Software Tools for Technology Transfer manuscript No.
(will be inserted by the editor)

Security risk analysis of system changes exemplified within the oil and gas domain

Atle Refsdal[1], Bjørnar Solhaug[1], Ketil Stølen[1,2]

[1] SINTEF ICT, Norway
[2] Department of Informatics, University of Oslo, Norway

The date of receipt and acceptance will be inserted by the editor

Abstract Changes, such as the introduction of new technology, may have considerable impact on the risk to which a system or organization is exposed. For example, in the oil & gas domain, introduction of technology that allows offshore installations to be operated from onshore means that fewer people are exposed to risk on the installation, but it also introduces new risks and vulnerabilities. We need suitable methods and techniques in order to understand how a change will affect the risk picture. This paper presents an approach that offers specialized support for analysis of risk with respect to change. The approach allows links between elements of the target of analyses and the related parts of the risk model to be explicitly captured, which facilitates tool support for identifying the parts of a risk model that need to be reconsidered when a change is made to the target. Moreover, the approach offers language constructs for capturing the risk picture before and after a change. The approach is demonstrated on a case concerning new software technology to support decision making on petroleum installations.

Key words: Security, risk analysis, change, oil & gas

1 Introduction

Changes, such as introduction of new technology, will often affect the risks to which a system, organization or person is exposed. For example, the introduction of Internet bank services means that someone may steal from our bank account without having to leave home. In the oil & gas domain, the introduction of technology that allows offshore installations to be (at least partly) operated from onshore means that fewer persons are exposed to risk on the installation in case of accidents, but it also introduces new risks and vulnerabilities.

When considering whether to adopt a new solution or choosing between a set of proposed solutions, we clearly need to understand the implications with respect to risk. We therefore need methods and techniques to support risk management that facilitate analysis of such changes. Most of the established methods and techniques for risk assessment provide little or no support for handling change. Typically, they are designed to establish a risk picture that applies at one specific point in time, without providing specific support for analyzing changes to a system. In order to analyze risk with respect to a change, the natural approach is therefore to first analyze the system before the change, and then perform a new analysis, more or less from scratch, for the system after the change. This may be costly and ineffective. In particular, it will be hard for analysts to know what parts of the first analysis will remain valid for the situation after the change.

Focusing on risk modeling and assessment, we present in this paper an approach based on CORAS [15] that is specifically designed to support change management. One important feature of the approach is the introduction of explicit links between parts of the target models describing the system under analysis and related parts of the resulting risk models. This helps the analysts to determine which parts of a risk model that will remain valid after a change to the target of analysis and which parts that will need to be reassessed. Another feature is a graphical risk model notation designed to capture, in the same diagram, the situation before and after the change in an intuitive manner. This facilitates easy comparison between the situations. For a detailed presentation of this approach to managing changing risk using CORAS, including the language and tool support, the reader is referred to our previous work [16,19,20].

The main contribution of this paper is the formal foundation for the risk modeling and assessment approach, including a semantics for risk models that ex-

plicitly capture change, and a calculus to support risk estimation. Unlike most approaches, the rules allow likelihood for threat scenarios and unwanted incidents to be expressed in terms of natural frequency rather than probability, which facilitates better understanding [6]. Our experience also indicates that in a practical risk analysis setting, it is often easier to obtain frequency values than probabilities from system logs, historical data and expert opinion.

The approach takes what we call the *before-after* perspective, where risk analysts are asked to predict the effect on the risk picture of implementing planned changes to the target of analysis. Based on a real case performed in cooperation with industry partners, we demonstrate the approach on a system for managing work permits on an oil & gas installation. The change in question involves the introduction of automated software agents to support human decision makers in order to increase safety through better and more informed decisions. Information security risk is then a major concern, as such risk may in the next instance affect safety risks.

The rest of this paper is organized as follows. In Section 2 we present the work permit system within the oil and gas domain that we will use for demonstration, before introducing the risk modeling and assessment approach in Section 3. An outline of the formal foundation is then given in Section 4. In Section 5 we apply the approach to the system introduced in Section 2. We then present related work in Section 6 before concluding in Section 7. Finally, the details of the formal foundation are presented in the appendices.

