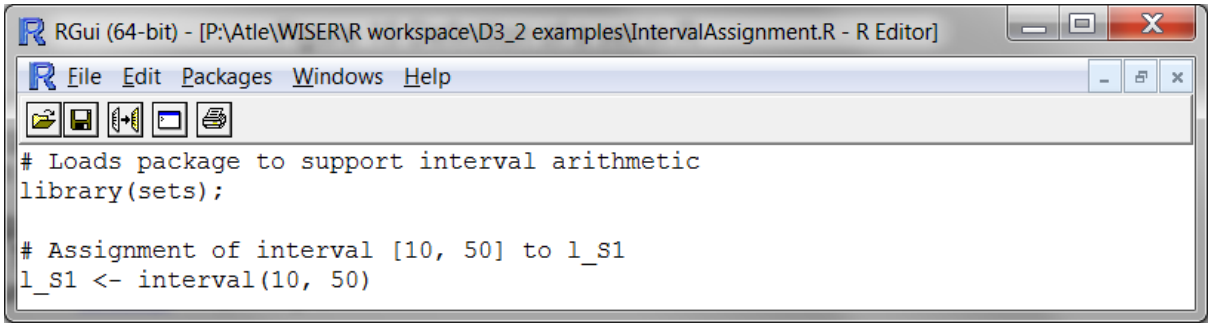


Figure 24 Definition of an interval in **R**

Here, l_S1 will be defined as the (closed) interval of decimal numbers from 10 to 50. By passing additional arguments to the constructor, it is also possible to specify whether the interval should be closed or open to the left and right, although this will not be needed for our presentation here. The 'sets' package allows us to use the arithmetic operators on intervals in the same way as on numbers. Hence, if x=[5,7] and y=[3,4], then x+y=[8,11] and x*y=[15,28].

In the following, we provide guidelines for defining quantitative assessment algorithms for the same CORAS fragments as in Section 7. We use the same naming convention, which is also summarized in Appendix II. This means, for example, that l_S1 denotes the likelihood of the node S1, which in this section is assumed to be a frequency interval, while cl_S1_to_S2 denotes the conditional likelihood that an occurrence of S1 will lead to S2, which in this section is assumed to be a probability interval. Although we use intervals in these guidelines, we may also consider other distributions later in the project.

We use a modular approach, and follow a similar structure to Section 7. This means that, for each case to be addressed, we start by referring to the relevant figure showing the relevant CORAS

[3] If desired, we could also use conditional ratios rather than conditional probabilities. Unlike a probability, a ratio can be higher than 1, which allows us to express that one occurrence of the source node may lead to several occurrences of the source node.
[4] See https://cran.r-project.org/web/packages/sets/index.html (accessed 4/5-2016).

fragment. Then we show an example of what a corresponding (fragment of) an **R** script may look like. It is important to understand that these are only meant as examples, as there is no single correct way to define the script fragment. Finally, we present some general restrictions that *any* such **R** fragment should fulfil.

8.1 Risk level

8.1.1 CORAS representation

The CORAS fragment is identical to Figure 13 presented in Section 7.1.1.

8.1.2 R representation

The risk level is determined by the likelihood of the incident in question and its consequence for the relevant asset. We assume that the consequence value, as well as the likelihood, is given as an interval. A common and simple way of defining the risk level function is to multiply the likelihood with the consequence. Figure 25 shows a fragment of an **R** script for doing this.

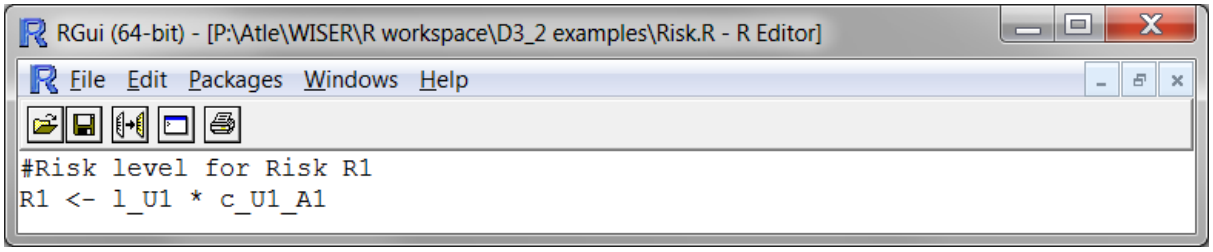


Figure 25 **R** fragment representing a simple risk level calculation

8.1.3 Restrictions on the calculation

Although defining risk level as the product of likelihood and consequence is quite common, it is not the only way to do it. However we choose to do it, increasing the likelihood and consequence should never lead to a reduction of risk level. Therefore, the calculation of risk level should ensure the following:

- The risk level is monotonically increasing in the likelihood and consequence. It does not have to be *strictly* increasing.

Example: For the CORAS fragment in Figure 13, the risk level for the risk of incident U1 harming A1 should be monotonically increasing in l_U1 and c_U1_A1. This is ensured by the script fragment in Figure 25, where the risk level (R1) is defined as the product of l_U and c_U1_A1.

Since we are using intervals rather than exact values, it is not obvious what we mean by monotonically increasing. This is defined in Appendix I.

8.2 Incoming 'leads-to' relations to a node

8.2.1 CORAS representation

The CORAS fragment is identical to Figure 16 presented in Section 7.2.1.

8.2.2 R representation

The computation of the likelihood of the target node is performed in two steps. We start by computing the likelihood contributions for each incoming branch. Then we aggregate all likelihood contributions. For each branch, its likelihood contribution is obtained by multiplying the likelihood of the source node with the conditional likelihood that an occurrence of the source node will lead to the target node. Figure 26 shows an example of a fragment of an **R** script corresponding to the CORAS fragment in Figure 16.

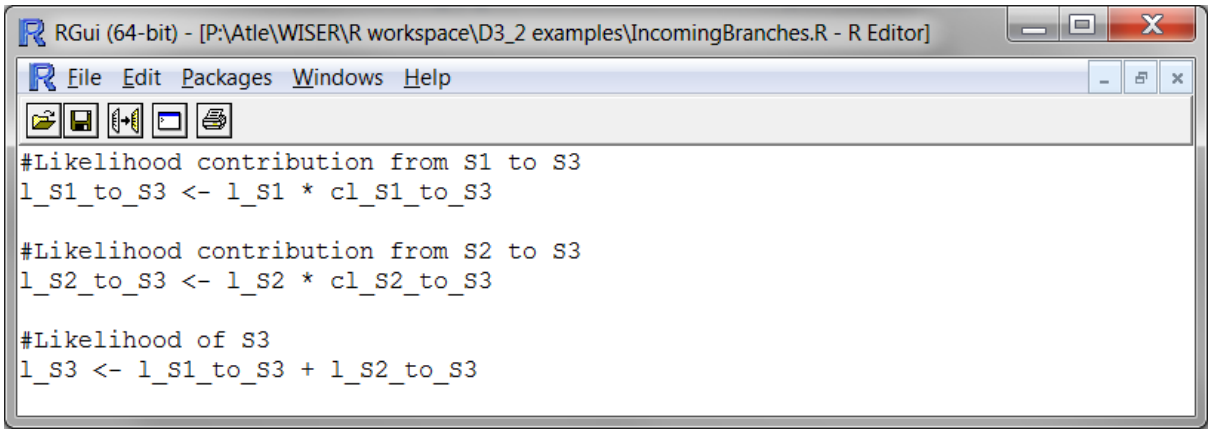


Figure 26 **R** fragment representing incoming 'leads-to' relations

Here, the aggregation of the likelihood contributions from S1 and S2 to S3 is defined by simple addition. This is, however, not the only way to do it, as further explained below.

