SINTEF

SINTEF ICT

Address: NO-7465 Trondheim, NORWAY
Location: Forskningsveien 1
Telephone: +47 22 06 73 00
Fax: +47 22 06 73 50

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

A modular approach to the modelling and analysis of risk scenarios with mutual dependencies

AUTHOR(S)

Gyrd Brændeland, Heidi E. I. Dahl, Ketil Stølen

CLIENT(S)

Norges forskningsråd/The Research Council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
A8360	Unrestricted	189952/S10	
CLASS. THIS PAGE	**ISBN**	**PROJECT NO.**	**NO. OF PAGES/APPENDICES**
Unrestricted	978-82-14-04404-1	DIGIT 90B245	27/0
ELECTRONIC FILE CODE		**PROJECT MANAGER (NAME, SIGN.)**	**CHECKED BY (NAME, SIGN.)**
Modular_risk.doc		Ketil Stølen	Mass Soldal Lund
FILE CODE	**DATE**	**APPROVED BY (NAME, POSITION, SIGN.)**	
	2008-11-18	Bjørn Skjellaug, Forskningssjef	

ABSTRACT

This report describes a modular approach to the modelling and analysis of risk scenarios with mutual dependencies. The presented approach may be used to deduce the risk-level of an overall system from previous risk analyses of its constituent systems. It may also be used to decompose the analysis of a complex system into separate parts that can be carried out independently.

A custom made assumption-guarantee style is put forward as a means to describe risk scenarios with external dependencies. The assumption-guarantee style is built on top of the CORAS risk modelling language. The report also presents a set of deduction rules to facilitate various kinds of reasoning, including the analysis of mutual dependencies between risk scenarios expressed in the assumption-guarantee style.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	Risk analysis	Risikoanalyse
GROUP 2	Mutual dependency	Gjensidig avhengighet
SELECTED BY AUTHOR	Risk scenario	Risikoscenario
	Critical infrastructure	Kritisk infrastruktur
	Threat modelling	Trusselmodellering

A modular approach to the modelling and analysis of risk scenarios with mutual dependencies

Gyrd Brændeland[*,a,b], Heidi E. I. Dahl[a], Ketil Stølen[a,b]

[a]*SINTEF ICT, Oslo, Norway*
[b]*Department of Informatics, University of Oslo, Oslo, Norway*

Abstract

The risk analysis of critical infrastructures such as the electric power supply or telecommunications is complicated by the fact that such infrastructures are mutually dependent. We propose a modular approach to the modelling and analysis of risk scenarios with mutual dependencies. Our approach may be used to deduce the risk-level of an overall system from previous risk analyses of its constituent systems. It may also be used to decompose the analysis of a complex system into separate parts that can be carried out independently. A custom made assumption-guarantee style is put forward as a means to describe risk scenarios with external dependencies. We also define a set of deduction rules facilitating various kinds of reasoning, including the analysis of mutual dependencies between risk scenarios expressed in the assumption-guarantee style. The assumption-guarantee style is built on top of the CORAS risk modelling language.

Key words: Risk analysis, risk scenario, mutual dependency, critical infrastructure, threat modelling

1. Introduction

Mutual dependencies in the power supply have been apparent in blackouts in Europe and North America during the early two thousands, such as the blackout in Italy in September 2003 that affected most of the Italian population [37] and in North America the same year that affected several other infrastructures such as water supply, transportation and communication [28]. These and similar incidents have lead to increased focus on the protection of critical infrastructures. The Integrated Risk Reduction of Information-based Infrastructure Systems (IRRIIS) project [11], identified lack of appropriate risk analysis models as one of the key challenges in protecting critical infrastructures. There is a clear need for improved understanding of the impact of mutual dependencies on the overall risk level of critical infrastructures. When systems are mutually dependent, a threat towards one of them may realise threats towards all the others [33, 32]. One example, from the Nordic power sector, is the situation with reduced hydro power capacity

*Corresponding author.
Email address: gyrd.brendeland@sintef.no (Gyrd Brændeland)

in southern Norway and full hydro power capacity in Sweden [8]. In this situation the export to Norway from Sweden is high, which is a potential threat towards the Swedish power production causing instability in the network. If the network is already unstable, minor faults in the Swedish north/south corridor can lead to cascading outages collapsing the network in both southern Sweden and Norway. Hence, the threat originating in southern Norway contributes to an incident in southern Sweden, which again leads to an incident in Norway. Due to the potential for cascading effects of incidents affecting critical infrastructures, Rinaldi et al. [33] argue that mutually dependent infrastructures must be considered in a holistic manner. Within risk analysis, however, it is often not feasible to analyse all possible systems that affect the target of analysis at once; hence, we need a modular approach. Assumption-guarantee reasoning has been suggested as a means to facilitate modular system development [20, 27, 2]. In this paper we try to use the same idea to achieve modularity in risk analysis.

We present an assumption-guarantee style for the specification of risk scenarios with respect to context assumptions. Our approach is built on top of the CORAS risk modelling language [7]. The CORAS language has a formal syntax and a structured semantics. Moreover, the applicability of the language has been thoroughly evaluated in a series of industrial case studies, and by empirical investigations documented by Hogganvik and Stølen [14, 15, 16].

We also present a set of deduction rules for risk scenarios written in the assumption-guarantee style. The rules characterise conditions under which

- the analysis of complex scenarios can be decomposed into separate parts that can be carried out independently;
- the dependencies between scenarios can be resolved distinguishing bad dependencies (i.e., circular dependencies) from good dependencies (i.e., non-circular dependencies);
- risk models capturing the results of analysing parts can be put together to provide a risk model for the whole.

1.1. Contribution

The approach presented in this paper consolidates and extends previous work by Brændeland et al. [4]. The contributions of this paper are:

- Consolidated syntax and semantics for the assumption-guarantee risk paradigm.
- Consolidated and extended set of deduction rules to reason about mutual dependencies and composition of risk scenarios.
- Set of deduction rules for combining risk scenarios into overall risk scenarios.
- Exemplification of the proposed approach on risk scenarios from a realistic case-study of mutually dependent critical infrastructures.

1.2. *Analysing risk scenarios with mutual dependencies*

In order to demonstrate the applicability of our approach, we present a case-study involving the power systems in the southern parts of Sweden and Norway. Due to the strong mutual dependency between these systems, the effects of threats to either system can be quite complex. We focus on the analysis of blackout scenarios. The scenarios are inspired by the SINTEF study *Vulnerability of the Nordic Power System* [8]. However, the presented results with regard to probability and consequences of events are fictitious.

1.3. *Structure of the paper*

In Section 2 we introduce the graphical CORAS language in an example-driven manner focusing on the power supply in southern Sweden. In Section 3 we present a corresponding textual syntax for the graphical language. In Section 4 we define the structured semantics that is used to extract the meaning of a CORAS diagram. In Section 5 we provide a set of deduction rules to reason about CORAS diagrams. We refer to this as the CORAS calculus. In Section 6 we introduce dependent CORAS which is basically the CORAS language extended with an assumption-guarantee paradigm. We illustrate how dependent CORAS diagrams may be used to model mutual dependencies between the power systems in the southern parts of Sweden and Norway. Furthermore, we extend the CORAS calculus to facilitate reasoning about mutual dependency and demonstrate its usefulness on the already developed models. In Section 7 we extend the CORAS calculus with rules for compacting diagrams and show how they can be used to improve the presentational aspects of a CORAS diagram. In Section 8 we discuss related work. Finally, in Section 9 we present our conclusions and outline plans for future research.

2. Example-driven introduction of the CORAS language

The CORAS language has been designed to document, facilitate analysis, and communicate risk relevant information throughout the various phases of a risk analysis process. The language is graphical and distinguishes between five kinds of diagrams that are applied during the seven steps of a CORAS risk analysis: (1) introduction, (2) high level analysis, (3) approval, (4) risk identification (5) risk estimation (6) risk evaluation and (7) treatment. See den Braber et al. [7] for details on each step. To facilitate communication between participants of diverse backgrounds, the language employs simple icons and relations that are easy to read. In particular, the CORAS language is meant to be used during brainstorming sessions where discussions are documented along the way. The CORAS language was developed with particular focus on security risk analysis.

In this paper, we focus on CORAS threat diagrams (referred to as CORAS diagrams in the following), which are used during the risk identification and estimation phases of risk analysis. However, the presented approach to capture and analyse dependency carries over to the full CORAS language. Syntax and semantics of the full language is available as a technical report [6].

2.1. *Modelling threats towards the Swedish power system*

In this section we illustrate how threat diagrams are used during the risk identification and estimation phases, through an example involving threats to the power system in southern Sweden. In Section 6 this basic approach to threat modelling is generalised to capture external dependencies.

Threat diagrams describe how different threats exploit vulnerabilities to initiate threat scenarios and unwanted incidents, and which assets the unwanted incidents affect. The basic building blocks of threat diagrams are: threats (deliberate, accidental and non-human), vulnerabilities, threat scenarios, unwanted incidents and assets. Figure 1 presents the icons representing the basic building blocks. We often refer to these building blocks (with the exception of vulnerability) as vertices.