2 The petroleum work permit example case

Accidents on oil & gas rigs can have large consequences in terms of loss of life, damage to the environment and economic loss. Non-routine work that takes place on a rig, such as welding or replacement of defect gas detectors, may increase the risk. Therefore, all work except daily routine tasks requires a work permit (WP). This allows decision makers to obtain an overview of all the different types of work that is planned and ongoing on all locations on the rig at all times, to oversee all extra safety measures related to the work, and to reject or stop work if necessary. Every 12th hour, a WP meeting is held on the rig to decide which work permits to release for the next shift. When deciding whether to release (accept) or reject a WP, the decision makers need to take a number of safety considerations into account, including potential conflicts or interference with other work, the current state of safety barriers, and the weather. This is very challenging, as the number of applications can be very high, meaning that only a few minutes or even less are available for each decision.

In the following we assume that a petroleum operator (henceforth referred to as Operator) has initiated a project in collaboration with a software tool and service provider (henceforth referred to as Provider) to update their ICT system for work permit management. The current solution offers functionality for registering a new WP application and to release or reject an application, but provides very little support for the actual decision making. The Operator wishes to introduce decision support in the form of an automated smart agent that collects relevant information for each WP application and provides advice to the human decision makers. The advice will be either a warning that the agent has detected something that might indicate that the WP should be rejected, accompanied by an explanation, or simply an empty message. Human decision makers will still be fully responsible for the final decision.

A couple of different architectures and solutions are being considered. Before selecting one of them, it has been decided to perform a risk analysis of each alternative, using the current system as a common baseline. We now first present UML [18] models of the current system that will serve as the baseline, before presenting one of the alternatives. These models will represent the target systems to be analyzed later.

2.1 Current solution

Figure 1 shows the communicating entities involved in the current system. The class RigSystem represents all ICT infrastructure related to WPs that are installed on the rig itself. WeatherService is an internet-based meteorological service offering weather forecasts. The small boxes on the borders represent communication ports. The port rs_ui represents the user interface. All other ports represent technical interfaces.

The internal details of RigSystem are shown in Figure 2. We have not assigned names to the internal communication ports. WPManager handles WP applications and release/reject decisions. All communication with users goes through WPManager, which also includes a screen showing weather data and forecasts that are continuously updated from WeatherService. DeviationsDB is a database where deviations related to the state, maintenance, testing etc. of equipment on the rig are recorded.

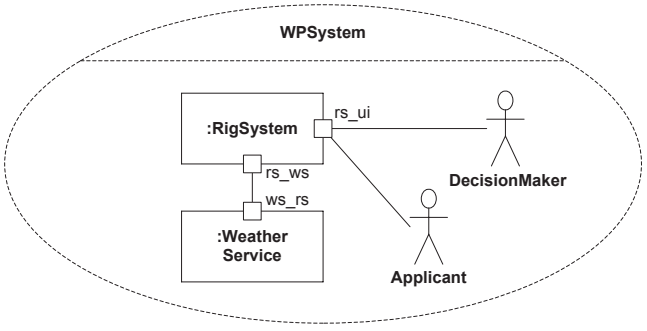

Figure 1. Current solution, overall view

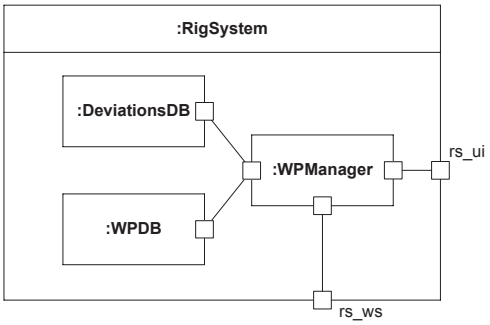

Figure 2. Current solution, internal details of RigSystem

For example, this includes information about any faults that have been detected, as well as tests and maintenance that have not been carried out. WPDB is a database that stores all WPs and related information, such as the location where the work takes place, who does the work, when it starts and stops, what type of equipment will be used, and so on. WPManager includes a user interface for querying DeviationsDB and WPDB, as the DecisionMaker might want to obtain information from these databases before deciding whether to release or reject a new WP.

