STF07 A970 – Unrestricted

REPORT

Structured Semantics for the CORAS Security Risk Modelling Language

Heidi E. I. Dahl, Ida Hogganvik and Ketil Stølen

SINTEF ICT

Cooperative and trusted systems

September 2007

SINTEF

SINTEF ICT

Address: P.O.Box 124, Blindern
0314 Oslo NORWAY
Location: Forskningsveien 1
0373 Oslo
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

Structured Semantics for the CORAS Security Risk Modelling Language

AUTHOR(S)

Heidi E. I. Dahl, Ida Hogganvik and Ketil Stølen

CLIENT(S)

n/a

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
A970	Unrestricted	n/a	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
Unrestricted	978-8214040494	SECURIS(152839/220) S3MS(IST-2006-027004) DIGIT (180052/S10)	34
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.)	CHECKED BY (NAME, SIGN.)
CORAS Semantics.doc		Ketil Stølen	Bjørnar Solhaug
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.)	
	2007-09-24	Bjørn Skjellaug, Forskningssjef	

ABSTRACT

The CORAS security risk modelling language is a graphical language customized for communication, documentation and analysis of security threat and risk scenarios. We present a textual syntax and a structured semantics for each of the five different types of CORAS diagrams, together with step-by-step instructions on how to translate a graphical diagram via the textual syntax into a readable paragraph of English prose, using the structured semantics.

This enables users of the CORAS language to easily extract the precise meaning of a given diagram. The semantics is modular in the sense that the semantics of a diagram can be deduced from the semantics of its constituents.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Security risk analysis	Sikkerhetsanalyse
GROUP 2	Modelling language	Modelleringsspråk
SELECTED BY AUTHOR	Semantics	Semantikk

Structured Semantics for the CORAS Security Risk Modelling Language

Heidi E. I. Dahl, Ida Hogganvik and Ketil Stølen

September 24, 2007

Abstract

The CORAS security risk modelling language is a graphical language customized for communication, documentation and analysis of security threat and risk scenarios. We present a textual syntax and a structured semantics for each of the five different types of CORAS diagrams, together with step-by-step instructions on how to translate a graphical diagram via the textual syntax into a readable paragraph of English prose, using the structured semantics. This enables users of the CORAS language to easily extract the precise meaning of a given diagram. The semantics is modular in the sense that the semantics of a diagram can be deduced from the semantics of its constituents.

Contents

1 Introduction

CORAS is a method for security risk analysis [4]. It comes with a specialized language for communication, documentation and analysis of security threat and risk scenarios. The language was originally defined as a UML [15] profile [14, 16], and has later been customized and refined in several aspects, based on experiences from industrial case studies, and by empirical investigations documented in [5], [6] and [7].

The CORAS language is in particular intended to support brainstorming sessions used to identify and estimate security risks. Such brainstorming sessions are characterized by the involvement of people with thorough knowledge of specific, but only partly overlapping aspects of the target of analysis. Typical participants are the intended users of the target, its designers, developers, and relevant decision makers. These people normally have quite different backgrounds and it may be difficult for the analysts to make them work well together as a group. Our experiences indicate that the CORAS language improves both the efficiency of the analysis process and the quality of the results.

We claim that our graphical approach to security risk modelling contributes to solving three issues related to security analysis:

- *How to facilitate communication in a group consisting of people with different backgrounds and competences:* Our aim has been to provide the participants with a means of communication that covers both technical and more high-level information, without being too complicated to understand. Offering a common basis for communication will hopefully reduce misunderstandings and thereby give a more correct risk picture.
- *How to estimate the likelihoods and consequences of identified risks:* In practice, reliable data on which this can be based is often not available. The participants must use their expert knowledge, experience and familiarity with the domain to estimate both the likelihoods and the consequences of incidents that might not have happened yet. Our aim has been to offer a structured, graphical risk picture to make the complexity more manageable. A graphical representation may illustrate who or what caused the incidents and the weaknesses in the system that made them possible.
- *How to document the security analysis in a comprehensible manner:* The findings of a security analysis constitute vital information not only to the participants in the analysis, but also to the organization as a whole. Our aim has been to define a documentation method that should be more or less self-explanatory, and not rely on extensive training to be understood.

Although we have aimed at making a language that is easily understandable, situations in which the intended meaning of a construct or an expression needs further explanation are bound to arise. The main contribution of this report is the definition of a structured semantics aiming to fulfil this need. The semantics takes an arbitrary CORAS diagram and delivers its intended meaning as a readable paragraph of English. It is structured in the sense that it comes with step-by-step instructions allowing the translation to be conducted automatically. The semantics has been developed to meet the following success criteria:

1. *The translation from CORAS diagrams to English should be modular.* If we add new relations and/or elements to a diagram we have already translated, the translation of the modified diagram is the union of the translation of the original diagram with the translation of the new relations and/or elements.
2. *The resulting paragraph should be understandable English.* The purpose of the translation is to provide a description, in English, of a CORAS diagram, in order to communicate the meaning of the diagram to those not familiar with the intended meaning of the various elements and relations of the CORAS language.

3. *The translation should be easy to perform.* Anyone, even someone unacquainted with CORAS diagrams, should be able to translate a CORAS diagram into English.

4. *The translation should be possible to automate.* Automatic translation is a feature that will be implemented in the CORAS tool in the future (see `http://coras.sourceforge.net` for downloads and documentation).

5. *It should be possible to translate inconsistent diagrams, and the translation should enable the user to identify inconsistencies.* Inconsistent diagrams should still be possible to translate, and the resulting paragraph in English should be sufficiently clear to allow the user to identify the cause of the inconsistency.

The next section introduces the different kinds of CORAS diagrams. Section 3 presents the notation and naming conventions used in the textual syntax and structured semantics. Sections 4 - 8 present the syntax and structured semantics of each of the five types of CORAS diagrams. Each of these sections is self-contained and includes instructions on how to translate an arbitrary syntactically correct diagram into English. An example diagram is given as a presentation of the graphical syntax of each diagram, which contains all possible relations. However, in order to keep the examples to a manageable size, not all permutations of optional elements such as likelihoods and treatment categories have been included. In the cases where one CORAS diagram type is based on another, we still give a complete definition of its textual syntax, translation rules from graphical to textual syntax, and from the textual syntax into English. At the end of each section we present an example of a translation.

2 The CORAS Language

The CORAS language originates from a UML profile developed as a part of the EU funded research project CORAS (IST-2000-25031) [1] (`http://coras.sourceforge.net`). As a result of our work to satisfy the modelling needs in a security risk analysis, the language and its guidelines have evolved into a more specialized and refined approach. The language is meant to support the analyst during the security risk analysis, and serves different purposes in each phase of the analysis. A security risk analysis is normally structured into five phases: (1) context establishment, (2) risk identification, (3) risk estimation, (4) risk evaluation and (5) treatment identification [3].

In the context establishment we employ *assets overview diagrams* to specify the parties of the security analysis and their assets. The purpose is to obtain a precise definition of what the valuable aspects of the target of analysis are, and which are the most important. From empirical investigations [5] and field trials we know that asset identification and valuation is very difficult, and that mistakes or inaccuracies made there may jeopardize the value of the whole security analysis.

During risk identification we use *threat diagrams* to identify and document how vulnerabilities may be exploited by threats to initiate unwanted incidents, and which assets the unwanted incidents affect. The threat diagrams give a clear and easily understandable overview of the risk picture and make it easier to see who or what the threat is, how the threat works (threat scenarios) and which vulnerabilities and assets they involve.

The threat diagrams are used as input to the risk estimation phase, where unwanted incidents are assigned likelihood estimates and possible consequences. The likelihood estimation is often a difficult task, but illustrating the unwanted incidents in the correct context has proved very helpful in practice.

After the risk estimation, the magnitude of each risk can be calculated on the basis of its likelihood and consequence, and modelled in *risk diagrams.* The risk diagrams specify which threats initiate the different risks, and exactly which assets they may harm. This risk representation is then compared to predefined risk tolerance levels to decide which ones need treatments.

In the treatment identification, the threat diagrams containing the risks that cannot be tolerated are used as basis for treatment identification. In this phase the appropriate treatments are identified and modelled in *treatment diagrams*. The resulting treatment diagrams can be seen as a plan for how to deal with the identified risks.

Communicating the results of an analysis in such a way that they are well understood by decision makers can be challenging. The CORAS language supports this by offering *treatment overview diagrams*. Treatment overview diagrams may for example be used to provide a high level summary when presenting the main findings from an analysis.

To summarize, the CORAS language consists of five different kinds of diagrams: asset diagrams, threat diagrams, risk diagrams, treatment diagrams and treatment overview diagrams. Their basic building blocks are presented in Figure 1.

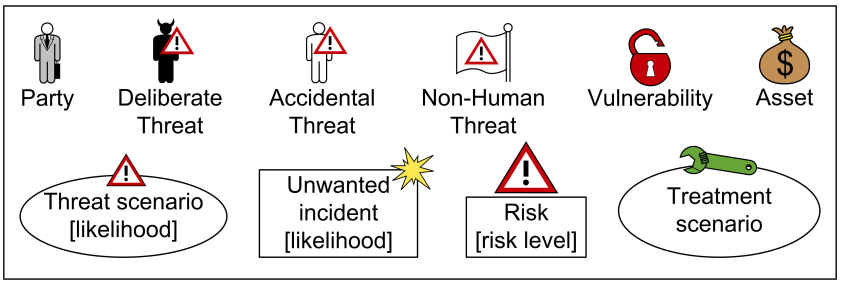


Figure 1: Basic building blocks of the CORAS diagrams

3 The Structured Semantics

The structured semantics for the CORAS language is divided into two separate steps:

(A) The translation of a diagram into its textual syntax, and

(B) The translation of the textual syntax into its meaning as a paragraph in English.

Hence, the semantics enables the user of CORAS to extract the meaning of an arbitrary CORAS diagram by applying first (A), then (B).

Both these steps, and therefore the structured semantics, are modular: a diagram is translated relation by relation, vertex by vertex.

To characterize the textual syntax for the CORAS language, we use ISO's standardized Extended BNF notation [10]. The vertical bar _|_ represents options, braces {_} (respectively $\{\_\}^-$) means an ordered sequence of zero (respectively one) or more repetitions of the enclosed element, and square brackets [_] denotes optional features. The terminal operators are surrounded by quotes: '_'.

To improve readability, we do not resolve all the EBNF definitions completely into terminal operators, but stop at the level of *identifier*, *linguistic term* and *numerical value*. An identifier is a name or a natural language description of an element, i.e. a finite sequence of upper- and lower-case letters, numbers, spaces and punctuation marks. Its precise definition in EBNF is

$$\begin{aligned} \textit{identifier} &= \{\textit{upper-case letter} | \textit{lower-case letter} | \textit{digit} | \text{`\ '} | \textit{punctuation mark}\}^- ; \\ \textit{uppercase letter} &= \text{'A'} | \text{'B'} | \text{'C'} | \ldots \\ \textit{lowercase letter} &= \text{'a'} | \text{'b'} | \text{'c'} | \ldots \\ \textit{punctuation mark} &= \text{'.'} | \text{','} | \text{':'} | \ldots \\ &\ldots \end{aligned}$$

Linguistic terms are identifiers which are members of a totally ordered set, for example likelihoods given as 'high', 'medium' and 'low'. Numerical values are just that: numbers allowing precise calculations of likelihoods, consequences and risk values.