8.2.3 Restrictions on the calculation

Since the calculation of the likelihood of the target node is performed in two steps, there are also two sets of restrictions on the calculation. For the likelihood contribution from a single source node to the target node, the computation should ensure the following:

- The likelihood contribution from the source node is a sub-interval of the likelihood of the source node multiplied with the conditional likelihood that an occurrence of the source node will lead to the target node.

Example: For the CORAS fragment in Figure 16, the likelihood contribution from S1 to S3 should be a sub-interval of l_S1 * cl_S1_to_S3. This is ensured by the script fragment in Figure 26, where the likelihood contribution l_S1_to_S2 is defined as the product l_S1 * cl_S1_to_S3, which is of course a sub-interval of itself.

This is the most typical way to define the likelihood contribution. However, the restriction also allows functions/algorithms that narrow this interval in order to reduce the uncertainty of the result. It is up to those defining the algorithm to decide whether this is appropriate in each case.

For the aggregation of all the likelihood contributions to a target node, we allow for cases where the joint occurrence of more than one source node will not necessarily lead to more occurrences of the target node. This may be the case, for example, if the source nodes are overlapping, meaning that an occurrence of one source node could also count as an occurrence of another. In such cases, adding up their likelihood contributions could potentially result in a likelihood estimate for the target node that is too high.

Therefore, for the aggregation of likelihood contributions to a source node, the computation should ensure the following:

- The maximum likelihood of the target node is equal to or lower than the sum of the maximum likelihood contributions from all incoming nodes.
- The minimum likelihood of the target node is equal to or higher than the highest of the minimum likelihood contributions from all the incoming nodes.

Example: The calculation of l_S3 should be defined such that the following holds:

- $\max(l\_S3) \leq \max(l\_S1\_to\_S3) + \max(l\_S2\_to\_S3)$
- $\min(l\_S3) \geq \max(\min(l\_S1\_to\_S3), \min(l\_S2\_to\_S3))$

This is ensured by the script fragment in Figure 26. Here, the likelihood of S3 is defined as l_S1_to_S3 + l_S2_to_S3, which means that max(l_S3) = max(l_S1_to_S3) + max(l_S2_to_S3) and min(l_S3) = min(l_S1_to_S3) + min(l_S2_to_S3).

8.3 Indicators attached to a node

8.3.1 CORAS representation

The CORAS fragment is identical to Figure 18 presented in Section 7.3.1.

8.3.2 R representation

Figure 27 shows an example of likelihood assignment to a node based on indicators.

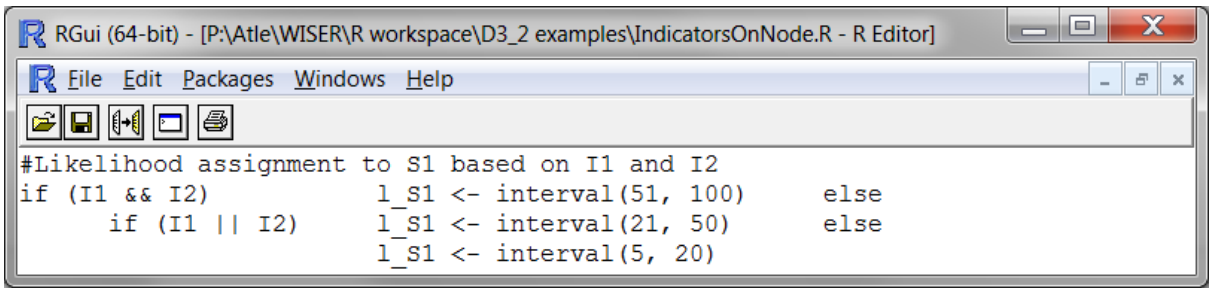


Figure 27 **R** fragment for assignment of node likelihood based on indicators

Two indicators are used in this example: I1 and I2. Both are Boolean and defined such that TRUE indicates a higher likelihood of the threat scenario than FALSE. Therefore, the highest likelihood is assigned to the node S1 if both indicators are TRUE, and the lowest if they are both false.

Due to the wide range of potential threat scenarios and indicators, it is very hard to give general advice on how to define the node likelihood from a set of attached indicators. We therefore recommend that such issues receive a bit of extra attention in the validation of the algorithm.

8.3.3 Restrictions on the calculation

Assuming that all indicators are defined such that a higher value (where FALSE < TRUE for Booleans) indicates a higher likelihood of the node, the likelihood should not decrease due to an increased indicator value. Therefore, the calculation should ensure the following:

- The likelihood of the node is monotonically increasing in all attached indicators.

Example: For the CORAS fragment in Figure 18, the likelihood of S1 should be monotonically increasing in I1 and I2. The two if-then-else statements in the script fragment in Figure 27 ensures that l_S1 has its highest value if both I1 and I2 are TRUE, and its lowest value if both are FALSE, while the middle value is assigned if one of I1 and I2 is TRUE and the other is false.

8.4 Indicators attached to a 'leads-to' relation

8.4.1 CORAS representation

The CORAS fragment is identical to Figure 20 presented in Section 7.4.1.

8.4.2 R representation

Figure 28 shows an example of assignment of conditional likelihood to a 'leads-to' relation based on indicators.

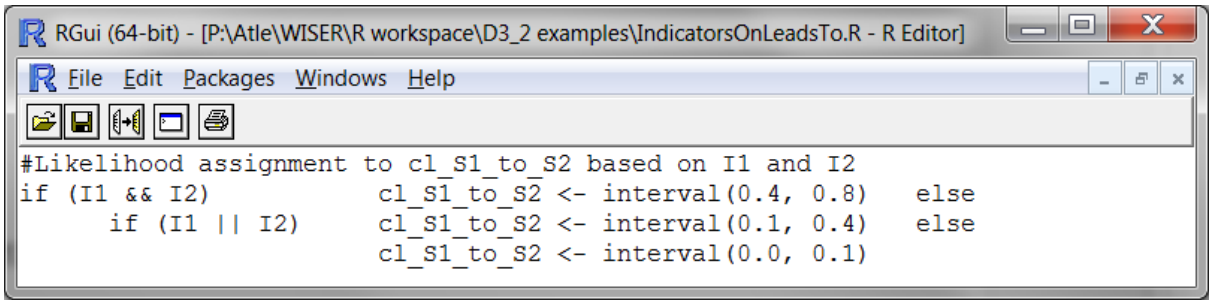


Figure 28 **R** fragment for assignment of conditional likelihood based on indicators

The example is similar to the one for assignment of likelihood to a node in Section 8.3.2, and the assumption made for the indicators there apply also here. Typically, the indicators attached to a 'leads-to' relation will relate to the presence and/or severity of vulnerabilities on the relation.