An overall view of the WP application process is shown in Figure 3. Note that the update of weather data from WeatherService to RigSystem/WPManager is a continuous process that is independent from the WP application process. It has therefore not been included in the diagram. The process starts with the Applicant registering a new application for a WP, represented by the applyForWP message. This application is then presented to DecisionMaker in the next WP meeting, as illustrated by the presentApplication message. At this point DecisionMaker may optionally decide to retrieve information about other WPs, deviations, and the weather. All this information is stored in WPDB, DeviationsDB and WeatherService, and made available to DecisionMaker through a user interface that is a part of WPManager (and therefore also RigSystem). In Figure 3 this is represented by the reference getAdditionalInfo, which has not been detailed further as its content is of little relevance for our purpose here. Finally, the DecisionMaker may either release or reject the WP, as illustrated by the two operands of the alt operator. We have chosen not to decompose the internal interaction for RigSystem, as all interaction with human users goes via WPManager. The decomposition would therefore basically just replace RigSystem with WPManager.

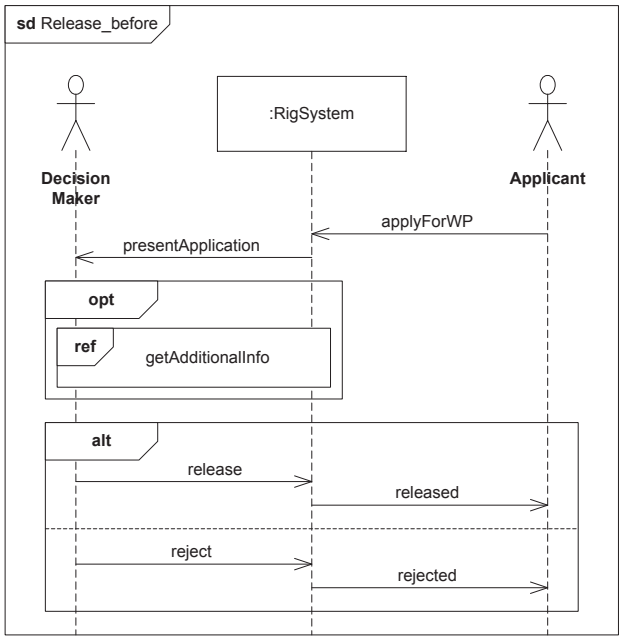

Figure 3. Current solution, overall view of the WP application process

2.2 New solution

The planned change introduces an automated WPAgent that will provide advice to the human decision maker. In the solution considered here, WPAgent will run on a server deployed with the Provider and made accessible on the rig through a web-based interface. This facilitates easy improvement, updating and maintenance of WPAgent by Provider, but also implies more non-local communication. An overview of the new solution is given in Figure 4. The WPAgent will need information from WeatherService. It will also need to interact with the components of RigSystem, which is why communication lines are included between WPAgent and each of these entities.

The internal details of RigSystem are shown in Figure 5. Each of the internal components of RigSystem is available to WPAgent through the port rs_wa on RigSystem.

Figure 6 shows an overall view of the WP application process with the new solution. After the Applicant has registered a new application, this information is forwarded to WPAgent, as represented by the newApplication message. WPAgent then collects the information it needs from the WeatherService and the (internal com-