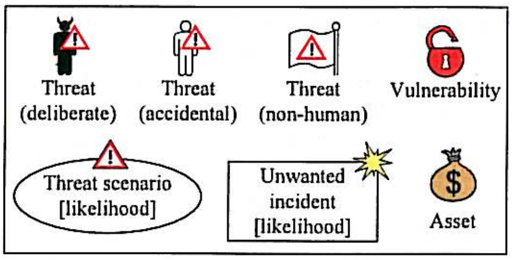

Figure 1: Basic building blocks of CORAS threat diagram

Figure 2 shows a threat diagram documenting possible threat scenarios leading to the unwanted incidents 'Blackout in southern Sweden' and 'Minor area blackout'. The target

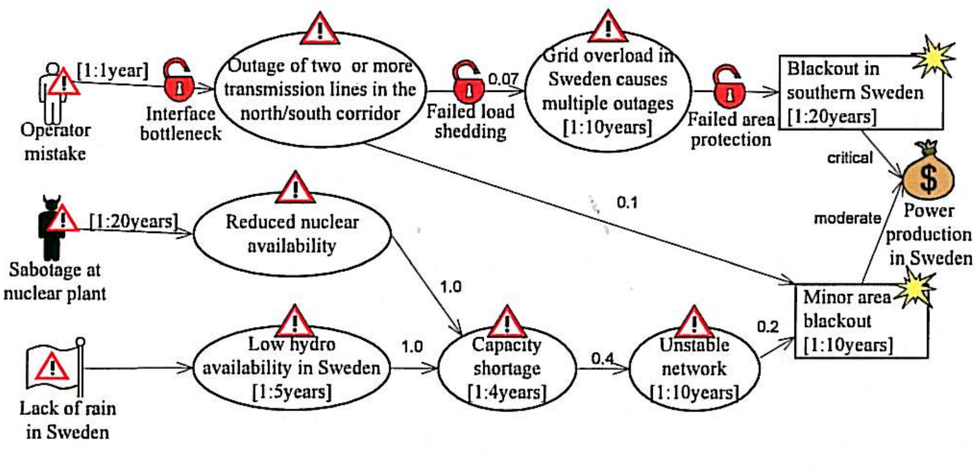

Figure 2: Threat scenarios leading to blackout in southern Sweden

of analysis in the example is limited to the power system in southern Sweden. We restrict ourselves to the potential risk of blackouts. A blackout is an unplanned and uncontrolled outage of a major part of the power system, leaving a large number of consumers without electricity [8].

When drawing a threat diagram, we start by placing the assets to the far right, and potential threats to the far left. The identified asset in the example is 'Power production in Sweden'. The construction of the diagram is an iterative process. We may add more threats later in the analysis. When the threat diagrams are drawn, the assets of relevance have already been identified and documented in an asset diagram, which for simplicity is left out here.

Next we place unwanted incidents to the left of the assets. In this case we have two unwanted incidents: 'Blackout in southern Sweden' and 'Minor area blackout'. The unwanted incidents represent events which have a negative impact on one or more of the identified assets. This impact relation is represented by drawing an arrow from the unwanted incident to the relevant asset.

The next step consists in determining the different ways a threat may initiate an unwanted incident. We do this by placing threat scenarios, each describing a series of events, between the threats and unwanted incidents and connecting them all with initiate relations and leads-to relations. An initiate relation originates in a threat and terminates in a threat scenario or an unwanted incident. A leads-to relation originates in a threat scenario or an unwanted incident and terminates in a threat scenario or an unwanted incident.

According to Doorman et al. [8] the most severe blackout scenarios affecting southern parts of Sweden are related to the main corridor of power transfer from mid Sweden to south Sweden. This is described by the threat scenario 'Outage of two or more transmission lines in the north/south corridor'.

In the example we have identified three threats: the accidental human threat 'Operator mistake', the deliberate human threat 'Sabotage at nuclear plant' and the non-human threat 'Lack of rain in Sweden'. In the case where a vulnerability is exploited when passing from one vertex to another, the vulnerability is positioned on the arrow representing the relation between them. For example the accidental human threat 'Operator mistake' exploits the vulnerability 'Interface bottleneck' to initiate the threat scenario 'Outage of two or more transmission lines in the north/south corridor'. This vulnerability refers to the fact that the corridor is a critical interconnection to the southern part of Sweden.

The threat diagram shows that the threat scenario 'Outage of two or more transmission lines in the north/south corridor' at best will lead only to the moderate unwanted incident 'Minor area blackout'. However, in combination with an already loaded transmission corridor, this threat scenario can exploit the vulnerability 'Failed load shedding' and cause the threat scenario 'Grid overload in Sweden causes multiple outages'. The vulnerability 'Failed load shedding' refers to the possible lack of sufficient countermeasures. The threat scenario 'Grid overload in Sweden causes multiple outages' can exploit the vulnerability 'Failed area protection' and cause the incident 'Blackout in southern Sweden'. Another scenario that can lead to 'Minor area blackout' is 'Unstable network' due to the threat scenario 'Capacity shortage'.

2.2. Annotating the diagram with likelihood and consequence values

In the risk estimation phase the CORAS diagrams are annotated with likelihoods (e.g. frequencies or probabilities) and consequence values. In most cases estimates of these values depend on input in the form of historical data or expert judgements. In many practical situations, it is difficult to find exact values. In such cases it can be useful to operate with intervals. Although the CORAS calculus as presented in Section 5 do not take intervals into consideration, it may still be used to reason about intervals since an interval may be understood as a set of exact values. In the same way as we use the calculus to check the consistency of different estimates specified as exact values we may also use the calculus to check the consistency of different estimates specified as intervals.

Both threat scenarios, unwanted incidents, initiate relations and leads-to relations may be annotated with likelihoods. Likelihoods on initiate relations, threat scenarios and unwanted incidents are most commonly given as frequencies, while likelihoods on leads-to relations are typically given as probabilities.

In Figure 2 we have assigned the frequency value once every year ([1 : 1year]) to the relation initiating the threat scenario 'Outage of two or more transmission lines in the north/south corridor' and probability 0.07 to the leads-to relation from this scenario to the threat scenario 'Grid overload in Sweden causes multiple outages'.

In Figure 2, we have also assigned a consequence value to each impact relation. In this example we use the following consequence scale: minor, moderate, major, critical and catastrophic. In a risk analysis such qualitative values are often mapped to concrete events. A minor consequence can for example correspond to a blackout affecting few people for a short duration, while a catastrophic consequence can be a blackout affecting more than a million people for several days.

In Figure 2 we have assigned the consequence value 'critical' to the impact relation from the unwanted incident 'Blackout in southern Sweden' to the asset 'Power production in Sweden' and the consequence value 'moderate' to the impact relation from the incident 'Minor area blackout'.

3. The textual syntax of CORAS diagrams

The graphical syntax of the CORAS language has been carefully designed to maximise the usability of the language. Although helpful in practical modelling situations, the graphical syntax is rather cumbersome to work with when defining the semantics and rules for the CORAS language. For this purpose we also provide an abstract textual syntax.[1] The abstract textual syntax for CORAS diagrams is defined in EBNF [19] as follows: [2]

$$\begin{aligned}
\mathit{diagram} &= \{\{\mathit{vertex}\}^-, \{\mathit{relation}\}\};\\
\mathit{vertex} &= \mathit{threat} \mid \mathit{threat\ scenario} \mid \mathit{unwanted\ incident} \mid \mathit{asset};\\
\mathit{relation} &= \mathit{initiate} \mid \mathit{leads\text{-}to} \mid \mathit{impact};\\
\mathit{initiate} &= \mathit{threat} \xrightarrow{[\mathit{vulnerability\ set}][\mathit{likelihood}]} \mathit{threat\ scenario} \mid\\
&\qquad \mathit{threat} \xrightarrow{[\mathit{vulnerability\ set}][\mathit{likelihood}]} \mathit{unwanted\ incident};
\end{aligned}$$

[1]Strictly speaking, we define here the textual syntax of CORAS threat diagrams only. See [6] for the textual syntax for the full language.

[2]Note that we lose information when we represent vertices and relations of a diagram by a single set. To avoid this we could have used a separate set for each of the six types of vertices. In that case whether a relation is an initiate, leads-to or impact relation is uniquely determined by the type of its argument vertices. However, to avoid introducing a more complicated notation that brings no real benefit, we choose the simpler notation and use syntactic variables when we need to distinguish types of vertices (see Table 1 on page 9.)

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

A CORAS diagram, as formalised in the EBNF above, consists of a finite non-empty set of vertices and a finite set of relations between them. The vertices correspond to the threats, threat scenarios, unwanted incidents and assets. The relations are of three kinds: initiate, leads-to and impact.