The main difference between the script fragments in Figure 27 and Figure 28 is that the latter assigns a probability interval rather than a frequency interval; hence, all numbers are between 0 and 1.

Regarding the difficulty of providing general guidelines and the importance of the validation, the same considerations as expressed in Section 8.3.2 apply also when assigning a conditional likelihood on a 'leads-to' relation.

8.4.3 Restrictions on the calculation

Making the same assumptions regarding indicators being defined such that increased indicator values indicate higher likelihood/risk contributions as in Section 8.3.3, the restriction on the calculation is also similar:

- The conditional likelihood of the 'leads-to' relation is monotonically increasing in all attached indicators.

Example: For the CORAS fragment in Figure 20, cl_S1_to_S2 should be monotonically increasing in I1 and I2. This is indeed ensured by the script fragment in Figure 28.

8.5 Mitigation proposal triggering

8.5.1 CORAS representation

The CORAS fragment is identical to Figure 22 presented in Section 7.5.1.

8.5.2 R representation

Figure 29 shows an **R** fragment used to trigger a proposal for implementing mitigation M1.

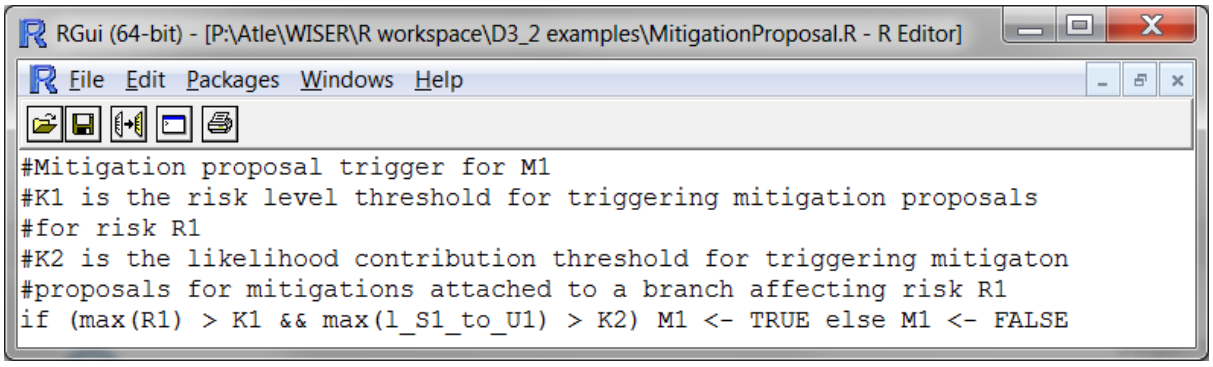


Figure 29 **R** fragment for triggering a mitigation proposal

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). As also stated in Section 7.5.2, 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 22, the only risk that M1 has the potential to reduce is the risk of incident U1 harming asset A1. In the script fragment of Figure 29, the risk level of this risk is represented by R1, which is an interval. The script fragment consists of a single if-then-else statement, where the first conjunct of the conditional corresponds to condition 1) above, and the second conjunct corresponds to condition 2). Notice that the 'sufficiently high' criterion is captured by the thresholds K1 and K2, as explained by the comments in the script fragment. To ensure a precautionary approach, these thresholds are compared to the worst case (maximum) elements of the interval sets.

8.5.3 Restrictions on the calculation

A mitigation proposal should never be turned off due to 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, assuming FALSE < TRUE, the computation of a mitigation proposal (i.e. the assignment of the value FALSE or TRUE) should ensure the following:

- The value of the mitigation proposal should be monotonically increasing in all the risk level assessments and likelihood assessments occurring in the condition of the if-statement. It does not have to be *strictly* increasing.

Example: For the CORAS fragment in Figure 22, the value of M1 should be monotonically increasing in R1 and l_S1_to_U1. This is indeed ensured by the script fragment in Figure 29, as increasing R1 and/or l_S1_to_U1 will never lead to a reduction of max(R1) and/or max(l_S1_to_U1), which occur in the conditional of the if-then-else statement.

9 Economic impact assessment

In this section, we present the WISER approach for economic impact assessment. We identify the factors to be taken into consideration for assessment of economic impact and show the way these factors are structured. In the coming months, we will create **R** scripts for performing economic impact assessments based on the approach presented below.

To estimate the economic impact of a cyber-risk event we will consider four different loss drivers and aggregation rules based on risk correlation.

1) Losses related to digital assets disruption;
2) Losses related to data breaches;
3) Losses related to business interruption;
4) Losses related to the unauthorised transfer of assets or funds.

For each driver we will define the basic information needed to derive a severity curve. The potential loss will be expressed in the form of a “return period”, meaning that WISER will provide an estimation of the potential loss that a firm could reasonably expect in one year, 5 years, 10 years, 20 years and 50 years. These are statistical measures well known and generally adopted by risk managers, and typically used by the insurance market.

9.1 Business interruption

9.1.1 BI Losses

The ICT infrastructure supports business operations in very different ways, like providing ecommerce infrastructures or the control room software in a power plant. A business interruption could potentially have different loss effects:

- Loss of revenues due to business interruption, with the inability to perform business operations. We consider these losses as having a direct impact on the firm P&L. Typical examples are:
  - Inability to take orders;
  - Inability to produce goods;
  - Inability to deliver services or products;
- Claims and Lawsuits deriving by contractual obligations. We consider these losses as having a direct impact on the firm P&L (Profit & Loss) in the form of refunds and legal expenses. Typical examples are:
  - Claims due to inability to deliver products or services;
  - Claims related to breach of service level agreements;
  - Lawsuits related to breach of contracts;
- 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.

9.1.2 Modelling variables

The formalised process to collect model variables will be an integral part of the WISER platform. WISER will provide guidelines, preformatted questionnaires and tutorials to help the Client collecting the model variables reported below.

Regarding the mode of operation: in CyberWiser Essential this material will be self-explicable and no interaction will be required, for CyberWiser Plus, we expect a certain level of interaction and consultancy activity.

9.1.2.1 Recovery time variables.

1) REC_Thres_Rev → Expressed in minutes, represents the minimum BI time necessary for a loss of revenues to be recorded;
2) REC_Thres_Lit → Expressed in minutes, represents the minimum BI time necessary for claims or lawsuits to potentially happen;
3) REC_Thres_Fines → Expressed in minutes, represents the minimum BI time necessary to be subject to a potential regulatory fine;
4) REC_Thres_Rep → Expressed in minutes, represents the minimum BI time necessary for a reputational (economic) loss to manifest itself;
5) Rec_BC → Expressed in minutes, represents the expected recovery time form business continuity or disaster recovery procedures;
6) Rec_Typ → Expressed in hours represents the expected recovery time in typical conditions;
7) Rec_Cat → Expressed in days represents the expected recovery time in worst case conditions;

9.1.2.2 Loss of revenues related variables.

1) BIMV_In → Expressed in €/min represents the monetary loss recorded by the firm in the first 2 hours of BI;
2) BIMV_BD → Expressed in €/h represents the monetary loss recorded by the firm in the 2nd to 8th hours of BI;
3) BIMV_W → Expressed in €/d represents the monetary loss recorded by the firm in the 2nd to 5th working days of BI;
4) BIMV_Cat → Expressed in €/d represents the monetary loss recorded by the firm after the 5th working day of BI;