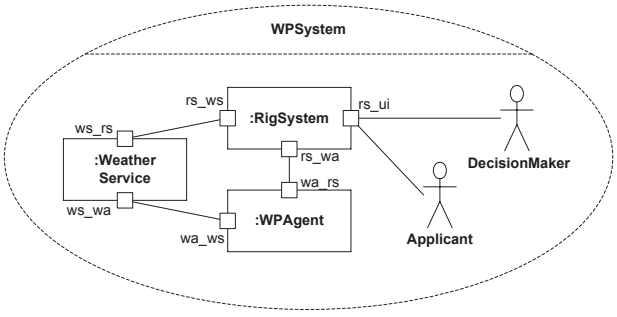

Figure 4. New solution, overall view

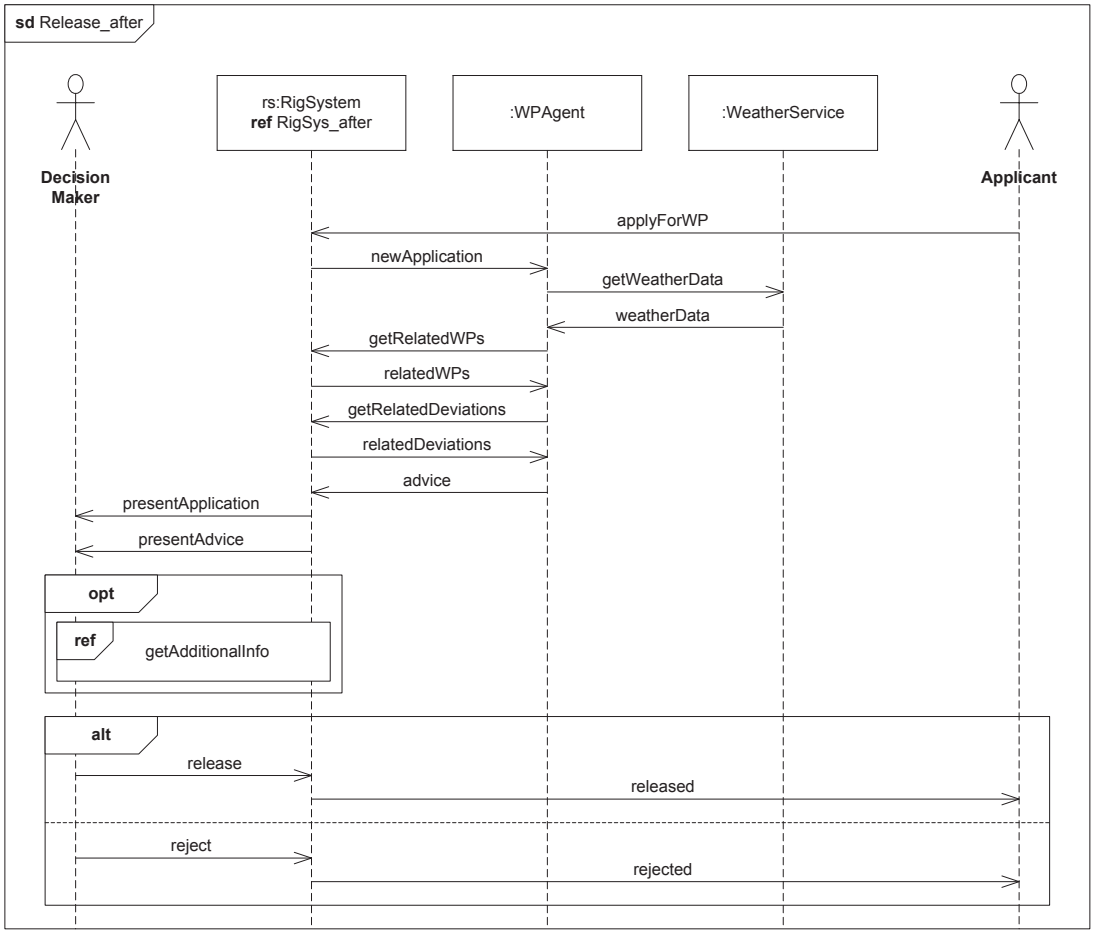

Figure 6. New solution, overall view of the WP application process

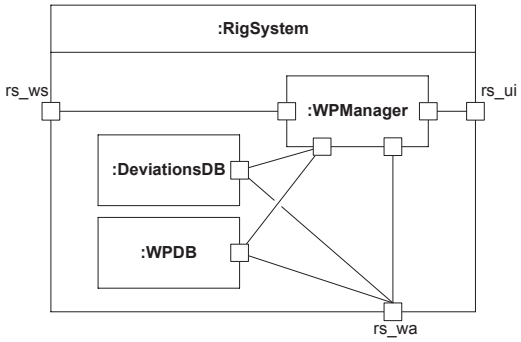

Figure 5. New solution, internal details of RigSystem

ponents of) the RigSystem, as represented by the next six messages going from and to WPAgent. After collecting this information, the WPAgent produces its advice (a purely internal process that is not shown) and sends it to RigSystem, which then presents the application and the advice from WPAgent to DecisionMaker. The rest of the process is identical to the current solution.

Figure 7 shows a decomposition of the RigSystem lifeline of Figure 6. All communication with external components goes to/from WPManager, except the requests from WPAgent to WPDB and DeviationsDB.

3 CORAS analysis of changing risk

This paper is to a large extent example-driven. Hence, the method will mainly be presented and explained at the point in the example where it is used.

Before applying the CORAS approach on the example case, we now give some overall background on CORAS and how CORAS supports risk modeling in the before-after style. This section has to a large extent been extracted from [15, 16].

3.1 CORAS approach

CORAS basically consists of three artefacts, namely a language, a tool and a method. We often refer to the three artefacts together as the CORAS approach.