In the translation rules, we use $\Rightarrow$ to denote the translation from graphical to textual syntax.

The translation rules for the vertices and annotations of the diagrams are given by the naming conventions in Table 1.

Vertex	**Instance**
party	p
asset	a
deliberate threat	dt
accidental threat	at
non-human threat	nht
threat scenario	ts
unwanted incident	ui
risk	r
treatment scenario	trs

Annotation	**Instance**
vulnerability	$v = \{v\} = V_1$
vulnerability set	$V_n = \{v_1, \ldots, v_n\}$
likelihood	l
consequence	c
risk value	rv
risk function	rf

Table 1: Naming conventions

When there are more than one instance of a type, we use subscripts, e.g. dt_1, dt_2 for two deliberate threats. The translation from the textual syntax into English is defined by a function $[\![\_]\!]$ on textual expressions. We use _ := _ to denote "defined as".

4 Asset Diagrams

Asset diagrams are used to give an overview of the assets within the scope of the security risk analysis, and to show how harm to one asset may affect other assets. They also present the parties involved in the analysis, and their interest in the assets. The asset diagram has two basic building blocks, parties and assets, shown in Figure 2. We refer to the parties and assets of an asset diagram

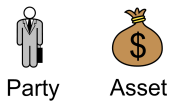


Figure 2: Basic building blocks of asset diagrams

as its *vertices*. The vertices may be connected by two kinds of ordered relations: the *protect* and the *indirect harm* relations. The protect relation displays the relationship inherent in the definition of an asset; i.e. that the asset is something whose value a party wants to protect. An important part of this is the level of risk that a party is willing to accept with respect to the asset in question, therefore we have the option to annotate the relation with the maximum acceptable risk level. The indirect harm relation captures dependency between two assets, in the sense that harm to the first asset may indirectly harm the other.

To summarize, asset diagrams may contain protect relations between parties and assets, and indirect harm relations between assets. Figure 3 shows the graphical syntax of asset diagrams.

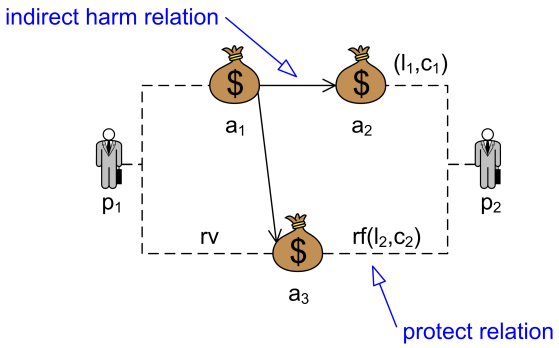


Figure 3: Graphical syntax of asset diagrams

4.1 Textual Syntax of Asset Diagrams

$$
\begin{aligned}
\mathit{diagram} &= (\{\mathit{vertex}\}^{-}, \{\mathit{relation}\});\\
\mathit{vertex} &= \mathit{party} \mid \mathit{asset};\\
\mathit{relation} &= \mathit{protect} \mid \mathit{indirect\ harm};\\
\mathit{protect} &= \mathit{party} \overset{[\mathit{risk\ level}]}{\cdots} \mathit{asset};\\
\mathit{indirect\ harm} &= \mathit{asset} \rightarrow \mathit{asset};\\
\mathit{party} &= \mathit{identifier};\\
\mathit{asset} &= \mathit{identifier};\\
\mathit{risk\ level} &= \mathit{risk\ value} \mid \mathit{risk\ function}(\mathit{likelihood}, \mathit{consequence}) \mid\\
&\quad (\mathit{likelihood}, \mathit{consequence});\\
\mathit{risk\ value} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{risk\ function} &= \mathit{identifier};\\
\mathit{likelihood} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{consequence} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};
\end{aligned}
$$

4.2 Translation from Graphical to Textual Syntax

We translate an asset diagram into the textual syntax schematically, vertex by vertex and relation by relation. The set of textual expressions we obtain is the translation of the complete diagram.

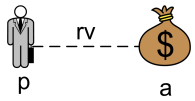
4.2.1 Vertices

The translation of a vertex in the textual syntax is the label of the icon representing it in the graphical syntax. For example, the translation of the set of vertices in Figure 3 is the set

$$\{p_1, p_2, a_1, a_2, a_3\}.$$

Note that in this translation, the type of each node is preserved through the naming conventions from Table 1. When the naming convention is not used we need to type each vertex, for example in a type table, or decompose the diagram into three sets: sets of parties, sets of assets and sets of relations.

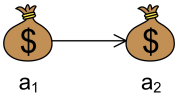
4.2.2 Protect relation



$\Rrightarrow\ p \overset{rv}{\cdots} a$

The translation is similar when the relation is annotated with a risk function or a likelihood and consequence pair instead of a risk value. In either case, replace rv with the appropriate annotation. If the protect relation is not annotated, remove rv altogether.

4.2.3 Indirect harm relation



$\Rrightarrow\ a_1 \rightarrow a_2$

4.3 Translation from the Textual Syntax to English

4.3.1 Diagram

A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0, m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation.

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

4.3.2 Vertices

$$[\![p]\!] := \boldsymbol{p} \text{ is a party.}$$
$$[\![a]\!] := \boldsymbol{a} \text{ is an asset.}$$

4.3.3 Protect relation

$$[\![p \cdots a]\!] := \boldsymbol{p} \text{ wants to protect the value of } \boldsymbol{a}.$$
$$[\![p \overset{rl}{\cdots} a]\!] := \boldsymbol{p} \text{ wants to protect the value of } \boldsymbol{a}, \text{ but accepts } [\![rl]\!] \text{ or less.}$$

The second expression is the translation of a protect relation annotated with a maximum acceptable risk level. The rl for risk level is replaced by either rv, $rf(l, c)$ or (l, c) depending on how the risk level is formulated.

4.3.4 Indirect harm relation

$$[\![a_1 \rightarrow a_2]\!] := \boldsymbol{a_2} \text{ is harmed indirectly via } \boldsymbol{a_1}.$$

4.3.5 Annotations

$$
\begin{aligned}
[\![\, rv \,]\!] &:= \text{risk value } \boldsymbol{rv} \\
[\![\, rf(l,c) \,]\!] &:= \text{risk function } \boldsymbol{rf} \text{ of } [\![\, (l,c) \,]\!] \\
[\![\, (l,c) \,]\!] &:= [\![\, l \,]\!] \text{ and } [\![\, c \,]\!] \\
[\![\, l \,]\!] &:= \text{likelihood } \boldsymbol{l} \\
[\![\, c \,]\!] &:= \text{consequence } \boldsymbol{c}
\end{aligned}
$$

4.4 Example

We illustrate the translation of asset diagrams with the following example:

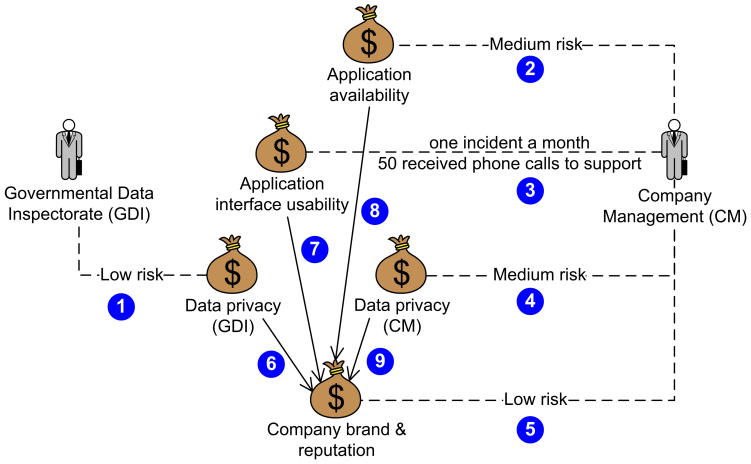


Figure 4: Asset diagram

First we translate the vertices of the diagram: the parties and assets.

1. ***Governmental Data Inspectorate (GDI)*** is a party.
2. ***Company Management (CM)*** is a party.
3. ***Application availability*** is an asset.
4. ***Application interface usability*** is an asset.
5. ***Data privacy (GDI)*** is an asset.
6. ***Data privacy (CM)*** is an asset.
7. ***Company brand & reputation*** is an asset.

We then translate the diagram relation by relation in the order suggested in Figure 4:

1. ***Governmental Data Inspectorate (GDI)*** wants to protect the value of ***Data privacy (GDI)***, but accepts risk value ***Low risk*** or less.
2. ***Company Management (CM)*** wants to protect the value of ***Application availability***, but accepts risk value ***Medium risk*** or less.

3. ***Company Management (CM)*** wants to protect the value of ***Application interface usability***, but accepts likelihood ***one incident a month*** and consequence ***50 received phone calls to support*** or less.
4. ***Company Management (CM)*** wants to protect the value of ***Data privacy (CM)***, but accepts risk value ***Medium risk*** or less.
5. ***Company Management (CM)*** wants to protect the value of ***Company brand & reputation***, but accepts risk value ***Low risk*** or less.
6. ***Company brand & reputation*** is harmed indirectly via ***Data privacy (GDI)***.
7. ***Company brand & reputation*** is harmed indirectly via ***Application interface usability***.
8. ***Company brand & reputation*** is harmed indirectly via ***Application availability***.
9. ***Company brand & reputation*** is harmed indirectly via ***Data privacy (CM)***.

We may choose to structure the sentences further to improve readability:

1. ***Governmental Data Inspectorate (GDI)*** wants to protect the value of ***Data privacy (GDI)***, but accepts risk value ***Low risk*** or less.

2. - 5. ***Company Management (CM)*** wants to protect the value of

- ***Application availability***, but accepts risk value ***Medium risk*** or less.
- ***Application interface usability***, but accepts likelihood ***one incident a month*** and consequence ***50 received phone calls to support*** or less.
- ***Data privacy (CM)***, but accepts risk value ***Medium risk*** or less.
- ***Company brand & reputation***, but accepts risk value ***Low risk*** or less.

6. - 9. ***Company brand & reputation*** is harmed indirectly via

- ***Data privacy (GDI)***.
- ***Application interface usability***.
- ***Application availability***.
- ***Data privacy (CM)***.

Note that the order of the sentences does not affect the translation. In other words, the sentences commute. The order we choose is merely a convention in order to provide a clear presentation.