9.1.2.3 Litigation variables.

1) Lig_Numb_In → Expressed as an integer, represents the number of clients potentially claiming for a BI of less than 2 hours;
2) Lig_Numb_BD → Expressed as an integer, represents the number of clients potentially claiming for a BI in the 2 to 8 hours;
3) Lig_Numb_W → Expressed as an integer, represents the number of clients potentially claiming for a BI in the 2 to 5 days;
4) Lig_Numb_Cat → Expressed as an integer, represents the number of clients potentially claiming for a BI for a block lasting more than 5 days;
5) Lig_Imp_Typ → Expressed in € represents the value of a typical claim (including legal expenses);
6) Lig_Imp_Cat → Expressed in € represents the value of the worst claim (including legal expenses);
7) Lig_Fine_Typ → Expressed in € represents the value of a typical fine related to BI;
8) Lig_Fine_Cat → Expressed in € represents the value of the worst fine related to BI;

9.1.2.4 Reputation variables.

1) Rep_BI_Typ → Expressed in € represents the value of a typical reputational loss related to BI;
2) Rep_BI_Cat → Expressed in € represents the value of the worst reputational loss related to BI;

9.1.2.5 Recovery cost variables.

1) Rec_BI_Typ → Expressed in € represents the value of a typical recovery operation related to BI;
2) Rec_BI_Cat → Expressed in € represents the value of the worst recovery operation related to BI;

9.2 Data breaches

9.2.1 Data breach related losses

In its operations every firm collect, elaborates and store data of different nature. To estimate the potential loss deriving from a data breach WISER will explicitly consider the following families of data:

- Personal data (PII), meaning the basic information needed to identify and get in contact with a physical person:
  - Full name;

- Address;
- Telephone numbers;
- Email address
- …

- Sensible personal data (SPI), meaning information that gives clear indications about political, sexual, ethnical and religious orientations of a physical person;
- Lifestyle personal data, meaning 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, meaning data used to activate payment systems (just as example: credit card data, login and psw for ebanking, a parking payment app …);
- Corporate sensible data, meaning data giving clear indications regarding clients, clients' profitability, market positioning, M&A (Merger & Acquisitions) operations …;
- IP data, meaning data reporting intellectual property of the firm.

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

9.2.2 Modelling variables

The formalised process to collect model variables will be an integral part of the WISER platform. WISER will provide guidelines, preformatted questionnaires and tutorials to help the Client collecting the model variables reported below.

Regarding the mode of operation: in CyberWiser Essential this material will be self-explicable and no interaction will be required, for CyberWiser Plus, we expect a certain level of interaction and consultancy activity.

9.2.2.1 Volume variables.

1) Pers_Vol → Expressed as an integer, indicating the number of records related to basic identification data;
2) Pers_Sens_Vol → Expressed as an integer, indicating the number of records related to sensible data;
3) Pers_LS_Vol → Expressed as an integer, indicating the number of records related to Lifestyle data;
4) Pay_Vol → Expressed as an integer, indicating the number of records related to payment data;
5) Corp_Sens_Vol → Expressed as an integer, indicating the number of records related to corporate sensible data;
6) IP_Vol → Expressed as Yes/No, indicating the presence of IP information recorded in the ICT infrastructure;

9.2.2.2 Loss related variables.

1) Pers_Loss → Expressed in €/rec represents the monetary loss recorded by the firm in case of data unauthorised publication. Includes claims and compensation, mandatory disclosure costs for clients/users, excluding fines;
2) Pers_Sens_Loss → Expressed in €/rec represents the monetary loss recorded by the firm in case of data unauthorised publication. Includes claims and compensation, mandatory disclosure costs for clients/users, excluding fines;

3) Pers_LS_Loss → Expressed in €/rec represents the monetary loss recorded by the firm in case of data unauthorised publication. Includes claims and compensation, mandatory disclosure costs for clients/users, excluding fines;

4) Pay_Loss → Expressed in €/rec represents the monetary loss recorded by the firm in case of data used for unauthorised transactions. Includes claims and compensation, mandatory disclosure costs for clients/users, compensation to payment circuits, excluding fines;

5) Corp_Sens_Loss → Expressed in €/rec represents the monetary loss recorded by the firm in case of data unauthorised publication. Includes claims and compensation, mandatory disclosure costs for clients/users, excluding fines;

6) IP_Loss_Typ → Expressed in € represents the typical monetary loss recorded by the firm in case of unauthorised publication of IP;

7) IP_Loss_Cat → Expressed in € represents the monetary loss recorded by the firm in case of unauthorised publication of IP, in worst case conditions;

8) Fine_Typ_Loss → Expressed in € represents the expected fine in case of data breach for typical conditions;

9) Fine_Typ_Cat → Expressed in € represents the expected fine in case of data breach for worst case conditions;

9.2.2.3 Reputation variables.

1) Rep_DB_Typ → Expressed in € represents the value of a typical reputational loss related to the data breach;

2) Rep_DB_Cat → Expressed in € represents the value of the worst reputational loss related to the data breach;

9.2.2.4 Recovery cost variables.

3) Rec_DB_Typ → Expressed in € represents the value of a typical recovery operation related to the data breach;

4) Rec_DB_Cat → Expressed in € represents the value of the worst recovery operation related the data breach;

9.3 Digital asset disruption

9.3.1 Digital asset disruption related losses

In its operations almost every firm relies on digital assets. 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 an non 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, WISER will consider a digital asset disruption related loss as the cost involved with the asset reconstruction. This will consider:

- Hardware cost;
- Software cost;
- Data recovery cost;
- Personnel cost;
- Forensic cost;
- Outsourcing cost.

Loss amount will not encompass business interruption that is considered separately and potential improvements.

9.3.2 Modelling variables

The formalised process to collect model variables will be an integral part of the WISER platform. WISER will provide guidelines, preformatted questionnaires and tutorials to help the Client collecting the model variables reported below.

Regarding the mode of operation: in CyberWiser Essential this material will be self-explicable and no interaction will be required, for CyberWiser Plus, we expect a certain level of interaction and consultancy activity.

9.3.2.1 Loss related variables.

1) HW_Loss → Expressed in € represents the monetary loss recorded by the firm, due to hardware replacement, if the hardware cannot be recovered after the cyber event;
2) SW_Loss → Expressed in € represents the monetary loss recorded by the firm, due to software replacement, if the software licence cannot be recovered after the cyber event. It doesn't encompass any personnel cost;
3) Pers_HW_Loss → Expressed in € represents the monetary loss recorded by the firm, due to personnel cost, internal only, needed for hardware replacement;
4) Pers_SW_Loss → Expressed in € represents the monetary loss recorded by the firm, due to personnel cost, internal only, needed for software replacement and installation;
5) Data_Recov_Loss → Expressed in € represents the monetary loss recorded by the firm, due to all actions required to recover the disrupted information;
6) For_Loss → Expressed in € represents the monetary loss recorded by the firm, due to forensic costs potentially involved with the disruption;
7) Out_Loss → Expressed in € represents the monetary loss recorded by the firm, due to external personnel costs potentially involved with the disruption;

9.3.2.2 Reputation variables.

3) Rep_DD_Typ → Expressed in € represents the value of a typical reputational loss related to the digital disruption;
4) Rep_DD_Cat → Expressed in € represents the value of the worst reputational loss related to the digital disruption;

9.4 Unauthorised transfer of assets or funds

9.4.1 Unauthorised transfer of assets or funds related losses

Many firms access the payment system through the digital channels. Other firms have assets delivered directly through the digital channels (app, software, services), or use the digital channels to sale or purchase goods.