4. The structured semantics of CORAS diagrams

The semantics of the CORAS language is defined by a formal translation of any CORAS diagram into a paragraph in English. By structured in this context we mean that any CORAS diagram may be schematically translated (e.g. by a computer) element for element. The resulting paragraph in English should be understandable also for non-technical people. We obviously also see the value of a more mathematical semantics, but this is an issue of further work.

The semantics of a CORAS diagram is defined in terms of two steps:

1. The translation of the diagram into its textual syntax.

2. The translation of its textual syntax into its meaning as a paragraph in English.

The semantics enables the user of CORAS to extract the meaning of an arbitrary CORAS diagram by applying first (1), then (2). In both steps, a diagram is translated vertex by vertex and relation by relation.

For simplicity we introduce a number of syntactic variables. As indicated by Table 1, we use a (possibly with decorations, e.g. subscripts) to range over assets, dt (possibly with decorations) to range over deliberate threats, etc.

Syntactic category	Variable
asset	a
deliberate threat	dt
accidental threat	at
non-human threat	nht
threat scenario	ts
unwanted incident	ui
vertex	v
relation	r

Syntactic category	Variable
vulnerability set	V
likelihood	l
conditional likelihood	cl
consequence	c
relation between v_1 and v_2	$v_1 \rightarrow v_2$

Table 1: Naming conventions

4.1. *Translation from the graphical into the textual syntax*

To translate a vertex from the graphical to the textual syntax, the icon is simply replaced by its label, possibly decorated by a frequency. Relations are represented as arrows from one label to another. Take for example the diagram in Figure 3. Replacing

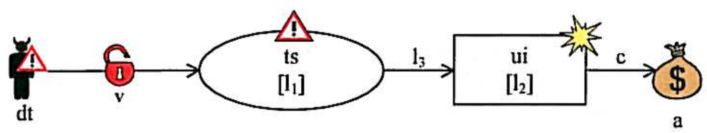

Figure 3: Example CORAS diagram

the icons for the deliberate threat, vulnerability, threat scenario, unwanted incident and asset with their labels gives us $dt, v, ts(l_1), ui(l_2)$ and a. The translation of the relations are

$$dt \xrightarrow{v} ts, ts \xrightarrow{l_3} ui, ui \xrightarrow{c} a$$

Hence, the textual representation of the diagram in Figure 3 is

$$\{dt, ts(l_1), ui(l_2), a, dt \xrightarrow{v} ts, ts \xrightarrow{l_3} ui, ui \xrightarrow{c} a\}$$

4.2. *Translation from the textual syntax into English*

In the second step of the structured semantics we apply the semantic function $[\![\_]\!]$ to the textual expressions resulting from the first step, obtaining a sentence in English for each expression. We start by defining the semantic function for the vertices, and then move on to the three kinds of relations and finally, the various annotations.

Complete threat diagram

$$[\![\{v_1, \ldots, v_n, r_1, \ldots, r_m\}]\!] := [\![v_1]\!] \ldots [\![v_n]\!][\![r_1]\!] \ldots [\![r_m]\!]$$

Vertices

$[\![dt]\!] :=$ ***dt*** is a deliberate threat.
$[\![at]\!] :=$ ***at*** is an accidental threat.
$[\![nht]\!] :=$ ***nht*** is a non-human threat.
$[\![a]\!] :=$ $\underset{\cdot}{a}$ is an asset.
$[\![ts]\!] :=$ Threat scenario ***ts*** occurs with undefined likelihood.
$[\![ts(l)]\!] :=$ Threat scenario ***ts*** occurs with $[\![l]\!]$.
$[\![ui]\!] :=$ Unwanted incident ***ui*** occurs with undefined likelihood.
$[\![ui(l)]\!] :=$ Unwanted incident ***ui*** occurs with $[\![l]\!]$.

Initiate relation

$[\![v_1 \rightarrow v_2]\!] := v_1$ initiates v_2 with undefined likelihood.
$[\![v_1 \xrightarrow{l} v_2]\!] := v_1$ initiates v_2 with $[\![l]\!]$.
$[\![v_1 \xrightarrow{V} v_2]\!] := v_1$ exploits $[\![V]\!]$ to initiate v_2 with undefined likelihood.
$[\![v_1 \xrightarrow{V\ l} v_2]\!] := v_1$ exploits $[\![V]\!]$ to initiate v_2 with $[\![l]\!]$.

Leads-to relation

$[\![v_1 \rightarrow v_2]\!] := v_1$ leads to v_2 with undefined conditional likelihood.
$[\![v_1 \xrightarrow{l} v_2]\!] := v_1$ leads to v_2 with $[\![cl]\!]$.
$[\![v_1 \xrightarrow{V} v_2]\!] := v_1$ leads to v_2 with undefined conditional likelihood, due to $[\![V]\!]$.
$[\![v_1 \xrightarrow{V\ l} v_2]\!] := v_1$ leads to v_2 with $[\![cl]\!]$, due to $[\![V]\!]$.

Impact relation

$[\![v_1 \rightarrow v_2]\!] := v_1$ impacts v_2.
$[\![v_1 \xrightarrow{c} v_2]\!] := v_1$ impacts v_2 with $[\![c]\!]$.

Annotations

$[\![V]\!] :=$ vulnerabilities V
$[\![l]\!] :=$ likelihood ***l***
$[\![cl]\!] :=$ conditional likelihood ***cl***
$[\![c]\!] :=$ consequence **c**

4.3. Example translation

We use the CORAS diagram in Figure 2 to illustrate the translation of a diagram using the structured semantics There are 12 vertices: three threats, six threat scenarios, two unwanted incidents and one asset. Translating all the vertices starting with the uppermost 'path' through the diagram gives us:

1. 'Operator mistake' is an accidental threat.
2. Threat scenario 'Outage of two or more transmission lines in the north/south corridor' occurs with undefined likelihood.
3. Threat scenario 'Grid overload in Sweden causes multiple outages' occurs with likelihood '1:10 years'.
4. Unwanted incident 'Blackout in southern Sweden' occurs with likelihood '1:20 years'.
5. 'Power production in Sweden' is an asset.
6. 'Sabotage at nuclear plant' is a deliberate threat.
7. Threat scenario 'Reduced nuclear availability' occurs with undefined likelihood.
8. 'Lack of rain in Sweden' is a non-human threat.
9. Threat scenario 'Low hydro availability in Sweden' occurs with likelihood '1:5 years'.
10. Threat scenario 'Capacity shortage' occurs with likelihood '1:4 years'.
11. Threat scenario 'Unstable network' occurs with likelihood '1:10 years'.
12. Unwanted incident 'Minor area blackout' occurs with likelihood '1:10 years'.

The diagram in Figure 2 contains 12 relations: three initiate relations of which one is annotated with a vulnerability, seven leads-to relations of which six are annotated with probabilities and two are annotated with vulnerabilities, and two impact relations annotated with consequences. Translating all the relations starting with the uppermost 'path' gives us:

1. 'Operator mistake' exploits vulnerability 'Interface bottleneck' to initiate 'Outage of two or more transmission lines in the north/south corridor' with likelihood '1:1 year'.
2. 'Outage of two or more transmission lines in the north/south corridor' leads to 'Grid overload in Sweden causes multiple outages' with conditional likelihood '0.07', due to vulnerability 'Failed load shedding'.
3. 'Outage of two or more transmission lines in the north/south corridor' leads to 'Minor area blackout' with conditional likelihood '0.1'.

4. 'Grid overload in Sweden causes multiple outages' leads to 'Blackout in southern Sweden' with undefined conditional likelihood, due to vulnerability 'Failed area protection'.
5. 'Blackout in southern Sweden' impacts 'Power production in Sweden' with consequence 'critical'.
6. 'Sabotage at nuclear plant' initiates 'Reduced nuclear availability' with likelihood '1:20 years'.
7. 'Reduced nuclear availability' leads to 'Capacity shortage' with conditional likelihood '1.0'.
8. 'Lack of rain in Sweden' initiates 'Low hydro availability in Sweden' with undefined likelihood.
9. 'Low hydro availability in Sweden' leads to 'Capacity shortage' with conditional likelihood '1.0'.
10. 'Capacity shortage' leads to 'Unstable network' with conditional likelihood '0.4'.
11. 'Unstable network' leads to 'Minor area blackout' with conditional likelihood '0.2'.
12. 'Minor area blackout' impacts 'Power production in Sweden' with consequence 'moderate'.

5. The CORAS calculus

By requesting the participants in brainstorming sessions to provide likelihood estimates both on threat scenarios, unwanted incidents and relations, the risk analyst[3] may uncover potential inconsistencies. The possibility for recording such inconsistencies is important from a methodological point of view. It helps identify misunderstandings and pinpoint aspects of the diagrams that must be considered more carefully.

In order to facilitate reasoning about CORAS diagrams we introduce the CORAS calculus consisting of some helpful deduction rules. The rules are of the following form.

$$\frac{P_1 \quad P_2 \quad \dots \quad P_i}{C_1 \quad C_2 \quad \dots \quad C_j}$$

We refer to $P_1, \dots, P_i$ as the premises and to $C_1, \dots, C_j$ as the conclusions. The interpretation is as follows: if the premises are valid so are the conclusions.