5 Threat Diagrams

A threat diagram is a presentation of the chains of events initiated by threats, that have consequences for the assets. Its basic building blocks are the seven elements of Figure 5: deliberate, accidental and non-human threats, vulnerabilities, threat scenarios, unwanted incidents, and assets. Threat scenarios and unwanted incidents may be assigned a likelihood. We refer to the threats, threat scenarios, unwanted incidents and assets of a threat diagram as its *vertices*. Vulnerabilities and likelihoods are used as annotations. Threat diagrams have three different relations: *initiate*, *leads to* and *impact*. A threat exploits one or more vulnerabilities, initiating a threat scenario or an unwanted incident. This is captured by the binary initiate relation. The threat scenario or unwanted incident may then lead to new threat scenarios or unwanted incidents, either as direct consequences or by the threat exploiting other vulnerabilities. This is captured by the binary leads

Figure 5: Basic building blocks of threat diagrams

to relation. Either of these relations may be assigned a likelihood and/or one or more vulnerabilities. The impact relation relates an unwanted incident and an asset that is harmed as a consequence of this incident. Each impact relation may be assigned a consequence.

Figure 6 presents the graphical syntax of a threat diagram.

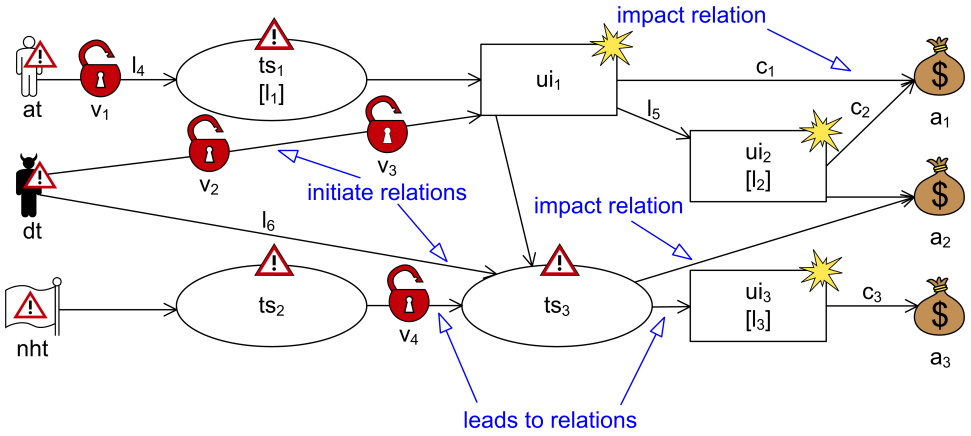


Figure 6: Graphical syntax of threat diagrams

5.1 Textual Syntax of Threat Diagrams

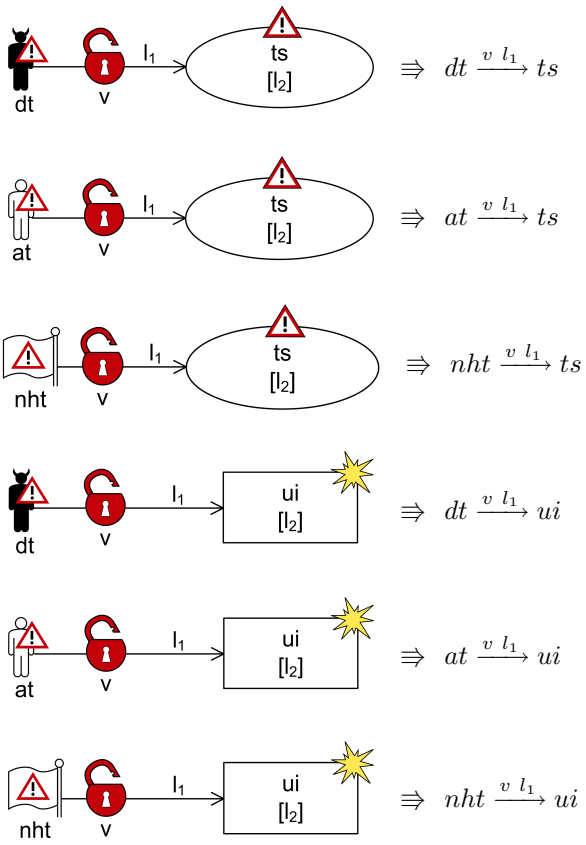
$$
\begin{aligned}
diagram &= (\{vertex\}^{-}, \{relation\});\\
vertex &= threat \mid threat\ scenario \mid unwanted\ incident \mid asset;\\
relation &= initiate \mid leads\ to \mid impact;\\
initiate &= threat \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\qquad threat \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident;\\
leads\ to &= threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\qquad threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident \mid\\
&\qquad unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\qquad unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident;\\
impact &= unwanted\ incident \xrightarrow{[consequence]} asset \mid\\
&\qquad threat\ scenario \rightarrow asset;\\
threat &= deliberate\ threat \mid accidental\ threat \mid non\text{-}human\ threat;\\
deliberate\ threat &= identifier;
\end{aligned}
$$

$$
\begin{aligned}
\textit{accidental threat} &= \textit{identifier};\\
\textit{non-human threat} &= \textit{identifier};\\
\textit{vulnerability set} &= \{\textit{vulnerability}\}^{-};\\
\textit{vulnerability} &= \textit{identifier};\\
\textit{threat scenario} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{unwanted incident} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{asset} &= \textit{identifier};\\
\textit{likelihood} &= \textit{linguistic term} \mid \textit{numerical value};\\
\textit{consequence} &= \textit{linguistic term} \mid \textit{numerical value};
\end{aligned}
$$

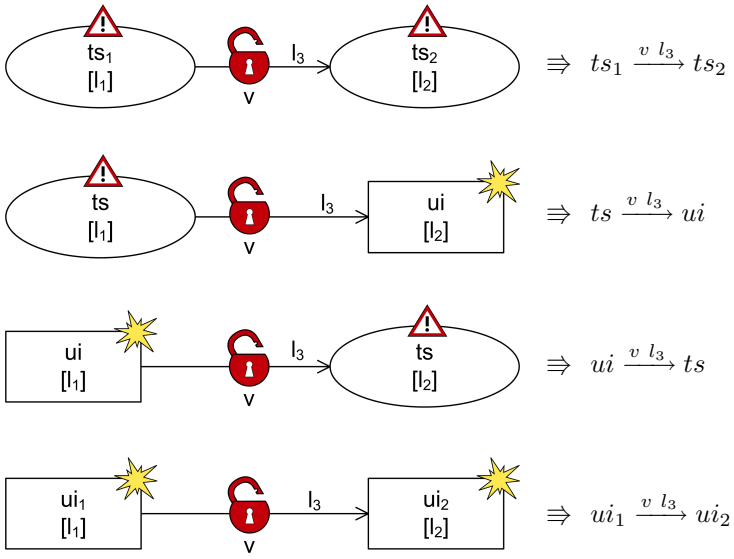
5.2 Translation from Graphical to Textual Syntax

See Section 4 for instructions on how to translate the vertices. Here we present the translation rules including all optional elements. We then explain how the rules may be modified if one or more of the optional elements are removed. To simplify the presentation of the translation rules, they contain only one vulnerability, as there is no icon for vulnerability sets. In order to translate a relation with more than one vulnerability, replace the v in the textual expression of the rule with the set of vulnerabilities.

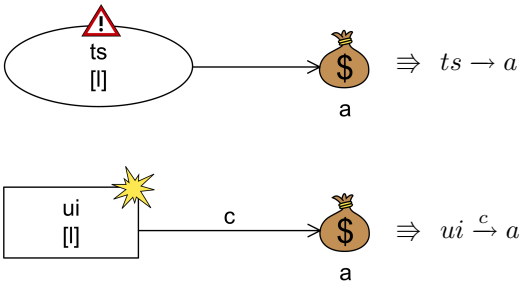
5.2.1 Initiate relation



5.2.2 Leads to relation



5.2.3 Impact relation



The optional elements of the threat diagram are vulnerability sets, likelihoods, and consequences. In the case where a vulnerability or consequence is absent in the graphical expression, simply remove the corresponding element from the textual expression (see Table 1 for an overview of naming conventions). The translation of the relation is the same whether or not likelihoods have been assigned to its vertices.

5.3 Translation from the Textual Syntax to English

5.3.1 Diagram

A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0, m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation.

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

5.3.2 Vertices

$$
\begin{aligned}
[\![\, dt \,]\!] &:= \textbf{\textit{dt}} \text{ is a deliberate threat.} \\
[\![\, at \,]\!] &:= \textbf{\textit{at}} \text{ is an accidental threat.} \\
[\![\, nht \,]\!] &:= \textbf{\textit{nht}} \text{ is a non-human threat.} \\
[\![\, a \,]\!] &:= \textbf{\textit{a}} \text{ is an asset.} \\
[\![\, ts \,]\!] &:= \text{Threat scenario } \textbf{\textit{ts}} \text{ occurs with undefined likelihood.} \\
[\![\, ts(l) \,]\!] &:= \text{Threat scenario } \textbf{\textit{ts}} \text{ occurs with } [\![\, l \,]\!]. \\
[\![\, ui \,]\!] &:= \text{Unwanted incident } \textbf{\textit{ui}} \text{ occurs with undefined likelihood.} \\
[\![\, ui(l) \,]\!] &:= \text{Unwanted incident } \textbf{\textit{ui}} \text{ occurs with } [\![\, l \,]\!].
\end{aligned}
$$

5.3.3 Initiate relation

The semantics for the initiate relation is independent of which kind of threat it originates from and whether it ends in a threat scenario or an unwanted incident. In the following definitions the threat t may be replaced by dt, at or nht, for deliberate, accidental and non-human threats respectively. The ts representing a threat scenario may be replaced with an unwanted incident ui.

$$
\begin{aligned}
[\![\, t \rightarrow ts \,]\!] &:= \textbf{\textit{t}} \text{ initiates } \textbf{\textit{ts}} \text{ with undefined likelihood.} \\
[\![\, t \xrightarrow{l} ts \,]\!] &:= \textbf{\textit{t}} \text{ initiates } \textbf{\textit{ts}} \text{ with } [\![\, l \,]\!]. \\
[\![\, t \xrightarrow{V_n} ts \,]\!] &:= \textbf{\textit{t}} \text{ exploits } [\![\, V_n \,]\!] \text{ to initiate } \textbf{\textit{ts}} \text{ with undefined likelihood.} \\
[\![\, t \xrightarrow{V_n\ l} ts \,]\!] &:= \textbf{\textit{t}} \text{ exploits } [\![\, V_n \,]\!] \text{ to initiate } \textbf{\textit{ts}} \text{ with } [\![\, l \,]\!].
\end{aligned}
$$

5.3.4 Leads to relation

As for the initiate relation, the semantics for the leads to relation is the same whether or not it originates and/or ends in a threat scenario or unwanted incident. The definitions below are given for a leads to relation between threat scenarios. For the other combinations replace one or more ts with *ui(l)* as before.

$$
\begin{aligned}
[\![\, ts_1 \rightarrow ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with undefined likelihood.} \\
[\![\, ts_1 \xrightarrow{l} ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with } [\![\, l \,]\!]. \\
[\![\, ts_1 \xrightarrow{V_n} ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with undefined likelihood, due to } [\![\, V_n \,]\!]. \\
[\![\, ts_1 \xrightarrow{V_n\ l} ts_2 \,]\!] &:= \textbf{\textit{ts}}_1 \text{ leads to } \textbf{\textit{ts}}_2 \text{ with } [\![\, l \,]\!], \text{ due to } [\![\, V_n \,]\!].
\end{aligned}
$$

5.3.5 Impact relation

As for the two previous relations, the semantics for the unannotated impact relation is the same independent of whether it originates in a threat scenario or an unwanted incident.

$$[\![\, ts \rightarrow a \,]\!] := \textbf{\textit{ts}} \text{ impacts } \textbf{\textit{a}} \text{ with undefined consequence.}$$

However, only impact relations originating in unwanted incidents may be annotated with consequences. This relation have the following semantics:

$$[\![\, ui \xrightarrow{c} a \,]\!] := \textbf{\textit{ui}} \text{ impacts } \textbf{\textit{a}} \text{ with } [\![\, c \,]\!].$$

5.3.6 Annotations

$$
\begin{aligned}
[\![v]\!] &:= \text{vulnerability } \boldsymbol{v} \\
[\![V_n]\!] &:= \text{vulnerabilities } \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \\
[\![l]\!] &:= \text{likelihood } \boldsymbol{l} \\
[\![c]\!] &:= \text{consequence } \boldsymbol{c}
\end{aligned}
$$

5.4 Example

We continue the example from the section on asset diagrams to illustrate the translation of threat diagrams. In the threat diagram in Figure 7, we have numbered the relations from 1 to 21 in the order in which they are translated.