As an non exhaustive example, the following are intended as assets potentially subject to unauthorised transfer:

- Goods sold on a market place;
- Tickets;
- Apps or software;
- Goods rented online.
- ....

To estimate the potential loss deriving from the unauthorised transfer of assets or funds, WISER will consider potential internal or external frauds, encompassing the violation of ICT security.

9.4.2 Modelling variables

The formalised process to collect model variables will be an integral part of the WISER platform. WISER will provide guidelines, preformatted questionnaires and tutorials to help the Client collecting the model variables reported below. They are assumed to apply for a specific time frame, typically one year.

Regarding the mode of operation: in CyberWiser Essential this material will be self-explicable and no interaction will be required, for CyberWiser Plus, we expect a certain level of interaction and consultancy activity.

9.4.2.1 Volume variables.

1) Bank_Vol → Number of online banking transactions;
2) Asset_Vol → Expressed as an integer, indicating the number of online transactions involving the transfer of assets;

9.4.2.2 Loss related variables.

1) Bank_Ave_Dep → Expressed in € represents the average amount of funds available on online banking accounts;
2) Bank_Max_Dep → Expressed in € represents the maximum amount of funds available on online banking accounts;
3) Asset_Ave → Expressed in € represents the average transaction registered online;
4) Asset_max → Expressed in € represents the maximum transaction registered online;

9.4.2.3 Reputation variables.

5) Rep_DD_Typ → Expressed in € represents the value of a typical reputational loss related to the funds or assets unauthorised transfer;
6) Rep_DD_Cat → Expressed in € represents the value of the worst reputational loss related to the funds or assets unauthorised transfer.

10 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. Although primarily qualitative, this approach employs numerical weights in a way not supported by DEXi. Support for modelling and assessment of societal impact will be offered by the DSS (Decision

Support System). Hence, models for societal impact assessment will be expressed in a dedicated format supported by the DSS.

10.1 Preliminary considerations

The process of evaluating the societal impact of a cyber-risk event comprises a certain number of steps:

- **Step 1:** Identification 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:** For each criterion, generate a utility function, mapping qualitative and quantitative values → (low, medium, high) to (0-10) and vice versa.
- **Step 4:** Give qualitative values to all the criteria and generate the numerical values to the utility functions.
- **Step 5:** 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 (step 3) and the qualitative values of the criteria (step 4). These inputs have to be introduced, and a specific module will take care of performing the computation of the societal impact (a practical example is presented in Section 10.2). This risk societal impact evaluation, once calculated, will be shown to the user in the Decision Support System interface.

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.

10.2 Societal impact classification

Table 4 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	Question
Society	Social cohesion	Does the risk entail social tensions?
	Trust in fellow citizens	Does the risk harm the trust in fellow citizens?
	Emotions	Does the risk provoke fear

		frustration, anger, etc?
	Social alertness	Does the risk produce social alertness?
	Job quality and labour market	Does this risk affect job quality? Does it increase the demand of jobs or the loss of jobs?
	Education	Do people know about this risk?
	Reputation	Does the risk influence the internal and external reputation of the victim?
	Interplay with media	How will the media react to this risk?
	Consumption	Does the risk influence consumption behavior?
	Market & trade relations	Does the risk influence the capacity of the company to compete in the market?
Individual	Motivation	How does the risk influence motivation to work, commit, etc?
	Quality of life / comfort	Does the risk affect quality of life or comfort?
Law	Accountability	How is the impact of the risk from the point of view of company´s accountability?
Rights and ethics	Privacy, personal data and liberty	Does the risk impact in privacy, family life, personal data protection, liberty and security of individuals?
	Freedoms of thought, conscience, religion, expression, information, movement	Does risk compromise the freedoms of thought, conscience, religion, expression, information and movement?
Politics	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?
	Trust	Does the risk affect trust in politics?
Environment	Hidden effects	Does the risk involve any chance of hidden environmental effects?
	Organization	Should reactions of national and international environmental organizations be expected or

		considered?

Table 4 Classification of societal assets with criteria and questions

Societal Assets are categorised as follow:

- Society → This is the largest category, where the most important societal effects 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 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 on the environment, considered as a societal asset, evaluating potential environmental effects and the reaction on environmental organizations.

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 5 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 5 Weights

In order to consider both qualitative and quali-quantitative models, utility functions are provided, as shown in Table 6.

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: 2%	Does the risk harm the trust in fellow citizens?	[L,M-L,M,M-H,H] [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, commit, 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%	How 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 in 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 overview / power over people´s movements, bodies and actions?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	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 6 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.

Table 7 reports an example of default values for a specific risk pattern. Default values will be generated during the course of the project as modelling evolves.

Category	**Criterion**	**Question**	**Utility function**	**Value**
Society: 50%	Social cohesion: 5%	Does the risk entail social tensions?	[L,M-L,M,M-H,H]	M-L: 4

			[1,4,7,9,10]	
	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, commit, 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%	How is the impact of the risk from the point of view of company´s	[L,M-L,M,M-H,H] [3,5,7,9,10]	M-H: 9

		accountability?		
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 and actions?	[L,M-L,M,M-H,H] [3,5,7,9,10]	L: 3
	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 7 An example of default values

If we do not consider any potential override, and we chose the proposed utility function, 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

With an overall impact of 6,527, considering the weighting mechanism introduced by WISER. This value in a 0 to 10 scale will highlight a medium to moderate societal impact related to a specific risk pattern and potential event. This will give the possibility to compare different risks as done in Table 8.

	Risk 1	**Risk 2**
Society	7,54	8,13
Individual	7,4	9
Law	9	10
Rights and ethics	3	4,2
Politics	3,1	6
Environment	2,75	2,75
Overall impact	**6,527**	**7,1635**

Table 8 Comparing risks

11 Conclusions

In this report, we have presented the preliminary version of the 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 a DEXi model and an **R** script resulting from following the guidelines are provided in the appendices.

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. The societal impact assessment combines qualitative assessments with the use of numerical weights, and will be supported by the Decision Support System.

The final version of this report is due in March 2017 (M22). We then expect to present more refined cyber risk modelling guidelines, with fully worked examples, based on the experiences we gather in the coming months, in particular in the context of the full scale pilots, as well as guidelines for target modelling.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[20] 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.

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

Appendix I Monotonically increasing functions on intervals

For closed intervals A and A', we say that A' ≥ A iff the following holds:

$$\max(A') \geq \max(A) \;\&\; \min(A') \geq \min(A)$$

(Notice that this means that there are cases where neither A' ≥ A nor A ≥ A' holds.)