In general, calculating the likelihood of a vertex v from the likelihoods of other vertices and connecting relations may be challenging. In fact, in practise we may often only be able to deduce upper or lower bounds, and in some situations the diagrams have to be decomposed or even partly redrawn to make likelihood calculations feasible. However, for the purpose of this paper with its focus on mutual dependency, we need only the basic

[3]The person in charge of the risk analysis and the leader of the brainstorming session.

rules as presented below. Their validity follow straightforwardly from the structured semantics of CORAS diagrams and elementary probability theory.

The initiate rule captures the semantics of the initiate relation. The likelihood of reaching the vertex v from the threat t is equal to the probability of the connecting initiate relation. The new vertex $v \sqcap t$ may be seen as a decomposition of the vertex v, namely the 'subset of the scenarios/incidents v caused by the threat t'[4].

Rule 1 (Initiate). *If the vertices t and v are related by initiate, we have:*

$$\frac{t \xrightarrow{l} v}{(t \sqcap v)(l)}$$

The leads-to rule formalises the conditional probability semantics embedded in the leads-to relation. In elementary statistics the rule corresponds to the multiplication law of probability. The likelihood of the intersection $v_1 \sqcap v_2$ is equal to the likelihood of v_1 multiplied by the conditional likelihood of v_2 given the likelihood of v_1. Again, the new vertex $v_1 \sqcap v_2$ may be seen as a decomposition of the vertex v_2.

Rule 2 (Leads-to). *If the vertices v_1 and v_2 are related by leads-to, we have:*

$$\frac{v_1(f) \quad v_1 \xrightarrow{cl} v_2}{(v_1 \sqcap v_2)(f \cdot cl)}$$

If two vertices are mutually exclusive the likelihood of their union is equal to the sum of their individual likelihoods.

Rule 3 (Mutually exclusive vertices). *If the vertices v_1 and v_2 are mutually exclusive, we have:*

$$\frac{v_1(f_1) \quad v_2(f_2)}{(v_1 \sqcup v_2)(f_1 + f_2)}$$

Finally, if two vertices are statistically independent the likelihood of their union is equal to the sum of their individual likelihoods minus the likelihood of their intersection.

Rule 4 (Independent vertices). *If the vertices v_1 and v_2 are statistically independent, we have:*

$$\frac{v_1(f_1) \quad v_2(f_2)}{(v_1 \sqcup v_2)(f_1 + f_2 - f_1 \cdot f_2)}$$

Consider once more the diagram in Figure 2. The frequency of the threat scenario 'Grid overload in Sweden causes multiple outages' has been estimated to '1:10 years'. To check this estimate we may first use Rule 1 to establish '1:1 year' as a minimum[5] frequency of the threat scenario 'Outage of two or more transmission lines in the north/south corridor'.

[4] We us the ⊓-symbol to signal that this is a kind of intersection. We do not use the standard intersection symbol ∩ to avoid confusion when we later use standard set-notation to manipulate diagrams.

[5] If we assume that the diagram is complete, i.e. if there are no other threats that may cause this threat scenario, we may deduce '1:1 year' as the exact frequency, and not just as a minimum frequency.

If we then use Rule 2 we get a frequency close to '1:14 years' as an estimated lower bound for 'Grid overload in Sweden causes multiple outages'. From this we may conclude that either something is wrong or the diagram is not complete. In the latter case there are other threat scenarios leading to 'Grid overload in Sweden causes multiple outages', that we have not yet identified. In the former case, the diagram needs to be corrected. In this paper we assume the former case, and in Figure 4 (the dependent version of the diagram in Figure 2) the probability 0.07 on the leads-to relation from 'Outage of two or more transmission lines in the north/south corridor' has been removed.

Similarly, since it is reasonable to assume that 'Sabotage at nuclear plant' and 'Low hydro availability in Sweden' are statistically independent events, we may use Rules 1, 2 and 4 to conclude that the estimated frequency of '1:4 years' in the case of 'Capacity shortage' is consistent with the rest of the diagram.

6. Dependent CORAS

A security risk analysis may target any system, including systems of systems. Even in a relatively small analysis there is a huge amount of information to process. When the analysis targets complex systems we need means to decompose the analysis into separate parts or modules that can be carried out independently. Moreover, it must be possible to combine the analysis results of these separate parts into a valid risk picture for the system as a whole. When there is mutual dependency between parts, and we want to deduce the effect of composition, we need means to distinguish mutual dependency that is well-founded from mutual dependency that is not (i.e., avoid circular reasoning).

This problem of modularity is not specific to the field of risk analysis. It is in fact at the very core of a reductionistic approach to science and life in general. Assumption-guarantee reasoning [20, 27] has been suggested as an approach to facilitate modular system development. In the assumption-guarantee approach specifications consists of two parts, an assumption and a guarantee:

- The assumption specifies the assumed environment for the specified system.
- The guarantee specifies how the system is guaranteed to behave when executed in an environment that satisfies the assumption.

Assumption-guarantee specifications are useful for specifying systems that interact with an environment. The idea is that the specification should state explicitly what the system requires or assumes of its environment. Dependent CORAS that is introduced below is inspired by the assumption-guarantee approach.

When two risk scenarios are mutually dependent, one scenario is in the context of the other and vice versa. By stating explicitly which aspects in the context that affect a scenario we get a means to capture dependency. Dependent CORAS extends the basic CORAS diagrams with facilities for documenting assumptions about external threats and incidents of relevance for the threat scenario being analysed. Such extended CORAS diagrams are in the following referred to as dependent CORAS diagrams.

6.1. Modelling context dependencies

The power sector in southern Sweden can be seen as a sub-system of the Nordic power sector. The power sectors of Sweden, Denmark, Norway and Finland are mutually dependent. Hence, the risk of a blackout in southern Sweden can be affected by the stability of the power sectors in the neighbouring countries. These neighbouring sectors are not part of the target of analysis as specified previously and therefore not analysed as such, but we should still take into account that the risk level of the power sector in southern Sweden depends on the risk levels of the power sectors in the Nordic countries. We do this by stating explicitly which external threat scenarios and incidents we take into consideration.

The dependent CORAS diagram in Figure 4 takes into consideration the external threat scenario 'Low hydro availability in Z', the leads-to relation connecting it to the threat scenario 'High import in Z from Sweden', as well as the external incident 'Minor export area blackout in Z'. The diagram states that high import of Swedish power to a neighbouring country contributes to the threat scenario 'Grid overload in Sweden causes multiple outages' and that a blackout in the export area of a neighbouring country contributes to the unwanted incident 'Minor area blackout' in southern Sweden. There may of course be many other threats and incidents of relevance in this setting, but this diagram makes no further assumptions. We refer to the content of the rectangular container including the crossing relations and the vertices on the border as the target scenario, and to the the rest as the context scenario.

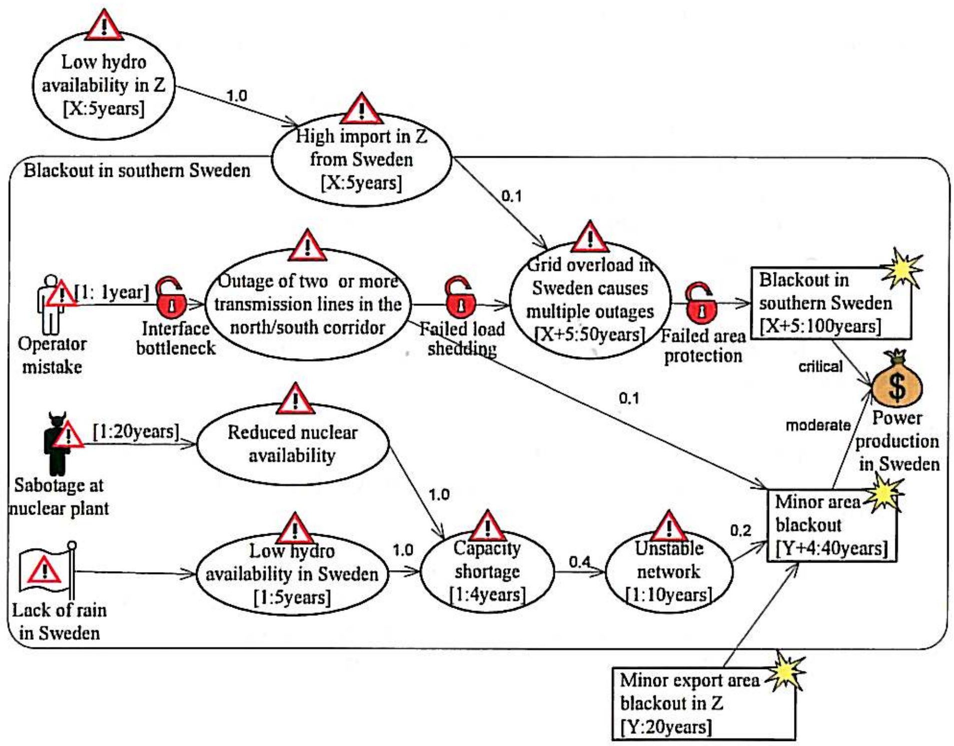

Figure 4: Dependent CORAS diagram for the power sector in southern Sweden

In order to facilitate reuse we keep our assumptions about the environment as generic as possible. By parameterising[6] the name of the offending power sector, we may later combine the risk analysis results for the Swedish power sector with results from any of the other Nordic countries. At this point in the analysis we leave open the likelihoods of the assumed external events. We must therefore also leave open the likelihoods of the events inside the target that are affected by the external events.