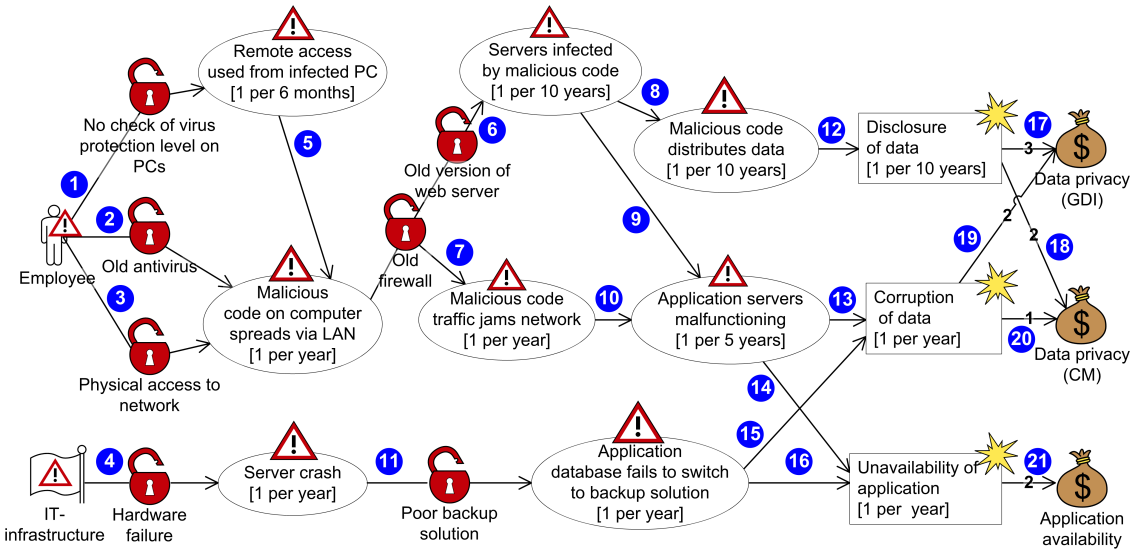


Figure 7: Threat diagram

We start by translating the vertices of the diagram, from the top left to the bottom right:

1. ***Employee*** is an accidental threat.
2. ***IT-infrastructure*** is a non-human threat.
3. Threat scenario ***Remote access used from infected PC*** occurs with likelihood ***1 per 6 months***.
4. Threat scenario ***Malicious code on computer spreads via LAN*** occurs with likelihood ***1 per year***.
5. Threat scenario ***Server crash*** occurs with likelihood ***1 per year***.
6. Threat scenario ***Servers infected by malicious code*** occurs with likelihood ***1 per 10 years***.
7. Threat scenario ***Malicious code traffic jams network*** occurs with likelihood ***1 per year***.
8. Threat scenario ***Malicious code distributes data*** occurs with likelihood ***1 per 10 years***.

9. Threat scenario ***Application servers malfunctioning*** occurs with likelihood ***1 per 5 years***.
10. Threat scenario ***Application database fails to switch to backup solution*** occurs with likelihood ***1 per year***.
11. Unwanted incident ***Disclosure of data*** occurs with likelihood ***1 per 10 years***.
12. Unwanted incident ***Corruption of data*** occurs with likelihood ***1 per year***.
13. Unwanted incident ***Unavailability of application*** occurs with likelihood ***1 per year***.
14. ***Data privacy (GDI)*** is an asset.
15. ***Data privacy (CM)*** is an asset.
16. ***Application availability*** is an asset.

Then we translate the relations of the diagram, in the order suggested by the numbering in Figure 7:

1. ***Employee*** exploits vulnerability ***No check of virus protection level on PCs*** to initiate ***Remote access used from infected PC*** with undefined likelihood.
2. ***Employee*** exploits vulnerability ***Old antivirus*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.
3. ***Employee*** exploits vulnerability ***Physical access to network*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.
4. ***IT-infrastructure*** exploits vulnerability ***Hardware failure*** to initiate ***Server crash*** with undefined likelihood.
5. ***Remote access used from infected PC*** leads to ***Malicious code on computer spreads via LAN*** with undefined likelihood.
6. ***Malicious code on computer spreads via LAN*** leads to ***Servers infected by malicious code*** with undefined likelihood, due to vulnerabilities ***Old firewall***, ***Old version of web server***.
7. ***Malicious code on computer spreads via LAN*** leads to ***Malicious code traffic jams network*** with undefined likelihood, due to vulnerability ***Old firewall***.
8. ***Servers infected by malicious code*** leads to ***Malicious code distributes data*** with undefined likelihood.
9. ***Servers infected by malicious code*** leads to ***Application servers malfunctioning*** with undefined likelihood.
10. ***Malicious code traffic jams network*** leads to ***Application servers malfunctioning*** with undefined likelihood.
11. ***Server crash*** leads to ***Application database fails to switch to backup solution*** with undefined likelihood, due to vulnerability ***Poor backup solution***.
12. ***Malicious code distributes data*** leads to ***Disclosure of data*** with undefined likelihood.
13. ***Application servers malfunctioning*** leads to ***Corruption of data*** with undefined likelihood.

14. ***Application servers malfunctioning*** leads to ***Unavailability of application*** with undefined likelihood.
15. ***Application database fails to switch to backup solution*** leads to ***Corruption of data*** with undefined likelihood.
16. ***Application database fails to switch to backup solution*** leads to ***Unavailability of application*** with undefined likelihood.
17. ***Disclosure of data*** impacts ***Data privacy (GDI)*** with consequence *3*.
18. ***Disclosure of data*** impacts ***Data privacy (CM)*** with consequence *2*.
19. ***Corruption of data*** impacts ***Data privacy (GDI)*** with consequence *2*.
20. ***Corruption of data*** impacts ***Data privacy (CM)*** with consequence *1*.
21. ***Unavailability of application*** impacts ***Application availability*** with consequence *2*.

As was the case with the asset diagrams, we may choose to structure the translation of the relations to reflect the structure of the diagrams, and to make the translation more readable:

1.-3. ***Employee*** exploits vulnerability

- ***No check of virus protection level on PCs*** to initiate ***Remote access used from infected PC*** with undefined likelihood.
- ***Old antivirus*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.
- ***Physical access to network*** to initiate ***Malicious code on computer spreads via LAN*** with undefined likelihood.

4. ***IT-infrastructure*** exploits vulnerability ***Hardware failure*** to initiate ***Server crash*** with undefined likelihood.
5. ***Remote access used from infected PC*** leads to ***Malicious code on computer spreads via LAN*** with undefined likelihood.

6.-7. ***Malicious code on computer spreads via LAN*** leads to

- ***Servers infected by malicious code*** with undefined likelihood, due to vulnerabilities ***Old firewall***, ***Old version of web server***.
- ***Malicious code traffic jams network*** with undefined likelihood, due to vulnerability ***Old firewall***.

8.-9. ***Servers infected by malicious code*** leads to

- ***Malicious code distributes data*** with undefined likelihood.
- ***Application servers malfunctioning*** with undefined likelihood.

10. ***Malicious code traffic jams network*** leads to ***Application servers malfunctioning*** with undefined likelihood.
11. ***Server crash*** leads to ***Application database fails to switch to backup solution*** with undefined likelihood, due to vulnerability ***Poor backup solution***.
12. ***Malicious code distributes data*** leads to ***Disclosure of data*** with undefined likelihood.

13.-14. ***Application servers malfunctioning*** leads to

- ***Corruption of data*** with undefined likelihood.
- ***Unavailability of application*** with undefined likelihood.

15.-16. ***Application database fails to switch to backup solution***

- ***Corruption of data*** with undefined likelihood.
- ***Unavailability of application*** with undefined likelihood.

17.-18. ***Disclosure of data*** impacts

- ***Data privacy (GDI)*** with consequence *3*.
- ***Data privacy (CM)*** with consequence *2*.

19.-20. ***Corruption of data*** impacts

- ***Data privacy (GDI)*** with consequence *2*.
- ***Data privacy (CM)*** with consequence *1*.

21. ***Unavailability of application*** impacts ***CM3.*** ***Application availability*** with consequence *2*.

As we see from the example, the optimal order of the sentences depends on how you want to structure them.

6 Risk Diagrams

A risk diagram provides a summary of a threat diagram, displaying the risks posed to the assets by the threats. It has five basic elements: deliberate, accidental, and non-human threats, risks, and assets (see Figure 8). A risk may be assigned a risk value, risk function, or a likelihood and consequence pair, indicating its severity.

Figure 8: Basic building blocks of risk diagrams

All of the basic building blocks of the risk diagram are referred to as *vertices*. A risk diagram has three kinds of relations: an *initiate* relation from a threat to a risk, a *leads to* relation between risks, and an *impact* relation from a risk to an asset. The graphical syntax is displayed in Figure 9.