Assume that the type of the function f is given by $f : C \times C \rightarrow C$, where C is the set of all closed intervals. In other words, f takes two closed intervals as input and returns one closed interval as output. Then f is monotonically increasing in its first argument iff the following holds:

$$A' \geq A \Rightarrow f(A',B) \geq f(A,B)$$

Similarly, f is monotonically increasing in its second argument iff the following holds:

$$B' \geq B \Rightarrow f(A,B') \geq f(A,B)$$

The definition generalizes to functions with more than two arguments in the obvious manner.

Appendix II 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 9 Naming conventions for CORAS and DEXi elements

Appendix III Guidelines application example: CORAS diagram

Figure 30 shows the CORAS diagram we use in order to demonstrate the results of applying the guidelines provided in Section 7 and Section 8 on an example. It is the same diagram as presented in Figure 3, except that we have added labels according to the convention presented in Appendix II, in order to make it easier to see the relationship between the CORAS diagram and the corresponding DEXi model and **R** script.

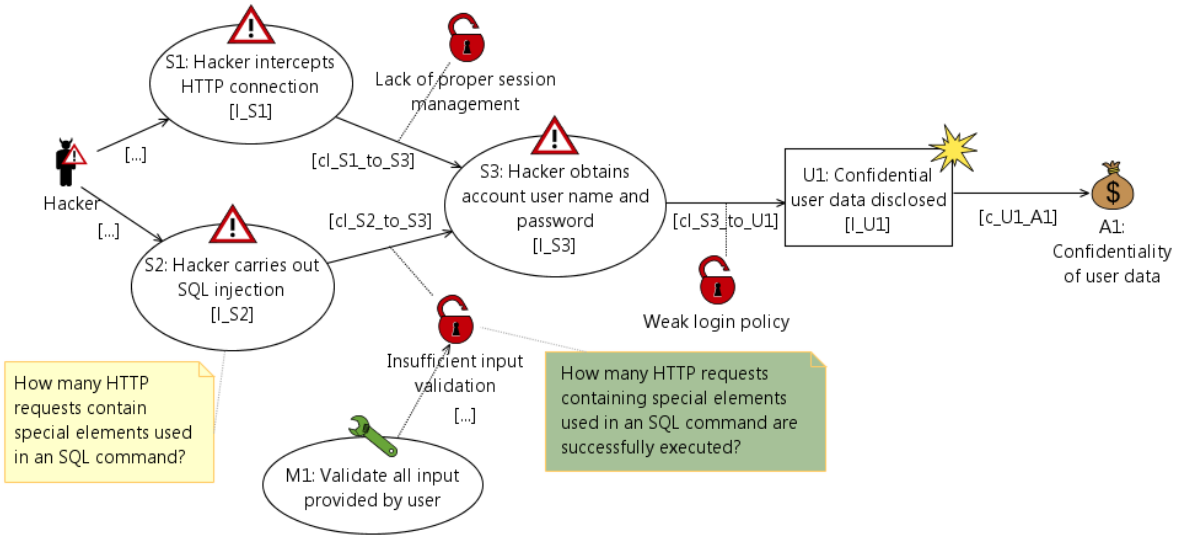


Figure 30 CORAS diagram for guidelines application example

Appendix IV Guidelines application example: DEXi model

We now present an example of a DEXi model obtained by following the guidelines provided in Section 7. 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 31 shows a DEXi model based on the CORAS diagram in Figure 30. 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 7.1.1, the risk is represented by the incident, together with its relation to the asset in the CORAS diagram.

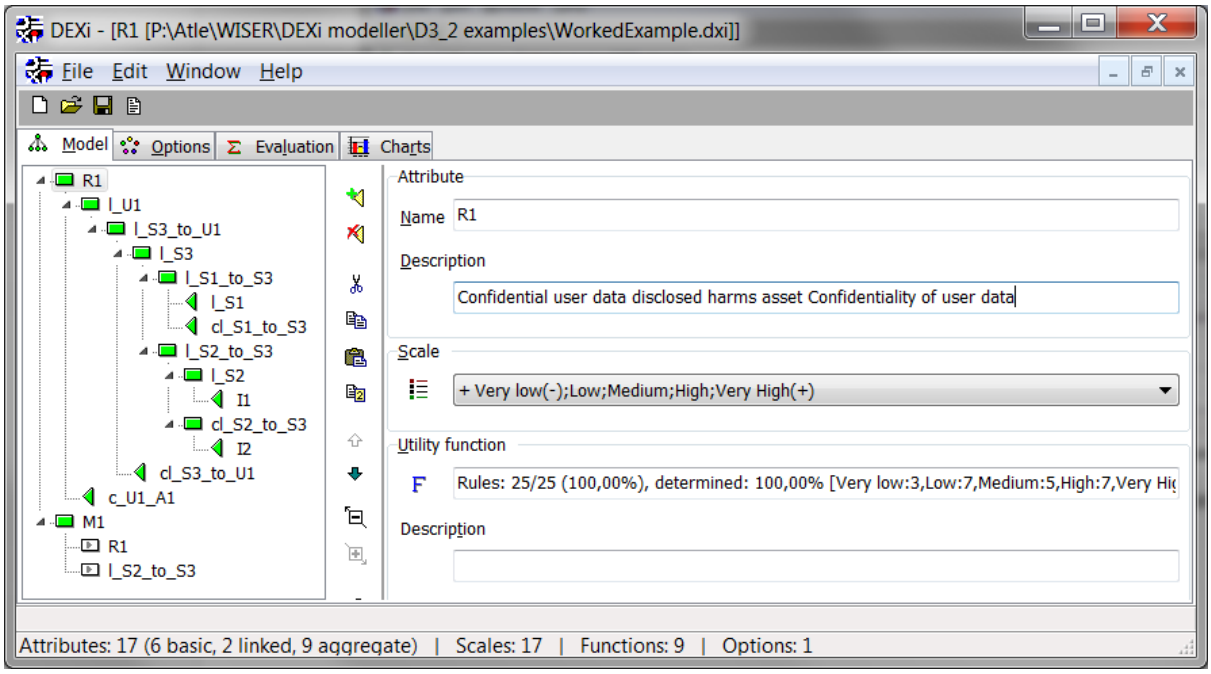
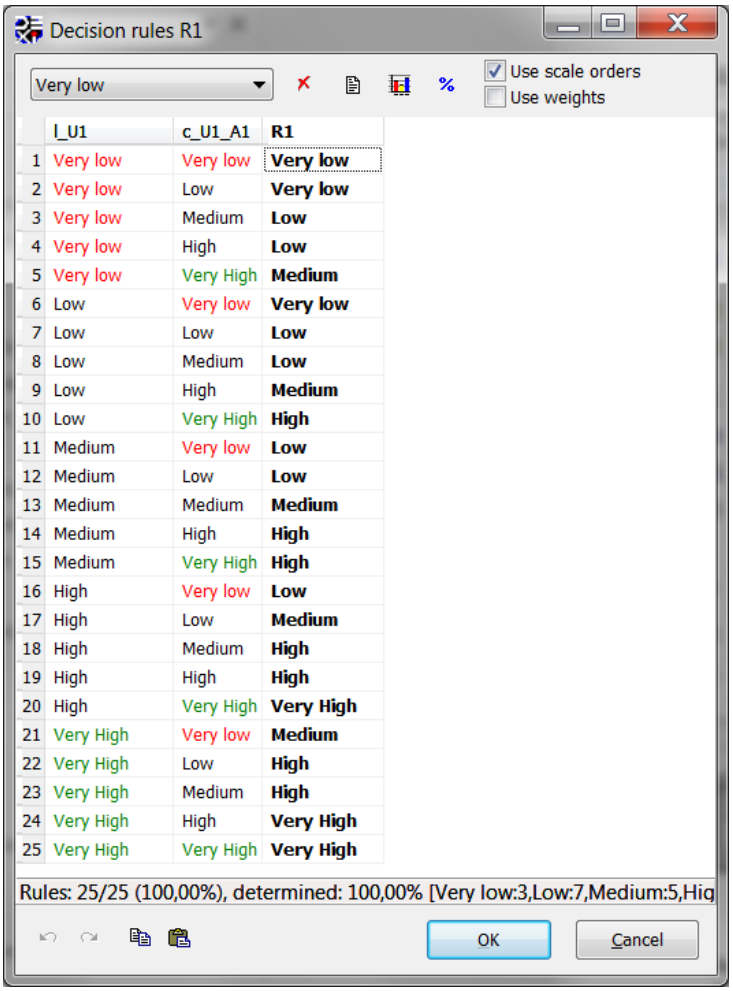