The CORAS method can be viewed as an instantiation of the ISO 31000 risk management standard [14]. As illustrated by Figure 8, the CORAS method is divided into eight steps. The first four of these correspond to

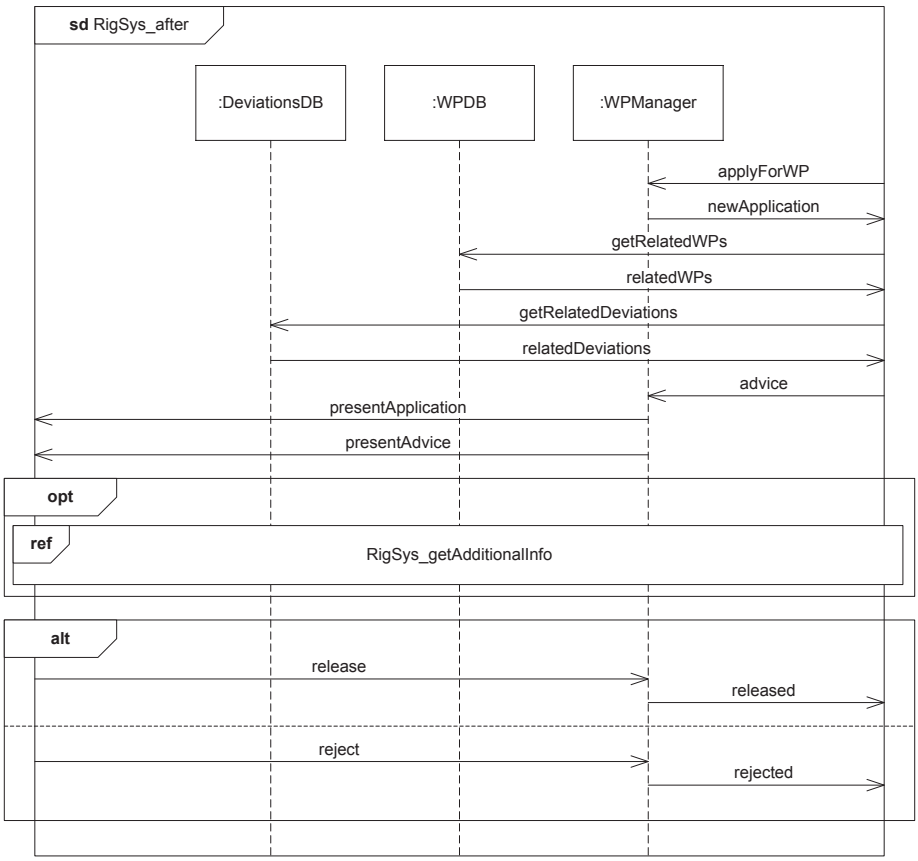

Figure 7. New solution, decomposition of RigSystem

what ISO 31000 refers to as context establishment. They are introductory in the sense that we use them to establish a common understanding of the target of the analysis, and to make the target description that will serve as a basis for the subsequent risk identification. The introductory steps include documenting all assumptions about the environment or setting in which the target is supposed to work, as well as making a complete list of constraints regarding which aspects of the target should receive special attention, which aspects can be ignored, and so forth. The remaining four steps are devoted to the actual detailed analysis. This includes identifying concrete risks and their risk level, as well as identifying and assessing potential treatments for unacceptable risks.

Although the CORAS language has similarities with other approaches to risk documentation, it is nevertheless unique in its support for the whole risk analysis process, from asset identification to risk treatment. It also differs from other approaches in that it has been developed to facilitate communication and interaction during structured brainstorming sessions involving people of heterogeneous backgrounds [7,8,21]. To this end the CORAS language makes use of graphical symbols, or icons, that are closely related to the underlying risk analysis concepts, and that are intended to be easily comprehensible. It also offers a schematic procedure allowing the translation of any fragment of a CORAS diagram into a paragraph in English.

Another unique feature of CORAS is the tight interweaving of the CORAS language into everything that takes place during the risk analysis process. The CORAS method offers detailed guidelines for how to fulfil the goals of the various steps and tasks in general, and how to fulfil these goals making efficient use of the CORAS language in particular.

The core part of the language is referred to as the basic CORAS language. It offers five different kinds of diagrams, namely *asset diagrams*, *threat diagrams*, *risk diagrams*, *treatment diagrams*, and *treatment overview diagrams*, each supporting a particular stage in the risk analysis process. These different diagrams are actually overlapping. The distinction between them is more of a pragmatic nature, and we have given them different names because they are used in different parts of the analysis and for different purposes; asset diagrams are used in asset identification, threat diagrams are used in risk identification and risk estimation, risk diagrams are used in risk evaluation, and treatment diagrams and treatment overview diagrams are used in treatment identification. This means that asset diagrams become the starting point for threat diagrams, threat diagrams become the starting point for risk diagrams, and so forth.