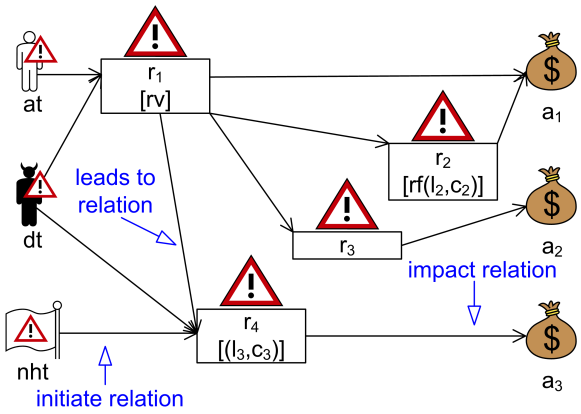


Figure 9: Graphical syntax of risk diagrams

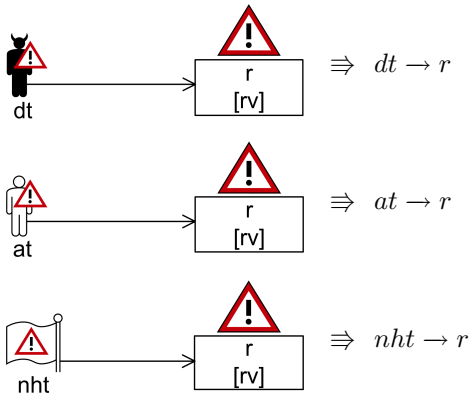
6.1 Textual Syntax of Risk Diagrams

$$
\begin{aligned}
\mathit{diagram} &= (\{\mathit{vertex}\}^{-}, \{\mathit{relation}\});\\
\mathit{vertex} &= \mathit{threat} \mid \mathit{risk} \mid \mathit{asset};\\
\mathit{relation} &= \mathit{initiate} \mid \mathit{leads\ to} \mid \mathit{impact};\\
\mathit{initiate} &= \mathit{threat} \rightarrow \mathit{risk};\\
\mathit{leads\ to} &= \mathit{risk} \rightarrow \mathit{risk};\\
\mathit{impact} &= \mathit{risk} \rightarrow \mathit{asset};\\
\mathit{threat} &= \mathit{deliberate\ threat} \mid \mathit{accidental\ threat} \mid \mathit{non\text{-}human\ threat};\\
\mathit{deliberate\ threat} &= \mathit{identifier};\\
\mathit{accidental\ threat} &= \mathit{identifier};\\
\mathit{non\text{-}human\ threat} &= \mathit{identifier};\\
\mathit{risk} &= \mathit{identifier}\,[(\mathit{risk\ level})]\,;\\
\mathit{asset} &= \mathit{identifier};\\
\mathit{risk\ level} &= \mathit{risk\ value} \mid \mathit{risk\ function}(\mathit{likelihood}, \mathit{consequence}) \mid\\
&\quad (\mathit{likelihood}, \mathit{consequence});\\
\mathit{risk\ value} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{risk\ function} &= \mathit{identifier};\\
\mathit{likelihood} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};\\
\mathit{consequence} &= \mathit{linguistic\ term} \mid \mathit{numerical\ value};
\end{aligned}
$$

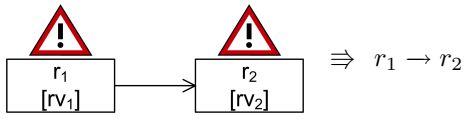
6.2 Translation from Graphical to Textual Syntax

See Section 4 for instructions on how to translate the vertices. To translate the relations from the graphical to the textual syntax we use the following rules:

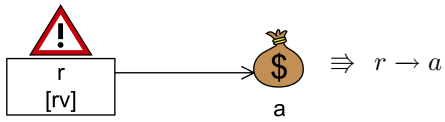
6.2.1 Initiate relation



6.2.2 Leads to relation



6.2.3 Impact relation



6.3 Translation from the Textual Syntax to English

6.3.1 Diagram

A diagram $D := (\{v_1, \ldots, v_n\}, \{r_1, \ldots, r_m\})$, $n > 0, m \geq 0$, is translated by translating each of its vertices and relations. Note that the logical conjunction implicit in the commas of the sets of vertices and relations translates into the period at the end of each English sentence representing a vertex or relation.

$$[\![D]\!] := [\![v_1]\!] \ldots [\![v_n]\!] [\![r_1]\!] \ldots [\![r_m]\!]$$

6.3.2 Vertices

$$\begin{aligned}
[\![dt]\!] &:= \textbf{\textit{dt}} \text{ is a deliberate threat.} \\
[\![at]\!] &:= \textbf{\textit{at}} \text{ is an accidental threat.} \\
[\![nht]\!] &:= \textbf{\textit{nht}} \text{ is a non-human threat.} \\
[\![a]\!] &:= \textbf{\textit{a}} \text{ is an asset.} \\
[\![r]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with undefined risk level.} \\
[\![r(rv)]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with } [\![rv]\!]. \\
[\![r(rf(l,c))]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with } [\![rf(l,c)]\!]. \\
[\![r(l,c)]\!] &:= \text{Risk } \textbf{\textit{r}} \text{ occurs with } [\![(l,c)]\!].
\end{aligned}$$

6.3.3 Initiate relation

As was the case for threat diagrams, the semantics of an initiate relation is independent of which kind of threat it originates from. In the following definition the threat t may be replaced by dt, at or nht, for deliberate, accidental and non-human threats respectively.

$$[\![t \rightarrow r]\!] := \boldsymbol{t} \text{ initiates } \boldsymbol{r}.$$

6.3.4 Leads to relation

$$[\![r_1 \rightarrow r_2]\!] := \boldsymbol{r}_1 \text{ leads to } \boldsymbol{r}_2.$$

6.3.5 Impact relation

$$[\![r \rightarrow a]\!] := \boldsymbol{r} \text{ impacts } \boldsymbol{a}.$$

6.3.6 Annotations

$$\begin{aligned} [\![rv]\!] &:= \text{ risk value } \boldsymbol{rv} \\ [\![rf(l,c)]\!] &:= \text{ risk function } \boldsymbol{rf} \text{ of } [\![(l,c)]\!] \\ [\![(l,c)]\!] &:= [\![l]\!] \text{ and } [\![c]\!] \\ [\![l]\!] &:= \text{ likelihood } \boldsymbol{l} \\ [\![c]\!] &:= \text{ consequence } \boldsymbol{c} \end{aligned}$$

6.4 Example

We continue the example from the last section to illustrate the translation of risk diagrams.

We start by translating the vertices of the risk diagram in Figure 10:

1. ***Employee*** is an accidental threat.
2. ***IT-infrastructure*** is a non-human threat.
3. Risk ***Disclosure of data 1*** occurs with risk value ***[medium]***.
4. Risk ***Disclosure of data 2*** occurs with risk value ***[low]***.
5. Risk ***Corruption of data 1*** occurs with risk value ***[medium]***.
6. Risk ***Corruption of data 2*** occurs with risk value ***[major]***.
7. Risk ***Unavailability of application*** occurs with risk value ***[medium]***.
8. ***Data privacy (GDI)*** is an asset.
9. ***Data privacy (CM)*** is an asset.
10. ***Application availability*** is an asset.

Translating the diagram relation by relation in the order suggested in Figure 10:

1. ***Employee*** initiates ***Disclosure of data 1***.
2. ***Employee*** initiates ***Disclosure of data 2***.

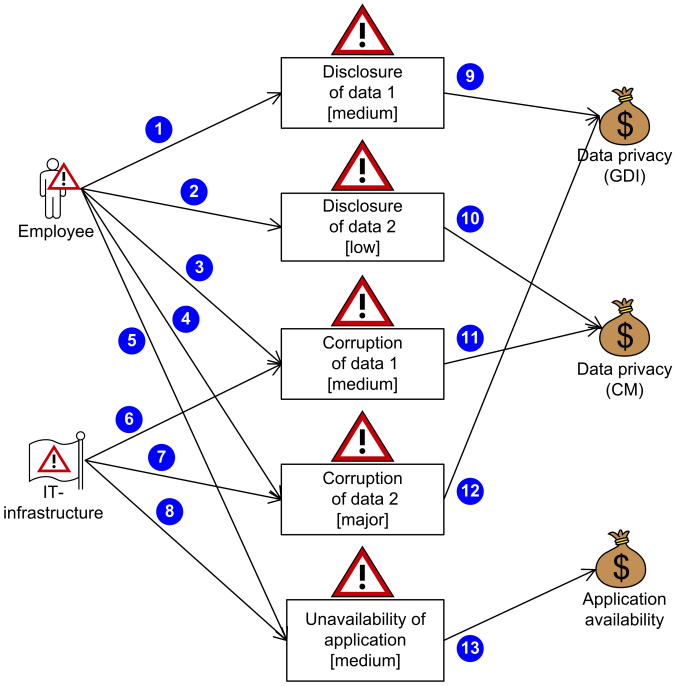


Figure 10: Risk diagram

3. ***Employee*** initiates ***Corruption of data 1***.
4. ***Employee*** initiates ***Corruption of data 2***.
5. ***Employee*** initiates ***Unavailability of application***.
6. ***IT-infrastructure*** initiates ***Corruption of data 1***.
7. ***IT-infrastructure*** initiates ***Corruption of data 2***.
8. ***IT-infrastructure*** initiates ***Unavailability of application***.
9. ***Disclosure of data 1*** impacts ***Data privacy (GDI)***.
10. ***Disclosure of data 2*** impacts ***Data privacy (CM)***.
11. ***Corruption of data 1*** impacts ***Data privacy (CM)***.
12. ***Corruption of data 2*** impacts ***Data privacy (GDI)***.
13. ***Unavailability of application*** impacts ***Application availability***.

Structuring the text as with the other diagrams:

1.-5. ***Employee*** initiates

- ***Disclosure of data 1***.
- ***Disclosure of data 2***.
- ***Corruption of data 1***.
- ***Corruption of data 2***.

- ***Unavailability of application***.

6.-8. ***IT-infrastructure*** initiates

- ***Corruption of data 1***.
- ***Corruption of data 2***.
- ***Unavailability of application***.

9. ***Disclosure of data 1*** impacts ***Data privacy (GDI)***.
10. ***Disclosure of data 2*** impacts ***Data privacy (CM)***.
11. ***Corruption of data 1*** impacts ***Data privacy (CM)***.
12. ***Corruption of data 2*** impacts ***Data privacy (GDI)***.
13. ***Unavailability of application*** impacts ***Application availability***.

7 Treatment Diagrams

A treatment diagram is a presentation of proposed treatments of elements of a threat diagram. It expands the threat diagram, inserting the risks from the risk diagram between the appropriate unwanted incidents and assets, and adds treatment scenarios. Its basic building blocks are shown in Figure 11.

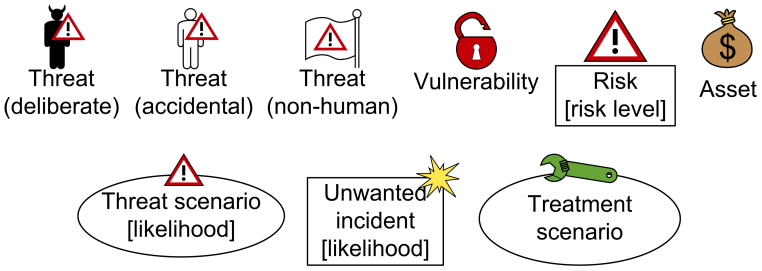


Figure 11: Basic building blocks of treatment diagrams

We refer to all the basic building blocks except vulnerabilities as *vertices*. The *initiate* and *leads to* relations are defined as for the threat diagram, adding the possibility of leads to relations from unwanted incidents to risks. This new relation is annotated with neither vulnerabilities nor likelihoods. As was the case for risk diagrams, the *impact* relation relates a risk to an asset.