Figure 31 DEXi model obtained from following the guidelines in Section 7

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 32 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 15.



	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 32 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 33. 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.

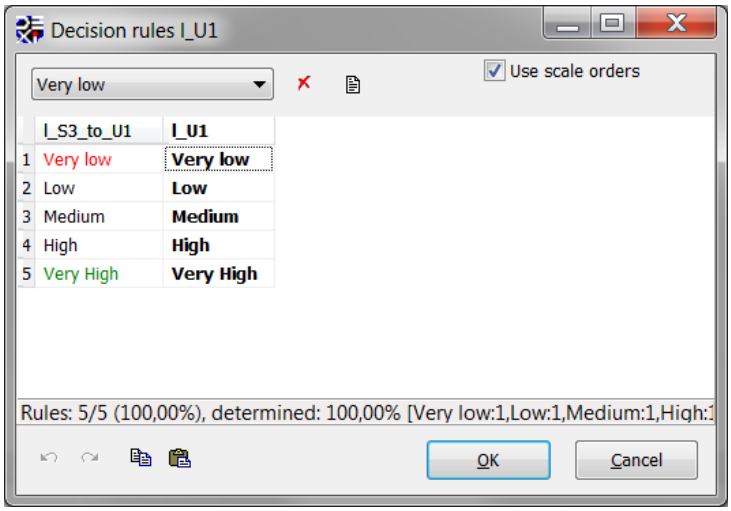


Figure 33 Utility function for l_U1

Likelihood of S3

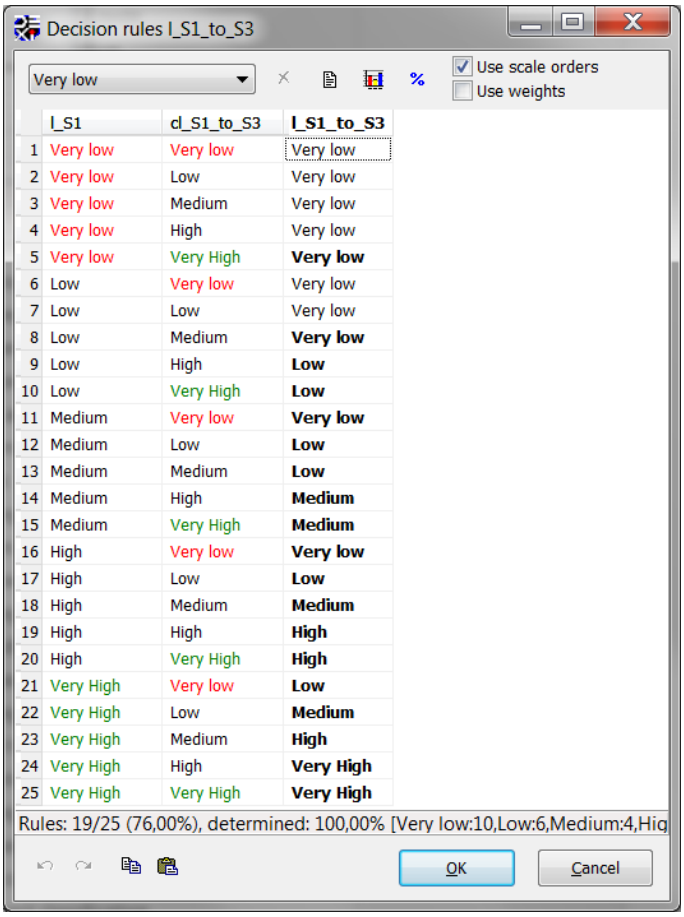
Figure 34 shows the definition of the utility function for l_S3, which depends on the combined likelihood contributions from S1 and S2. The utility function can therefore be viewed as a kind of "qualitative addition", ensuring that l_S3 is always at least as high as the highest of l_S1_to_S3 and l_S2_to_S3. The definition is also monotonically increasing in both arguments.

	l_S1_to_S3	l_S2_to_S3	l_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 34 Utility function for l_S3

Likelihood contribution from S1 to S3

Figure 35 shows the definition of the utility function for l_S1_to_S3, which depends on l_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 l_S1_to_S3 is never higher than l_S1. The definition is also monotonically increasing in both arguments.



	l_S1	cl_S1_to_S3	l_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 35 Utility function for l_S1_to_S3

As explained above, neither l_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 l_S2_to_S3 is identical to the one for l_S1_to_S3, so we do not include it here.

The likelihood contribution l_S2_to_S3 depends on l_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 36 shows the defined scale for I1. The scale is identical for I2.

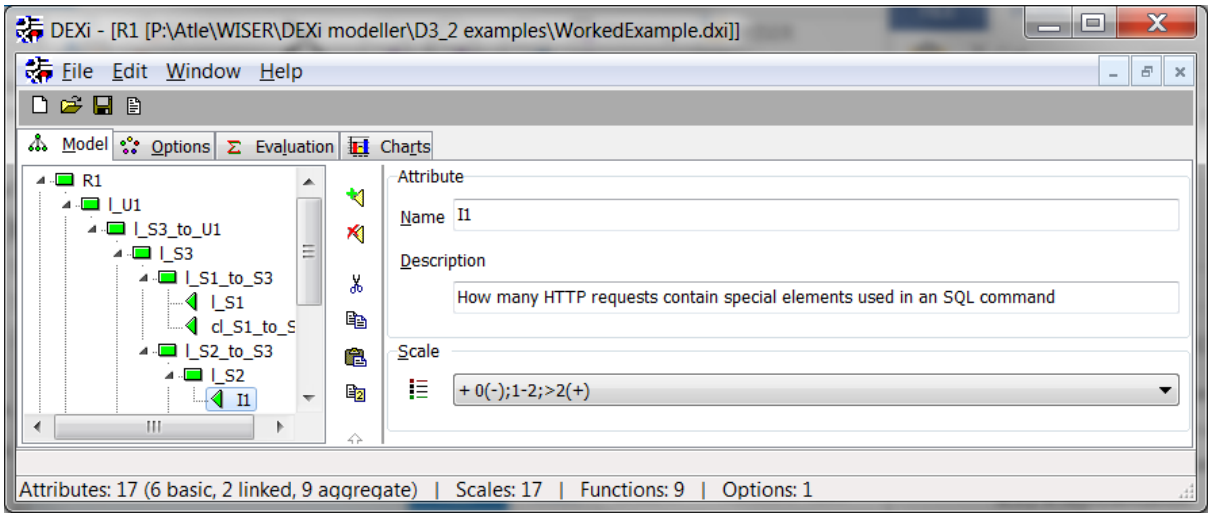


Figure 36 Definition of the scale for an indicator of type integer

Figure 37 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.