6.2. Textual syntax of dependent CORAS diagrams

In the textual syntax a dependent CORAS diagram is written

$$C \triangleright T$$

where C and T are referred to as the context (which may be empty) and target scenarios, respectively. In the EBNF this is captured as follows:

$$\begin{aligned} \textit{dependent diagram} &= \textit{context scenario} \triangleright \textit{target scenario}; \\ \textit{context scenario} &= \textit{diagram} \mid \textit{empty}; \\ \textit{target scenario} &= \textit{diagram}; \end{aligned}$$

6.3. Structured semantics of dependent CORAS diagrams

The translation from the graphical into the textual syntax is as for ordinary CORAS diagrams with the exception that the vertices and relations are split between the target and context scenarios. Any vertex or relation that is completely inside the rectangular container belongs to the target scenario; any that is completely outside belongs to the context scenario; the relations that cross the rectangular container (e.g. in Figure 4, the relation from 'Minor export area blackout in Z') belong to the target scenario; the vertices of the border (e.g.'High import in Z from Sweden' in Figure 4) belong to the target scenario; the relations that point to a vertex on the border belong to the context scenario (e.g. in Figure 4 the relation from 'Low hydro availability in Z').

We are only interested in the textual "diagrams" that can be obtained from graphical diagrams as described above. In the following we assume that every dependent CORAS diagram in the textual syntax fulfils this constraint. Hence, we do not consider expressions fulfilling the EBNF in which the target for example contains the relation $v \rightarrow v'$ but not the vertex v' to be syntactically correct.

The translation from the textual syntax into English via the semantic function is almost unchanged. We need only one additional rule:

$$[\![C \triangleright T]\!] := [\![T]\!] \text{ assuming } [\![C]\!] \text{ to the extent there are explicit dependencies}$$

The suffix 'to the extent there are explicit dependencies' is significant. It implies that if there are no relations connecting C to T explicitly, we do not gain anything from C. For example, with respect to Figure 4, since there are relations connecting the vertex 'Low hydro availability in Z' to the vertex 'Blackout in southern Sweden' we may use the former to deduce the likelihood of the latter. On the other hand, since there are no relations connecting C to the vertex 'Reduced nuclear availability' the assumption C is of no significance for this particular vertex.

[6]The syntactic definition of the CORAS language in Section 3 does not take parameterisation into account. This is however a straightforward generalisation.

6.4. Extending the calculus to handle dependent CORAS diagrams

In order to facilitate reasoning about dependent CORAS diagrams we extend the CORAS calculus to reason about dependency. We may for example use the calculus to argue that an overall risk scenario captured by a dependent CORAS diagram D follows from n dependent CORAS diagrams $D_1, \ldots, D_n$ describing mutually dependent sub-scenarios.

In order to extend the CORAS calculus with rules addressing dependent CORAS diagrams, we first introduce some helpful notation. A set of connected relations

$$P = \{v_1 \rightarrow v_2, v_2 \rightarrow v_3, \ldots, v_{n-1} \rightarrow v_n\}$$

is a path. We say that P is a path in a diagram D if $P \subseteq D$. We write $v_1 \rightarrow P$ and $P \rightarrow v_n$ to state that P is a path commencing in vertex v_1 and ending in vertex v_n, respectively.

Let D be a dependent CORAS diagram and let $T, T' \subseteq D$. A vertex $v \in T'$ is independent of T if for any path $P \subseteq T \cup T'$ and vertex $v' \in T \cup T'$

$$v' \rightarrow P \wedge P \rightarrow v \Rightarrow v' \notin T$$

Hence, v is independent of T if there are no paths to v in the diagram commencing from a vertex v' in T.

The sub-diagram T' is independent of the sub-diagram T if each vertex in T' is independent of T in which case we write $T \ddagger T'$. Hence, the target scenario T is independent of the context scenario C if each vertex in T is independent of C.

The following rule states that if we have deduced T assuming C, and T is independent of C, then we may deduce T.

Rule 5 (Independence).

$$\frac{C \triangleright T \quad C \ddagger T}{\triangleright T}$$

From the second premise it follows that there is no path from C to a vertex in T. Since the first premise states T assuming C to the extent there are explicit dependencies, we may deduce T.

The following rule allows us to remove a part of the context that is not connected to the rest.

Rule 6 (Context simplification).

$$\frac{C \cup C' \triangleright T \quad C \ddagger C' \cup T}{C' \triangleright T}$$

The second premise implies that there are no paths from C to the rest of the diagram. Hence, the validity of the first premise does not depend upon C in which case the conclusion is also valid.

The following rule allows us to remove part of the target scenario as long as it is not situated in-between the context and the part of the target we want to keep.

Rule 7 (Target simplification).

$$\frac{C \triangleright T \cup T' \quad T' \ddagger T}{C \triangleright T}$$

The second premise implies that there is no path from C to T via T'. Hence, the validity of the first premise implies the validity of the conclusion.

To make use of these rules, when scenarios are composed, we also need modus ponens for the $\triangleright$-operator.

Rule 8 (Modus ponens).

$$\frac{C \triangleright T \quad \triangleright C}{\triangleright T}$$

Hence, if T holds assuming C to the extent there are explicit dependencies, and we can also show C, then it follows that T.

6.5. Reasoning about mutually dependent systems

To illustrate how the CORAS calculus can be used to reason about risks in mutually dependent systems we consider once more the power sector. This time we widen the scope to include the power sector in southern Norway in addition to that of southern Sweden. Figure 5 presents a dependent CORAS diagram for the power sector in southern Norway. As in the example with southern Sweden we parameterise on likelihoods and offending power sector.

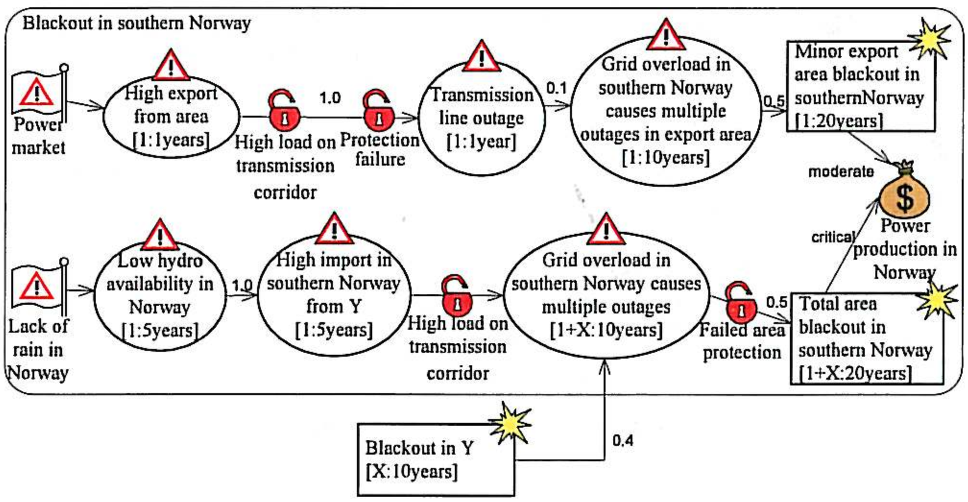

Figure 5: Dependent CORAS diagram for the power sector in southern Norway

Using the CORAS calculus on the dependent CORAS diagrams for the two target scenarios, we may deduce the diagram for the combined target scenario 'Blackout in southern Sweden and Norway', presented in Figure 6. The main clue is of course that