Treatment Category	Abbreviation
avoid	*av*
decrease likelihood	*dl*
decrease consequence	*dc*
share	*sh*
retain	*re*

Table 2: Treatment category abbreviations

We also have an additional relation, *treat*, from a treatment scenario to any other kind of element. It may be annotated with a treatment category, cf. the Joint Australian/New Zealand Standard for Risk Management [3, p. 20]. The abbreviations in Table 2 are used to annotate the treat relations in the semantics for the treatment and treatment overview diagrams.

Figure 12 shows the graphical syntax of treatment diagrams.

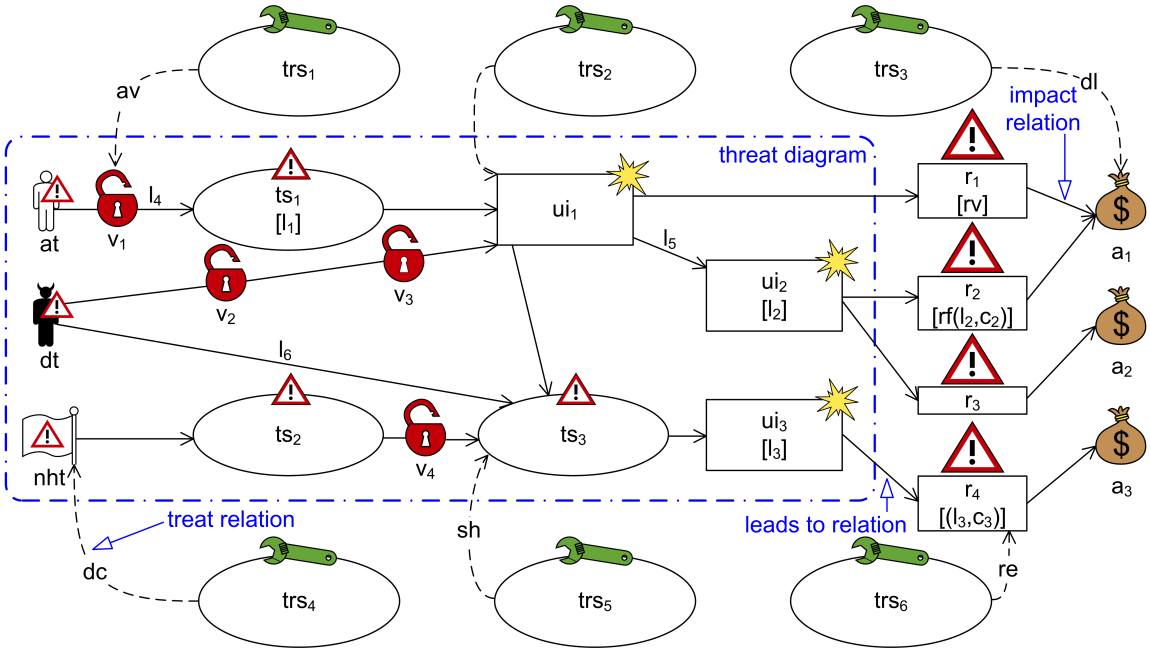


Figure 12: Graphical syntax of treatment diagrams

7.1 Textual Syntax of Treatment Diagrams

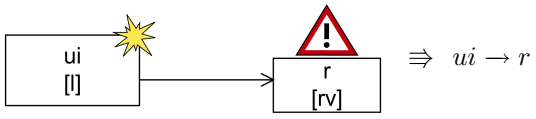
$$
\begin{aligned}
diagram &= (\{vertex\}^{-}, \{relation\});\\
vertex &= threat \mid threat\ scenario \mid unwanted\ incident \mid risk \mid asset \mid\\
&\quad\ treatment\ scenario;\\
relation &= initiate \mid leads\ to \mid impact \mid treat;\\
initiate &= threat \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\quad\ threat \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident \mid\\
&\quad\ threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\quad\ threat\ scenario \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident \mid\\
&\quad\ unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} threat\ scenario \mid\\
&\quad\ unwanted\ incident \xrightarrow{[vulnerability\ set][likelihood]} unwanted\ incident;\\
leads\ to &= unwanted\ incident \rightarrow risk;\\
impact &= risk \rightarrow asset;\\
treat &= treatment\ scenario \xrightarrow{[treatment\ category]} threat \mid\\
&\quad\ treatment\ scenario \xrightarrow{[treatment\ category]} vulnerability \mid\\
&\quad\ treatment\ scenario \xrightarrow{[treatment\ category]} threat\ scenario \mid\\
&\quad\ treatment\ scenario \xrightarrow{[treatment\ category]} unwanted\ incident \mid
\end{aligned}
$$

$$
\begin{aligned}
&\textit{treatment scenario} \xrightarrow{[\textit{treatment category}]} \textit{risk} \mid \\
&\textit{treatment scenario} \xrightarrow{[\textit{treatment category}]} \textit{asset};\\
\textit{threat} &= \textit{deliberate threat} \mid \textit{accidental threat} \mid \textit{non-human threat};\\
\textit{deliberate threat} &= \textit{identifier};\\
\textit{accidental threat} &= \textit{identifier};\\
\textit{non-human threat} &= \textit{identifier};\\
\textit{vulnerability set} &= \{\textit{vulnerability}\}^{-};\\
\textit{vulnerability} &= \textit{identifier};\\
\textit{threat scenario} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{unwanted incident} &= \textit{identifier}\,[(\textit{likelihood})];\\
\textit{risk} &= \textit{identifier}\,[(\textit{risk level})];\\
\textit{asset} &= \textit{identifier};\\
\textit{treatment scenario} &= \textit{identifier};\\
\textit{treatment category} &= \text{‘avoid’} \mid \text{‘decrease likelihood’} \mid \text{‘decrease consequence’} \mid \text{‘share’} \mid \text{‘retain’};\\
\textit{risk level} &= \textit{risk value} \mid \textit{risk function}(\textit{likelihood}, \textit{consequence}) \mid \\
&\quad (\textit{likelihood}, \textit{consequence});\\
\textit{risk value} &= \textit{linguistic term} \mid \textit{numerical value};\\
\textit{risk function} &= \textit{identifier};\\
\textit{likelihood} &= \textit{linguistic term} \mid \textit{numerical value};\\
\textit{consequence} &= \textit{linguistic term} \mid \textit{numerical value};
\end{aligned}
$$

7.2 Translation from Graphical to Textual Syntax

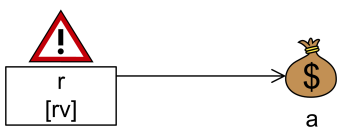
The part of the treatment diagram in Figure 12 within the blue threat diagram-box are translated in the section on threat diagrams. What remains to be translated are the leads to relation from an unwanted incident to a risk, the impact relation, and the treat relations. As mentioned in earlier sections, we include all optional elements in the translation rules.

7.2.1 Leads to relation



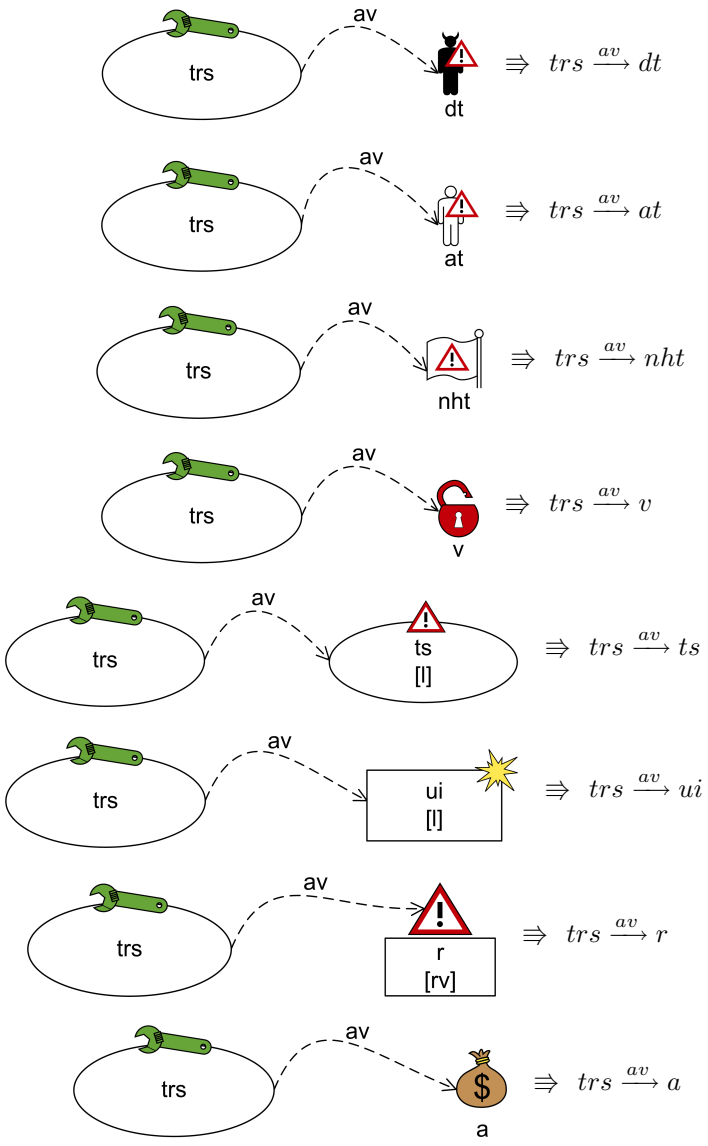
$\Rrightarrow\ ui \rightarrow r$

7.2.2 Impact relation



$\Rrightarrow\ r \rightarrow a$

7.2.3 Treat relation



To obtain the translation rule for a treat relation without treatment category, remove the *av*-annotation from the arrow in the textual expression. If the treat relation is annotated with a different treatment category, simply replace *av* with the appropriate abbreviation.

7.3 Translation from the Textual Syntax to English

The translation of the vertices and relations within the blue threat diagram-box in Figure 12 is given in Section 5.3. The translations of risks and assets, and of the impact relation, are given in Section 6.3. This leaves treatment scenarios, treat relations and the new leads to relation.

7.3.1 Vertices

$$[\![\, trs \,]\!] := \textbf{\textit{trs}} \text{ is a treatment scenario.}$$

7.3.2 Leads to relation

$$[\![\, ui \rightarrow r \,]\!] := \textbf{\textit{ui}} \text{ leads to } \textbf{\textit{r}}.$$

7.3.3 Treat relation

An unannotated treat relation has the same translation, independent of the object it treats:

$$[\![\, trs \rightarrow dt \,]\!] := \textbf{\textit{trs}} \text{ treats } \textbf{\textit{dt}}.$$

Here the dt for deliberate threat may be replaced by at, nht, v, ts, ui, r or a.