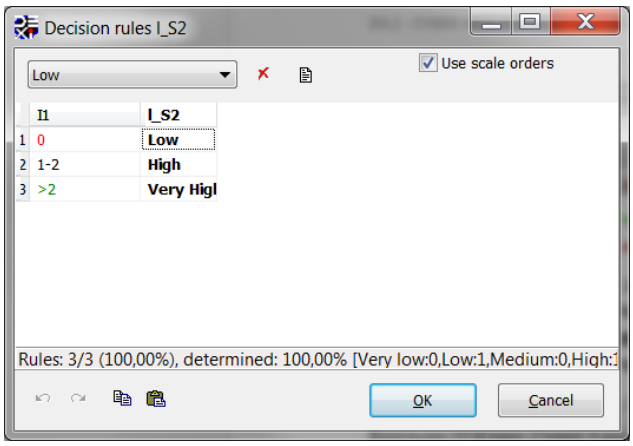


Figure 37 Utility function for I_S2

The definition of the utility function for cl_S2_to_S3 is identical to the one in Figure 37. We therefore do not include it here.

Mitigation triggering

The CORAS diagram in Figure 30 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 31, 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 38 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 38 Utility function for M1

Appendix V Guidelines application example: R script

We now present an example of an **R** script obtained by following the guidelines provided in Section 8. The purpose here is only to illustrate the relation between the CORAS diagram and a corresponding **R** script. The script is not intended to be used for real-life assessments, and has not been validated for this purpose.

Figure 39 shows an **R** script based on the CORAS diagram in Figure 30. Notice that we assume that values have been assigned to the indicators (I1 and I2) and the constants (l_S1, cl_S1_to_S3, cl_S3_to_U1, c_U1_A1, K1 and K2) before the script is executed. The constant K1 represents the risk level threshold for triggering mitigation proposals, while K2 represents the likelihood contribution threshold. See Section 8.5.2 for a further explanation of this.

Unlike DEXi models, which are created in a top-down manner starting with the root nodes, when developing **R** scripts we need to ensure that all variables are assigned their values before being used in further calculations. It is therefore convenient to start from the left-hand side of the CORAS diagram.

The script in Figure 39 is a straightforward implementation of the guidelines. We therefore do not explain each part. However, the assignment of values to l_S2 and cl_S2_to_S3, which are based on the values of I1 and I2, respectively, is worth a comment. In both cases, we have decided to use a simple case-based assignment, distinguishing only between three different cases, in a similar way as we did for the DEXi model in Appendix IV. This is, however not prescribed by the guidelines. How to define the functions from indicator values to the corresponding likelihood values is entirely up to those writing the script, as long as they comply with the restrictions presented in Section 8.3.3 and Section 8.4.3.

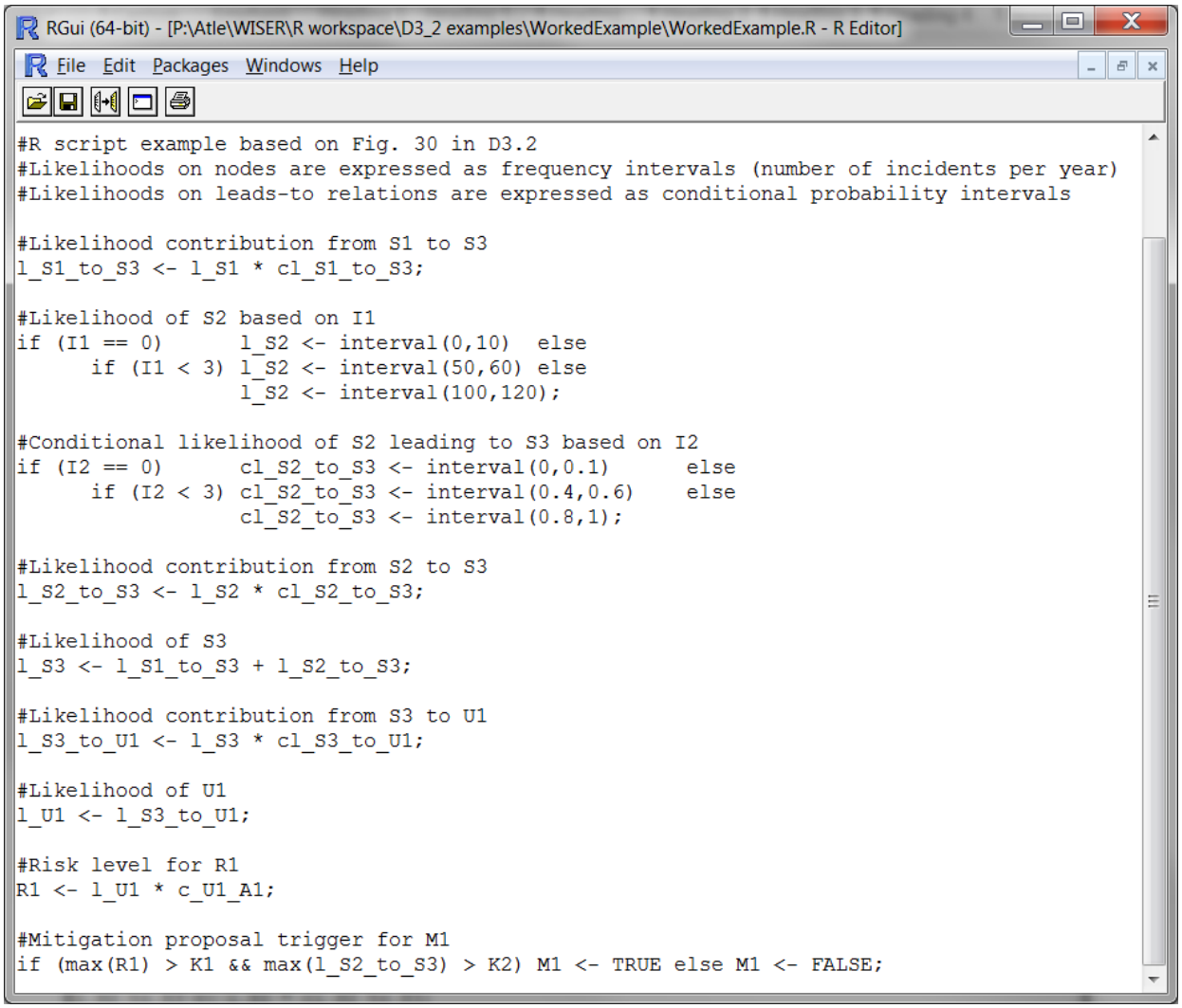


Figure 39 **R** script obtained from following the guidelines in Section 8