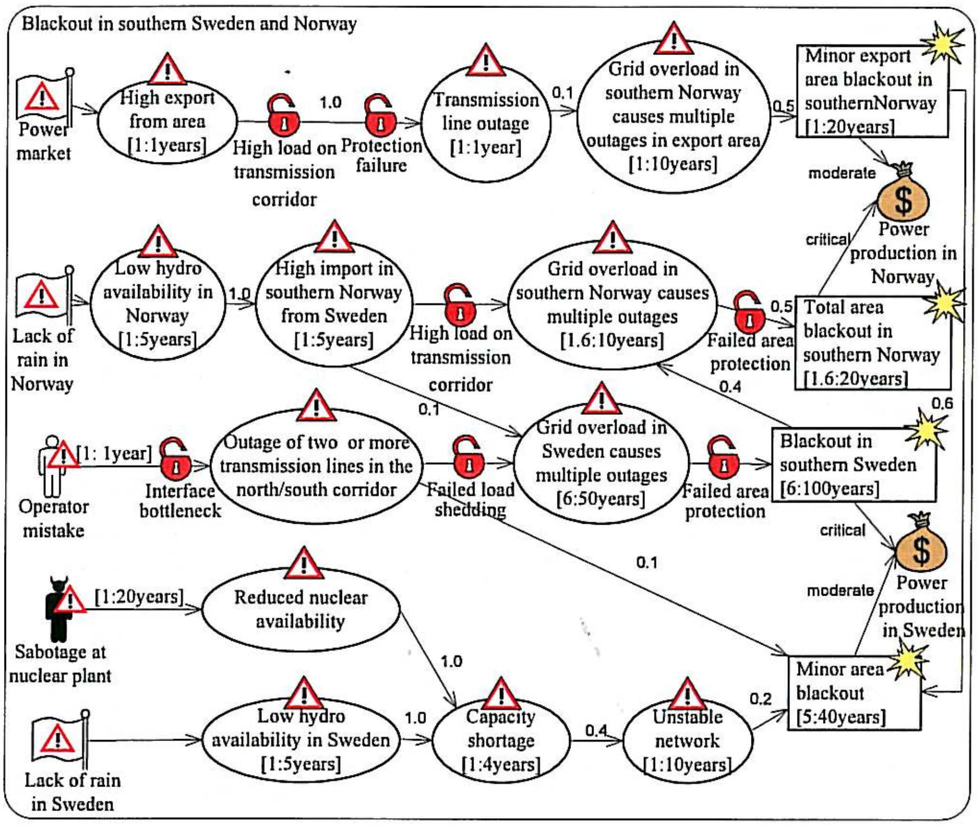

Figure 6: The threats for the composite system

the paths of dependencies between the two diagrams are well-founded: when we follow a path backwards we will cross between the two diagrams only a finite number of times.

More rigorously, assume the validity of

$$C_1 \triangleright T_1, \qquad C_2 \triangleright T_2$$

obtained from the diagrams in Figures 4 and 5, respectively, via the substitutions

$$\{X \mapsto 1, Y \mapsto 1, Z \mapsto \mathit{Norway}\}, \qquad \{X \mapsto 1.6, Y \mapsto \mathit{Sweden}\}$$

We want to deduce

$$\triangleright T_1 \cup T_2$$

which corresponds to the diagram in Figure 6. We may understand the union operator on scenarios as a logical conjunction. Hence, from $\triangleright S_1$ and $\triangleright S_2$ we may deduce $\triangleright S_1 \cup S_2$, and the other way around.

The context C_1 is naturally decomposed into C_1' and C_1'' where C_1' is the part connected to 'Grid overload in Sweden causes multiple outages' and C_1'' is the part connected to 'Minor area blackout'. We may use Rule 7 and 5 to deduce $\triangleright C_1'$ from $C_2 \triangleright T_2$ since $C_1' \subset T_2$ and C_1' does not depend on C_2. We may deduce $\triangleright C_1''$ from $C_2 \triangleright T_2$ accordingly, in which case we have deduced $\triangleright C_1$. But then we have $\triangleright T_1$ by Rule 8. It follows from $\triangleright T_1$ that 'Blackout in southern Sweden' occurs with likelihood '6:100 years'. This corresponds to '1.6:20 years', which means that we may deduce $\triangleright C_2$ from $\triangleright T_1$. But then we have $\triangleright T_2$ by Rule 8 in which case we have deduced the validity of the diagram in Figure 6.

Note that we may also deduce useful things about diagrams with cyclic dependencies. For example, if we add a dependency from 'Minor area blackout' to 'Minor export area blackout in southern Norway' in the context of the diagram in Figure 5, we may still use the CORAS calculus to deduce useful information about the part of Figure 6 that does not depend on the cycle (i.e., cannot be reached from the two vertices connected by the cycle).

Let C_2' and C_2'' be the decomposition of C_2, in the augmented diagram, where C_2' is the part connected to 'Minor export area blackout in southern Norway' and C_2'' is the part connected to 'Grid overload in Norway causes multiple outages'. By applying rules 5 to 8 we may deduce

$$\triangleright T_1 \setminus C_2' \cup T_2 \setminus C_1''$$

First we deduce $\triangleright C_1'$ as before since C_1' is not affected by the new dependency. We then apply Rule 7, 6 and 8 to deduce $\triangleright T_1 \setminus C_2'$ and subsequently $\triangleright C_2''$ from $\triangleright T_1 \setminus C_2'$. From $\triangleright C_2''$ and $C_2 \triangleright T_2$ we deduce $\triangleright T_2 \setminus C_1''$, accordingly.

7. Making composite diagrams compact

When we combine two or more dependent CORAS diagrams the result is a diagram consisting of all the elements from the original diagrams. In order to present the combined threat scenarios in a comprehensible manner, an analyst must be able to reduce the level of detail without compromising the obtained result. This may involve combining or eliminating vertices as well as drawing new relations. The following deduction rules impose formal constraints on this process; in particular, they describe how likelihoods may be recalculated in order to be consistent with the original diagram.

While we in some cases are able to use Rules 1-4 to calculate the likelihood of a vertex or a relation, most cases are more complex. We start with a rule for decomposing relations.

Rule 9 (Relation decomposition). *If the vertices v_1 and v_2 are related by initiate or leads-to, we have:*

$$\frac{v_1 \xrightarrow{l} v_2}{v_1 \xrightarrow{l} (v_2 \sqcap v_1) \qquad (v_2 \sqcap v_1) \xrightarrow{1} v_2}$$

If the likelihood of getting from v_1 to v_2 is l, then the likelihood of getting from v_1 to v_2 via v_1 is also l. Moreover, since $v_2 \sqcap v_1$ may be seen as a decomposition of v_2 the likelihood of getting from $v_2 \sqcap v_1$ to v_2 is obviously 1.

Rule 10 formalises the transitivity of relations implied by the semantics.

Rule 10 (Transitivity). *Given relations from v_1 to v_2 and from v_2 to v_3, both of which are assigned likelihoods; we then have:*

$$\frac{v_1 \xrightarrow{l_1} v_2 \qquad v_2 \xrightarrow{l_2} v_3}{v_1 \xrightarrow{l_1 \cdot l_2} (v_3 \sqcap v_2)}$$

Note that replacing the conclusion with $v_1 \xrightarrow{l_1 \cdot l_2} v_3$ would not be sound since there may also be relations from v_1 to v_3 that do not involve v_2. Note also that the validity of Rule 10 builds on the understanding that if there is a direct relation from v_1 to v_2, then the likelihood of this relation contains the likelihoods of all indirect routes from v_1 to v_2. Hence, if $v_1 \xrightarrow{l} v_2$ and $v_1 \xrightarrow{l_1} v \wedge v \xrightarrow{l_2} v_2$ then $l_1 \cdot l_2$ is already included in the probability l of the direct relation.

Having combined two vertices using Rules 3 or 4, we want to deduce the likelihoods of the relations terminating or originating in the composite vertex from the likelihoods of the relations connected to the original vertices. We distinguish between the mutually exclusive and the statistically independent cases. The following two rules address the case where two relations from the same vertex terminate in each of the two vertices that are composed.

Rule 11 (Composing relations to mutually exclusive vertices). *If the vertices v_1 and v_2 are mutually exclusive, we have:*

$$\frac{v \xrightarrow{l_1} v_1 \qquad v \xrightarrow{l_2} v_2}{v \xrightarrow{l_1 + l_2} (v_1 \sqcup v_2)}$$

Rule 12 (Composing relations to statistically independent vertices). *If the vertices v_1 and v_2 are statistically independent, we have:*

$$\frac{v \xrightarrow{l_1} v_1 \qquad v \xrightarrow{l_2} v_2}{v \xrightarrow{l_1 + l_2 - l_1 \cdot l_2} (v_1 \sqcup v_2)}$$

The case where two relations to the same vertex originate in each of the two vertices that are composed is covered by the next two rules.

Rule 13 (Composing relations from mutually exclusive vertices). *If the vertices v_1 and v_2 are mutually exclusive, we have:*

$$\frac{v_1(f_1) \qquad v_2(f_2) \qquad v_1 \xrightarrow{l_1} v \qquad v_2 \xrightarrow{l_2} v}{(v_1 \cup v_2) \xrightarrow{\frac{f_1 \cdot l_1 + f_2 \cdot l_2}{f_1 + f_2}} v}$$

Rule 14 (Composing relations from statistically independent vertices). *If the vertices v_1 and v_2 are statistically independent, we have:*

$$\frac{v_1(f_1) \qquad v_2(f_2) \qquad v_1 \xrightarrow{l_1} v \qquad v_2 \xrightarrow{l_2} v}{(v_1 \cup v_2) \xrightarrow{\frac{f_1 \cdot l_1 + f_2 \cdot l_2 - f_1 \cdot l_1 \cdot f_2 \cdot l_2}{f_1 + f_2 - f_1 \cdot f_2}} v}$$

Finally, we have to handle how vertex composition affects the impact relation. We have two cases: either two assets, or two unwanted incidents or threat scenarios are composed.