When the treat relation is annotated with a treatment category, the translation depends on the object being treated. In the following, the annotation av may be replaced by any of the other annotations. t for threat may be replaced by dt, at or nht.

$$\begin{aligned}
[\![\, trs \xrightarrow{av} t \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{t}} \text{ attacking the system.} \\
[\![\, trs \xrightarrow{av} v \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{v}} \text{ being exploited.} \\
[\![\, trs \xrightarrow{av} ts \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{ts}} \text{ being initiated.} \\
[\![\, trs \xrightarrow{av} ui \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{ui}} \text{ being initiated.} \\
[\![\, trs \xrightarrow{av} r \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{r}}. \\
[\![\, trs \xrightarrow{av} a \,]\!] &:= \textbf{\textit{trs}}\ [\![\, av \,]\!] \text{ of } \textbf{\textit{a}} \text{ being harmed.}
\end{aligned}$$

7.3.4 Annotations

The annotations left to translate are the treatment categories:

$$\begin{aligned}
[\![\, av \,]\!] &:= \text{avoids the risk} \\
[\![\, dl \,]\!] &:= \text{decreases the likelihood} \\
[\![\, dc \,]\!] &:= \text{decreases the consequences} \\
[\![\, sh \,]\!] &:= \text{shares the risk} \\
[\![\, re \,]\!] &:= \text{retains the risk}
\end{aligned}$$

7.4 Example

We continue the example from the last section to illustrate the translation of treatment diagrams:

Most of the diagram has already been translated in Sections 5.4 and 6.4. The only remaining vertices are the treatment scenarios:

1. ***Increase awareness of security risks*** is a treatment scenario.
2. ***Upgrade server*** is a treatment scenario.
3. ***Limit access to network*** is a treatment scenario.

We translate the remaining relations in the order suggested in Figure 13:

1. ***Increase awareness of security risks*** treats ***No check of virus protection level on PCs***.
2. ***Upgrade server*** treats ***Old version of webserver***.

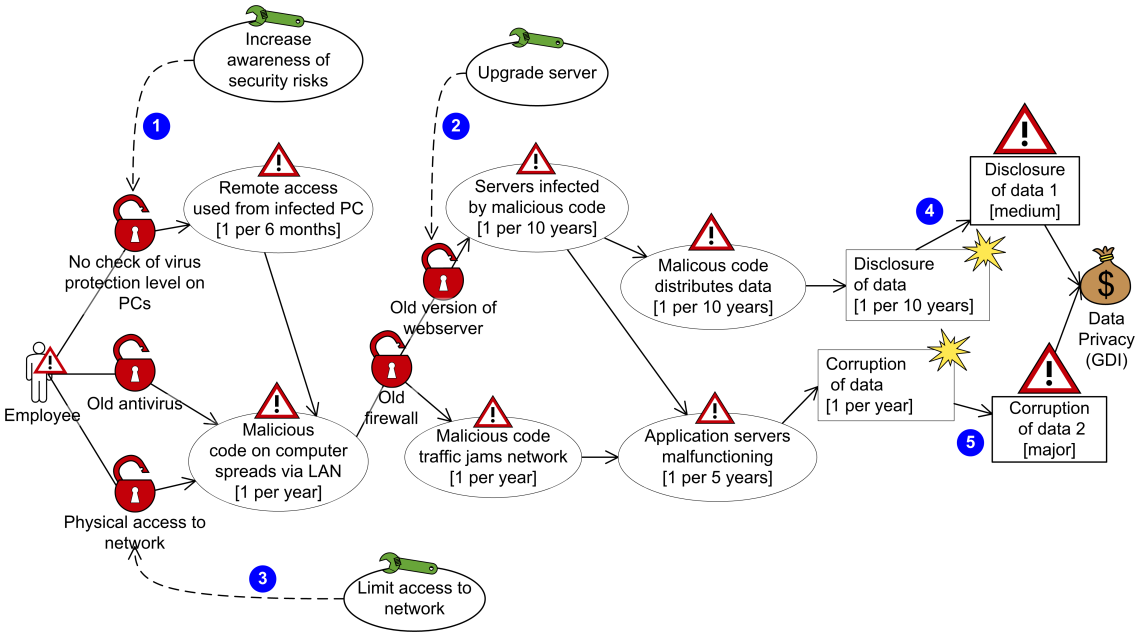


Figure 13: Treatment diagram

3. ***Limit access to network*** treats ***Physical access to network***.

4. ***Disclosure of data*** leads to ***Disclosure of data 1***.

5. ***Corruption of data*** leads to ***Corruption of data 2***.

8 Treatment Overview Diagrams

The last kind of CORAS diagram is the treatment overview diagram. Its basis is a risk diagram, to which the treatment scenarios of a corresponding treatment diagram are added. As in the treatment diagram, *treat* relations are used to indicate the elements affected by each treatment scenario. They may be annotated with a treatment category, cf. the Joint Australian/New Zealand Standard for Risk Management [3, p. 20]. See Table 2, p. 23, for the treatment category abbreviations. Figure 14 displays the basic component of treatment overview diagrams, and their graphical syntax is shown in Figure 15. We refer to all of the basic building blocks as *vertices*.

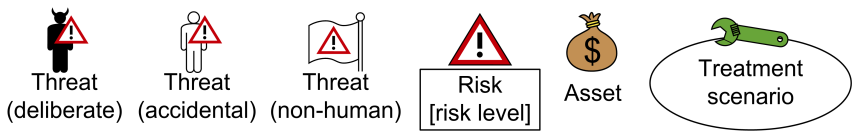


Figure 14: Basic building blocks of treatment overview diagrams

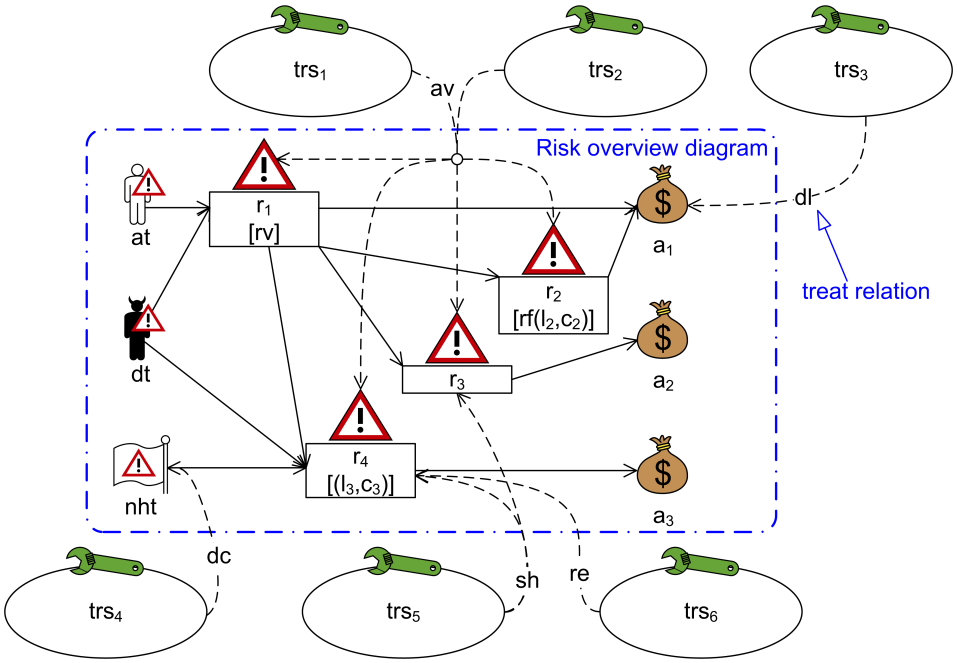


Figure 15: Graphical syntax of treatment overview diagrams

8.1 Textual Syntax of Treatment Overview Diagrams

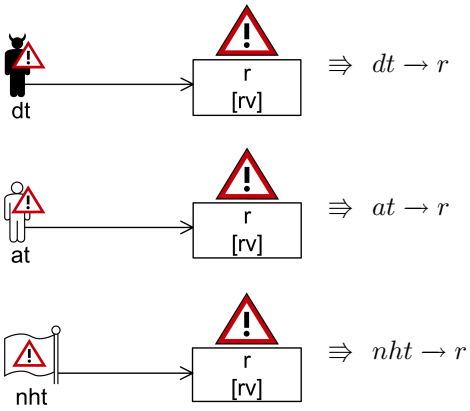
$$
\begin{aligned}
\mathit{diagram} &= (\{\mathit{vertex}\}^{-}, \{\mathit{relation}\});\\
\mathit{vertex} &= \mathit{threat} \mid \mathit{risk} \mid \mathit{asset} \mid \mathit{treatment\ scenario};\\
\mathit{relation} &= \mathit{initiate} \mid \mathit{leads\ to} \mid \mathit{impact} \mid \mathit{treat};\\
\mathit{initiate} &= \mathit{threat} \rightarrow \mathit{risk};\\
\mathit{leads\ to} &= \mathit{risk} \rightarrow \mathit{risk};\\
\mathit{impact} &= \mathit{risk} \rightarrow \mathit{asset};\\
\mathit{treat} &= \mathit{treatment\ scenario} \xrightarrow{[\mathit{treatment\ category}]} \mathit{threat} \mid\\
&\qquad \mathit{treatment\ scenario} \xrightarrow{[\mathit{treatment\ category}]} \mathit{risk} \mid\\
&\qquad \mathit{treatment\ scenario} \xrightarrow{[\mathit{treatment\ category}]} \mathit{asset};\\
\mathit{threat} &= \mathit{deliberate\ threat} \mid \mathit{accidental\ threat} \mid \mathit{non\text{-}human\ threat};\\
\mathit{deliberate\ threat} &= \mathit{identifier};\\
\mathit{accidental\ threat} &= \mathit{identifier};\\
\mathit{non\text{-}human\ threat} &= \mathit{identifier};\\
\mathit{risk} &= \mathit{identifier}\ [(\mathit{risk\ level})]\ ;\\
\mathit{asset} &= \mathit{identifier};\\
\mathit{treatment\ scenario} &= \mathit{identifier};\\
\mathit{treatment\ category} &= \text{‘avoid’} \mid \text{‘decrease likelihood’} \mid \text{‘decrease consequence’} \mid \text{‘share’} \mid \text{‘retain’};\\
\mathit{risk\ level} &= \mathit{risk\ value} \mid \mathit{risk\ function}(\mathit{likelihood}, \mathit{consequence}) \mid\\
&\qquad (\mathit{likelihood}, \mathit{consequence});
\end{aligned}
$$

$$\begin{aligned} \textit{risk value} &= \textit{linguistic term} \mid \textit{numerical value};\\ \textit{risk function} &= \textit{identifier};\\ \textit{likelihood} &= \textit{linguistic term} \mid \textit{numerical value};\\ \textit{consequence} &= \textit{linguistic term} \mid \textit{numerical value}; \end{aligned}$$

8.2 Translation from Graphical to Textual Syntax

Most of the treatment overview diagram was translated in Section 6.2, while treatment scenarios and the treat relation was translated in Section 7.2. The following are the relevant translation rules:

8.2.1 Initiate relation

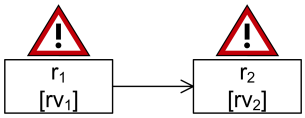


$\Rrightarrow dt \rightarrow r$

$\Rrightarrow at \rightarrow r$

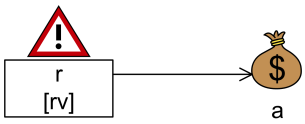
$\Rrightarrow nht \rightarrow r$

8.2.2 Leads to relation



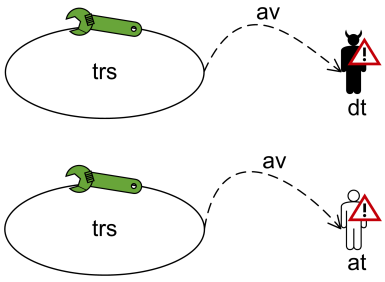
$\Rrightarrow r_1 \rightarrow r_2$

8.2.3 Impact relation



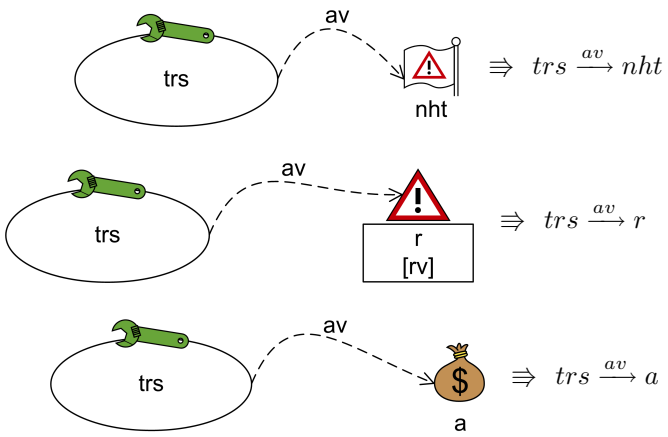
$\Rrightarrow r \rightarrow a$

8.2.4 Treat relation



$\Rrightarrow trs \xrightarrow{av} dt$

$\Rrightarrow trs \xrightarrow{av} at$



8.3 Translation from the Textual Syntax to English

The basic components, the treatment categories and the treat relations have the same structured semantics as in the treatment diagram. The rest of the semantics can be found in Section 6 on risk diagrams.

8.4 Example

We continue the example from the last section to illustrate the translation of treatment overview diagrams:

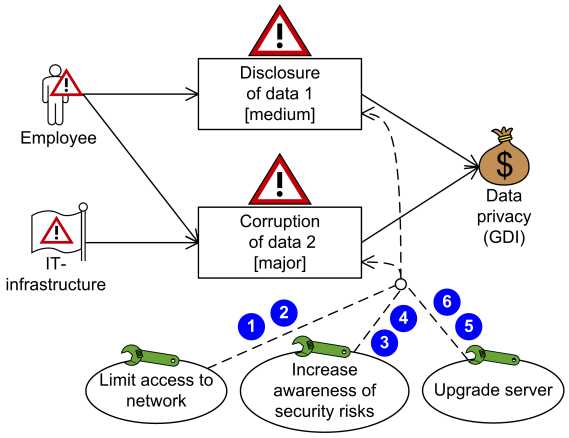


Figure 16: Treatment overview diagram

Everything but the treat relations have already been translated:

1. ***Limit access to network*** treats ***Disclosure of data 1***.
2. ***Limit access to network*** treats ***Corruption of data 2***.
3. ***Increase awareness of security risks*** treats ***Disclosure of data 1***.

4. ***Increase awareness of security risks*** treats ***Corruption of data 2***.
5. ***Upgrade server*** treats ***Disclosure of data 1***.
6. ***Upgrade server*** treats ***Corruption of data 2***.

Organising the results:

1.-2. ***Limit access to network*** treats

- ***Disclosure of data 1***.
- ***Corruption of data 2***.

3.-4. ***Increase awareness of security risks*** treats

- ***Disclosure of data 1***.
- ***Corruption of data 2***.

5.-6. ***Upgrade server*** treats

- ***Disclosure of data 1***.
- ***Corruption of data 2***.

9 Conclusion

The CORAS language has been designed to be easily understandable in order to aid communication in a security risk analysis context. Even so, situations in which there is a need to explain the intended meaning of a construct or expression are bound to arise. An example of such a situation is when the analysis results are distributed to parties, within the client company, which have not been part of the analysis process.

In order to fill this need, this paper has presented a structured semantics for the CORAS security risk modelling language. We have provided instructions on how to translate the two main CORAS diagrams, via the textual syntax, into a paragraph of English.

The report satisfies the success criteria stated at the end of Section 1:

1. *The translation from CORAS diagrams to English should be modular.* We divided the translation into two independent steps: (A) Graphical to textual syntax, and (B) Textual syntax to English. Both of these component translations are modular (the diagram and textual expressions are translated relation by relation) so the complete translation is modular.

2. *The resulting paragraph should be understandable English.* The wording of the English phrases in the structured semantics is based on the descriptions used by CORAS developers to explain the diagrams to non-specialists during a CORAS security risk analysis. This gives us a translation into phrases of clear, understandable English.

3. *The translation should be easy to perform.* The translation of a diagram is done by pattern matching, first by matching each relation to a translation rule and removing unwanted optional elements, then by matching the resulting textual expression to a rule in the structured semantics.

4. *The translation should be possible to automate.* The translation rules and the structured semantics are presented in such a way that the pattern matching may be done automatically. However, the structuring of the translation depends to a large degree on the structure of the original diagram. Thus it is difficult to give a general recommendation on how this is done.

This means that while it is possible to automatically structure the translation to reflect the branching nature of the CORAS diagrams, a more comprehensive structuring may require human intervention unless the structure of the diagram adheres to a predefined style.

5. *It should be possible to translate inconsistent diagrams, and the translation should enable the user to identify inconsistencies.* As a CORAS diagram is translated relation by relation and not from a more global perspective, it does not matter to the translation whether or not the diagram is inconsistent. However, the inconsistencies may not be conspicuous before the translation is appropriately structured.

Future Work

The work presented in this paper is the starting point for several research activities. The most immediate would be empirical testing of the translation process and the resulting sentences. The CORAS tool (see `http://coras.sourceforge.net`) will be updated to reflect the structure of the textual syntax and facilitate automatic translation.

The development of the CORAS method and language continues in several projects at SINTEF ICT, building on experiences from industrial case studies. There is also ongoing work aiming for an integrated approach to security and usability analysis.

Related Work

Misuse cases [2, 18, 19] was an important source of inspiration in the development of the UML profile mentioned in Sec. 1. A misuse case is a kind of UML use case [11] which characterizes functionality that the system should not allow. There are a number of security oriented extensions of UML, e.g. UMLSec [12] and SecureUML [13]. These and related approaches, however, have all been designed to capture security properties and security aspects at a more detailed level than our language. Moreover, their focus is not on brainstorming sessions as in our case.

Fault tree is a tree-notation used in fault tree analysis (FTA) [9]. The top node represents an unwanted incident, or failure, and the different events that may lead to the top event are modelled as branches of nodes, with the leaf node as the causing event. Our threat diagrams often resemble fault trees, but may have more than one top node.

Event tree analysis (ETA) [8] focuses on illustrating the consequences of an event and the probabilities of these. Event trees can to a large extent also be simulated in our notation.

Attack trees [17] aim to provide a formal and methodical way of describing the security of a system based on the attacks it may be exposed to. The notation uses a tree structure similar to fault trees, with the attack goal as the top node and different ways of achieving the goal as leaf nodes. Our approach supports this way of modelling, but additionally facilitates the specification of the attack initiators (threats) and the harm caused by the attack (damage to assets).

Acknowledgements

The research for this paper has been funded by the projects SECURIS (152839/220) and DIGIT (180052/S10) funded by the Research Council of Norway, and the EU-project S3MS (IST-2006-027004). The authors thank Gyrd Brændeland, Iselin Engan, Mass Soldal Lund, Atle Refsdal and Bjørnar Solhaug for valuable input.

References

[1] Jan Øyvind Aagedal, Folker den Braber, Theo Dimitrakos, Bjørn Axel Gran, Dimitris Raptis, and Ketil Stølen. Model-based risk assessment to improve enterprise security. In *EDOC'02*, pages 51–64. IEEE Computer Society, 2002.

[2] Ian F. Alexander. Misuse cases: Use cases with hostile intent. *IEEE Software*, 20(1):58–66, 2003.

[3] AS/NZS 4360:2004. *Australian/New Zealand Standard for Risk Management*, 2004.

[4] Folker den Braber, Ida Hogganvik, Mass Soldal Lund, Ketil Stølen, and Fredrik Vraalsen. Model-based security analysis in seven steps – a guided tour to the CORAS method. *BT Technology Journal*, 25(1):101–117, 2007.

[5] Ida Hogganvik and Ketil Stølen. On the comprehension of security risk scenarios. In *Proc.of 13th Int. Workshop on Program Comprehension*, pages 115–124. IEEE Computer Society, 2005.

[6] Ida Hogganvik and Ketil Stølen. Risk Analysis Terminology for IT systems: Does it match Intuition? In *Proc. of Int. Symposium on Empirical Software Engineering*, pages 13–23. IEEE Computer Society, 2005.

[7] Ida Hogganvik and Ketil Stølen. A Graphical Approach to Risk Identification, Motivated by Empirical Investigations. In *Proc. of 9th Int. Conf. on Model Driven Engineering Languages and Systems*, volume 4199 of *LNCS*, pages 574–588. Springer, 2006.

[8] IEC60300. *Event Tree Analysis in Dependability management – Part 3: Application guide – Section 9: Risk analysis of technological systems.* 1995.

[9] IEC61025. *Fault Tree Analysis (FTA).* 1990.

[10] ISO/IEC 14977:1996(E). *Information technology — Syntactic metalanguage — Extended BNF*, first edition, 1996.

[11] Ivar Jacobson, Magnus Christenson, Patrik Jonsson, and Gunnar Övergaard. *Object-Oriented Software Engineering. A Use Case Driven Approach.* Addison-Wesley, 1992.

[12] Jan Jürjens. *Secure Systems Development with UML.* Springer, 2005.

[13] Torsten Lodderstedt, David A. Basin, and Jürgen Doser. SecureUML: A UML-based modeling language for model-driven security. In *UML'02*, volume 2460 of *LNCS*, pages 426–441. Springer, 2002.

[14] Mass Soldal Lund, Ida Hogganvik, Seehusen Fredrik, and Ketil Stølen. UML profile for security assessment. Technical Report STF40 A03066, SINTEF ICT, 2003.

[15] OMG. *Unified Modeling Language Specification, version 2.0*, 2004.

[16] OMG. *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms*, 2005.

[17] Bruce Schneier. Attack trees: Modeling security threats. *Dr. Dobb's Journal of Software Tools*, 24(12):21–29, December 1999.

[18] Guttorm Sindre and Andreas L. Opdahl. Eliciting security requirements with misuse cases. In *TOOLS-PACIFIC'00*, pages 120–131, 2000.

[19] Guttorm Sindre and Andreas L. Opdahl. Templates for misuse case description. In *REFSQ'01*, pages 125–136, 2001.