Rule 15 (Composing impact relations to composite asset). *If v impacts both a_1 and a_2, $\oplus$ is the operator for consequence summation, and $a_1 \cup a_2$ is the composition of a_1 and a_2, we have:*

$$\frac{v \xrightarrow{c_1} a_1 \qquad v \xrightarrow{c_2} a_2}{v \xrightarrow{c_1 \oplus c_2} (a_1 \cup a_2)}$$

Rule 16 (Composing impact relations from composite vertex). *If both v_1 and v_2 impact the asset a, and $\oplus$ is the operator for consequence summation, we have:*

$$\frac{v_1 \xrightarrow{c_1} a \qquad v_2 \xrightarrow{c_2} a}{(v_1 \cup v_2) \xrightarrow{c_1 \oplus c_2} a}$$

7.1. Compacting the example diagram

The threat diagram in Figure 7 has resulted from compacting the diagram in Figure 6. The threat scenario 'High export leads to grid overload in Norway' in the uppermost branch is the composition of the threat scenarios 'High export from area', 'Transmission line outage' and 'Grid overload in southern Norway causes multiple outages'. By applying Rule 2 two times we calculate the frequency value of the composite vertex to be once every ten years. The unwanted incident 'Total area blackout in southern Norway' in the second branch is unchanged but is no longer related to the threat scenario 'Grid overload in southern Norway causes multiple outages' which has been removed. The unwanted incident 'Total area blackout via blackout in southern Sweden' is the composition of the unwanted incidents 'Blackout in southern Sweden' and 'Total area blackout in southern Norway'. The likelihood follows again by repeated use of Rule 2.

Since this incident affects both the original assets 'Power production in Norway' and 'Power production in Sweden' we have chosen to combine these two assets into 'Power production in Norway and Sweden' using Rules 15 and 16.

The scale of qualitative consequence values: minor, moderate, major, critical and catastrophic, is mapped to numbers 1 to 5. We let the function for calculating new

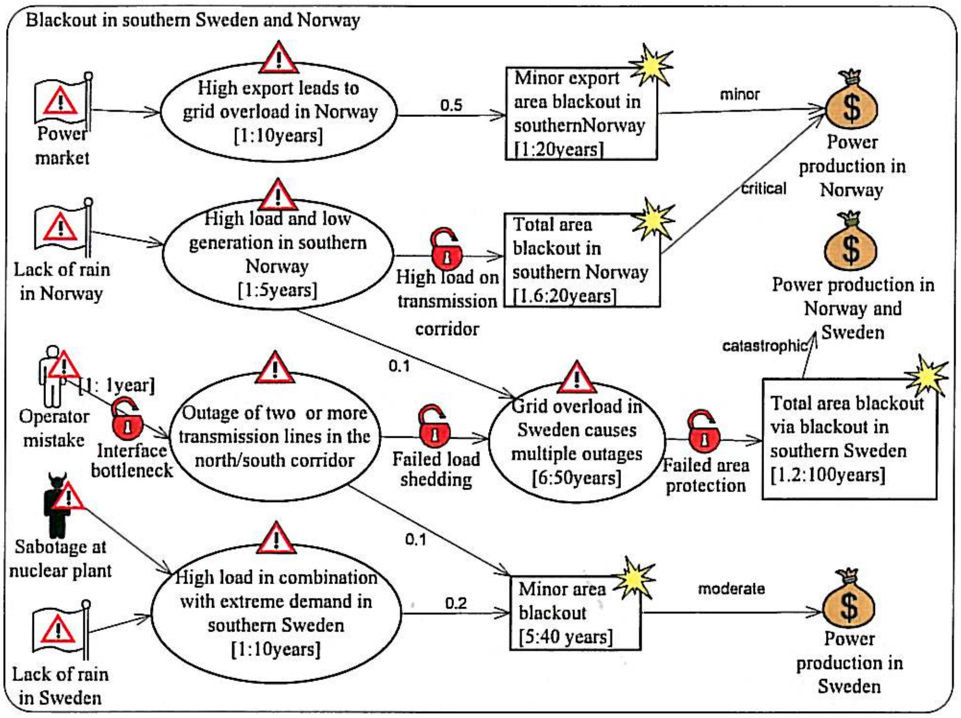

Figure 7: The threats for the compacted system

consequence values on relations to composite assets be: $c_1 \oplus c_2 = \min(\max(c_1, c_2) + 1, 5)$. Hence, the consequence value on the combined relation is $\min(\max(4, 4) + 1, 5) = 5$ which is mapped to catastrophic.

The threat scenario 'High load in combination with extreme demand in southern Sweden' is the result of first doing composition on the two threat scenarios 'Reduced nuclear availability' and 'Low hydro availability in Sweden' and then compose this vertex with the threat scenarios 'Capacity shortage' and 'Unstable network'.

Note that compacting a diagram with a nonempty context requires more care since we have to make sure that the context dependencies are not removed. However, although the rules are slightly more complex, the basic principles remain the same.

8. Related work

The CORAS language [7] originates from a UML [29] profile [24, 30] developed as a part of the EU funded research project CORAS (IST-2000-25031) [1]. The CORAS language has later been customised and refined in several respects, based on experiences from industrial case studies, and by empirical investigations [14, 15, 16].

The idea of applying specialised use cases for the purpose of threat identification was first proposed by McDermott and Fox [26, 25]. Sindre and Opdahl [35, 36] later explained how to extend use cases with misuse cases as a means to elicit security requirements. The CORAS language is inspired by misuse cases but is much richer and is tailored to support the risk analysis process, while misuse cases is used for requirements capture. There are a number of security oriented extensions of UML, e.g. UMLSec [21] and SecureUML [23]. These and related approaches have however all been designed to capture security properties and security aspects at a more detailed level than our language. Moreover, their focus is not on risk identification using structured brainstorming as in the case of CORAS.

There are several notations for modelling and analysing threats that are related to CORAS diagrams. One example is the fault tree notation used in Fault Tree Analysis (FTA) [17]. The top vertex in a fault tree may be thought of as an unwanted incident in the meaning of CORAS. The vertices further down the tree may be seen as threat scenarios or threats of which the relationships are captured by logical combinators. Hence, it may be argued that fault trees resembles CORAS diagrams. However, fault trees focus more on the logical decomposition of an incident into its constituents, and less on the causal relationship between events which is the emphasis in CORAS. Furthermore, CORAS diagrams may have more than one top vertex and can be used to model assets and consequences. Moreover, in CORAS likelihoods may be assigned to both vertices and relations, whereas in fault trees only the vertices have likelihoods. The likelihood of a vertex in a CORAS diagram can be calculated from the likelihoods of its parent vertices and connecting relations. The possibility to assign likelihoods to both vertices and relations have methodological benefits during brainstorming sessions because it may be used to uncover inconsistencies. Uncovering inconsistencies helps to clarify misunderstandings and pinpoint aspects of the diagrams that must be considered more carefully.

The structuring of events in a CORAS diagram also have similarities to Bayesian networks, but the probability model of Bayesian diagrams is more complex than the model used for computing probabilities in CORAS. A Bayesian network is used to specify a joint probability distribution for a set of variables [5]. It is a directed acyclic graph consisting of vertices that represent random variables and directed edges that specify dependence assumptions that must hold between the variables.

Event trees [18] are also related to the CORAS notation. Event Tree Analysis (ETA) focuses on illustrating the (forward) consequences of an event and the probabilities of these. CORAS diagrams on the other hand are typically developed backwards; from the assets towards the threats. Event trees are developed through success/failure gates for each defence mechanism that is activated.

Attack trees [34] are basically fault trees with a security-oriented terminology. Attack trees 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 vertex and different ways of achieving the goal as leaf vertices. CORAS diagrams differ from attack trees in the same way as CORAS diagrams differ from fault trees.

Several approaches to component-based hazard analysis describe system failure propagation by matching ingoing and outgoing failures of individual components. Giese et al. [13, 12] have defined a method for compositional hazard analysis of restricted UML component diagrams and deployment diagrams. They employ fault tree analysis to describe hazards and the combination of component failures that can cause them. For each component they describe a set of incoming failures, outgoing failures, local failures (events) and the dependencies between incoming and outgoing failures. Failure information of components can be composed by combining their failure dependencies. The approach of Giese et al. is similar to ours in the sense that it is partly model-based, as they do hazard analysis on UML diagrams. Their approach also has an assumption-guarantee flavour, as incoming failures can be seen as a form of assumptions. There are, however, also some important differences. The approach of Giese et al. is limited to hazard analysis targeting hazards caused by software or hardware failures. The CORAS method has a broader scope. It can be used both for security risk analysis and safety analysis. The CORAS threat diagrams documents not only system failures, but also the threats that may cause them, such as for example human errors, and the consequences the may lead to. Furthermore, the hazard analysis of Giese et al. is linked directly to the system components. CORAS diagrams are not linked directly to system components, as the target of an analysis may be restricted to an aspect or particular feature of a system. The modularity of dependent CORAS diagrams is achieved by the assumption-guarantee structure of the diagrams, not by the underlying component structure and composition is performed on risk analysis results, not components. CORAS does not require any specific type of system specification diagram as input for the risk analysis, the way the approach of Giese et al. does.

Papadoupoulos et al. [31] apply a version of Failure Modes and Effects Analysis (FMEA) [3] that focuses on component interfaces, to describe the causes of output failures as logical combinations of internal component malfunctions or deviations of the component inputs. They describe propagation of faults in a system by synthesising fault trees from the individual component results.

Kaiser et al. [22] propose a method for compositional fault tree analysis. Component failures are described by specialised component fault trees that can be combined into system fault trees via input and output ports.

Fenton et al. [10, 9] addresses the problem of predicting risks related to introducing a new component into a system, by applying Bayesian networks to analyse failure probabilities of components. They combine quantitative and qualitative evidence concerning the reliability of a component and use Bayesian networks to calculate the overall failure probability. As opposed to our approach, theirs is not compositional. They apply Bayesian networks to predict the number of failures caused by a component, but do not attempt to combine such predictions for several components.

9. Conclusion

We have presented a modular approach to the modelling and analysis of risk scenarios with mutual dependencies. The approach makes use of a graphical language for risk modelling with external dependencies. The graphical language is an extension of CORAS threat diagrams. The extended language, referred to as the Dependent CORAS langauge, has a well-defined syntax and a structured semantics.

A dependent CORAS diagram is divided into a context scenario and a target scenario. The context scenario makes assumptions about the external threats and incidents that are of relevance for the target scenario and the target scenario describes risk scenarios for the target under these assumptions. By making assumptions about the external behaviour explicit, we manage to decompose analyses of complex systems with mutual dependencies.

We have also introduced a set of rules for reasoning about dependent threat diagrams. The deduction rules can be applied to dependent diagrams, represented by their textual syntax, to resolve dependencies among them. Once dependencies are resolved, diagrams documenting component risks can be combined into composite diagrams documenting system level risks.

Finally, we have exemplified the modular approach to analyse risks in mutually dependent systems, by applying it to an example involving the power sectors in southern Sweden and Norway. We show that in this example we can resolve dependencies. In general, our approach is able to handle arbitrary long chains of dependencies, as long as they are well-founded.

9.1. Acknowledgements

The research for this paper has been partly funded by the DIGIT (180052/S10) and COMA (160317) projects of the Research Council of Norway, and partly through the SINTEF-internal project Rik og Sikker. We would like to thank Mass Soldal Lund for useful comments.

References

[1] J. Ø. Aagedal, F. den Braber, T. Dimitrakos, B. A. Gran, D. Raptis, K. Stølen, Model-based risk assessment to improve enterprise security, in: Proc. 6th International Enterprise Distributed Object Computing Conference (EDOC'02), IEEE Computer Society, 2002.

[2] M. Abadi, L. Lamport, Conjoining specifications, ACM Transactions on Programming Languages and Systems 17 (3) (1995) 507–534.

[3] A. Bouti, D. A. Kadi, A state-of-the-art review of FMEA/FMECA, International Journal of Reliability, Quality and Safety Engineering 1 (4) (1994) 515–543.

[4] G. Brændeland, H. E. I. Dahl, I. Engan, K. Stølen, Using dependent CORAS diagrams to analyse mutual dependency, in: Proc. 2nd International Workshop on Critical Information Infrastructures Security, vol. 5141 of LNCS, Springer, 2008.

[5] E. Charniak, Bayesian networks without tears: making Bayesian networks more accessible to the probabilistically unsophisticated, AI Magazine 12 (4) (1991) 50–63.
[6] H. E. I. Dahl, I. Hogganvik, K. Stølen, Structured semantics for the CORAS security risk modelling language, Tech. Rep. A970, SINTEF ICT (2007).
[7] 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, BT Technology Journal 25 (1) (2007) 101–117.
[8] G. Doorman, G. Kjølle, K. Uhlen, E. S. Huse, N. Flatbø, Vulnerability of the nordic power system, Tech. Rep. A5962, SINTEF Energy Research (2004).
[9] N. Fenton, M. Neil, Combining evidence in risk analysis using Bayesian networks, Agena White Paper W0704/01, Agena (2004).
[10] N. E. Fenton, P. Krause, M. Neil, Software measurement: Uncertainty and causal modeling, IEEE Software 19 (4) (2002) 116–122.
[11] F. Flentge, Project description, Tech. Rep. D 4.4.5, Integrated Risk Reduction of Information-based Infrastructure Systems (IRRIS) and Fraunhofer-Institut Autonome Intelligente Systeme (2006).
[12] H. Giese, M. Tichy, Component-based hazard analysis: Optimal designs, product lines, and online-reconfiguration, in: Proc. 25th International Conference on Computer Safety, Security and Reliability (SAFECOMP'06), vol. 4166 of LNCS, Springer, 2006.
[13] H. Giese, M. Tichy, D. Schilling, Compositional hazard analysis of UML component and deployment models, in: Proc. 23rd International Conference on Computer Safety, Reliability and Security (SAFECOMP'04), vol. 3219 of LNCS, Springer, 2004.
[14] I. Hogganvik, K. Stølen, On the comprehension of security risk scenarios, in: Proc. 13th International Workshop on Program Comprehension (IWPC'05), IEEE Computer Society, 2005.
[15] I. Hogganvik, K. Stølen, Risk analysis terminology for IT systems: Does it match intuition?, in: Proc. 4th International Symposium on Empirical Software Engineering (ISESE'05), IEEE Computer Society, 2005.
[16] I. Hogganvik, K. Stølen, A graphical approach to risk identification, motivated by empirical investigations, in: Proc. 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS'06), vol. 4199 of LNCS, Springer, 2006.
[17] IEC, Fault Tree Analysis (FTA), IEC 61025 (1990).
[18] IEC, Event Tree Analysis in Dependability management – Part 3: Application guide – Section 9: Risk analysis of technological systems, IEC 60300 (1995).
[19] ISO/IEC, Information technology – Syntactic metalanguage – Extended BNF, Tech. Rep. 14977, ISO/IEC (1996).
[20] C. B. Jones, Development methods for computer programmes including a notion of interference, Ph.D. thesis, Oxford University (1981).
[21] J. Jürjens, Secure Systems Development with UML, Springer, 2005.
[22] B. Kaiser, P. Liggesmeyer, O. Mäckel, A new component concept for fault trees, in: Proc. 8th Australian workshop on Safety critical systems and software (SCS'08), Australian Computer Society, Inc., 2003.
[23] T. Lodderstedt, D. A. Basin, J. Doser, SecureUML: A UML-based modeling language for model-driven security, in: Proc. 5th International Conference on UML (UML'02), vol. 2460 of LNCS, Springer, 2002.
[24] M. S. Lund, I. Hogganvik, F. Seehusen, K. Stølen, UML profile for security assessment, Tech. Rep. A03066, SINTEF ICT (2003).
[25] J. P. McDermott, Abuse-case-based assurance arguments, in: Proc. 17th Annual Computer Security Applications Conference (ACSAC'01), IEEE Computer Society, 2001.
[26] J. P. McDermott, C. Fox, Using abuse case models for security requirements analysis, in: Proc. 15th Annual Computer Security Applications Conference (ACSAC'99), IEEE Computer Society, 1999.
[27] J. Misra, K. M. Chandy, Proofs of networks of processes, IEEE Transactions on Software Engineering 7 (4) (1981) 417–426.
[28] NYISO, Final report: On the August 14, 2003 blackout, Tech. rep., New York Independent System Operator (NYISO) (2005).
[29] OMG, Unified Modeling Language Specification, version 2.0 (2004).
[30] OMG, UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms (2005).
[31] Y. Papadoupoulos, J. McDermid, R. Sasse, G. Heiner, Analysis and synthesis of the behaviour of complex programmable electronic systems in conditions of failure, Reliability Engineering and System Safety 71 (3) (2001) 229–247.

[32] C. E. Restrepo, J. S. Simonoff, R. Zimmerman, Unraveling geographic interdependencies in electric power infrastructure, in: Proc. 39th Annual Hawaii International Conference on System Sciences (HICSS'06), vol. 10, 2006.

[33] S. M. Rinaldi, J. P. Peerenboom, T. K. Kelly, Identifying, understanding and analyzing critical infrastructure dependencies, IEEE Control Systems Magazine (2001) 11–25.

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

[35] G. Sindre, A. L. Opdahl, Eliciting security requirements with misuse cases, in: Proc. 37th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS-PACIFIC'00), IEEE Computer Society, 2000.

[36] G. Sindre, A. L. Opdahl, Eliciting security requirements with misuse cases, Requirements Engineering 10 (1) (2005) 34–44.

[37] UCTE, Final report of the investigation committee on the 28 September 2003 blackout in Italy, Tech. rep., Union for the Coordination of Transmission of Electricity (UCTE) (2004).