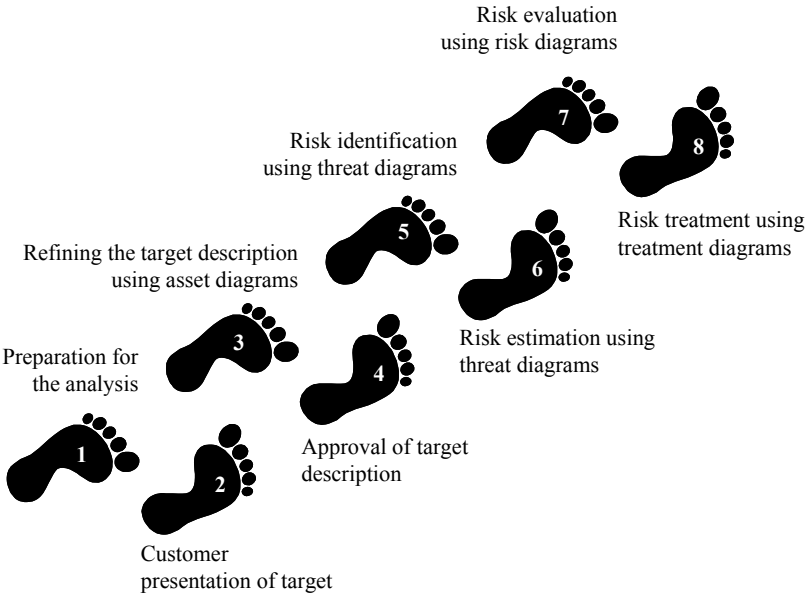

Figure 8. The eight steps of the CORAS method

3.2 Using CORAS for before-after risk modeling

In a CORAS risk analysis, diagrams are used intensively to facilitate risk identification and risk estimation. The diagrams are furthermore used as a part of the documentation and reporting of the analysis results. Figure 9 depicts an example of a threat diagram for the before-after modeling. Threat diagrams describe how threats may exploit vulnerabilities to initiate threat scenarios, how threat scenarios may lead to unwanted incidents or other threat scenarios, and the assets harmed by the unwanted incidents. The language constructs are deliberate threats (r_2, r_3), accidental threats (r_1), non-human threats (not represented in Figure 9), vulnerabilities (u_1), threat scenarios $(v_1 - v_6)$, unwanted incidents (v_7), assets (a) and target references $(t_1 - t_3)$. Only threat scenarios and unwanted incidents may be assigned likelihoods.

Solid lines, like on the vertex v_1 and the relation from v_1 to v_3, indicate elements that only exists in the *before* state, while dashed lines indicate elements that exist *after*. The vertices with a white shadow, like v_2, are those that exist both *before* and *after*, while those with black shadows, like v_5, exist only *after*. The dashed relations with a single likelihood set, like the relation from v_5 to v_6, exist only *after*, while those with double likelihood sets, like the relation from v_3 to v_4, exist both *before* and *after*.

There are furthermore three kinds of relations in threat diagrams, namely initiates relations, leads-to relations and impacts relations. An initiates relation has a threat as source and a threat scenario or unwanted incident as target. It can be annotated with a likelihood that describes the likelihood for the threat to initiate the related scenario or incident. A leads-to relation has a threat scenario or unwanted incident as both source and target. It can be annotated with a conditional likelihood. An impacts relation has an unwanted incident as source and an asset as target, and can be annotated with a consequence value that describes the harm of the incident on the asset when the incident occurs.

We use target references to link elements of the threat diagram to specific aspects of the target. For example, the target reference t_1 may be understood as a pointer to the aspect of the target of relevance for the threat scenario v_1.

The tool [19,20] we have developed for our approach supports the modeling of these kinds of CORAS diagrams. It comes with features for preventing or detecting syntactical errors, and for specifying the traceability links from the CORAS models to the target models. The target models may be UML or some other kind of notation. The traceability links correspond to the target references, and the tool has functionality for automatically flagging CORAS diagrams that may need to be reassessed due to changes in the target. The user can moreover select between different views in order to display only the *before* or only the *after* model elements.

4 Outline of semantic foundation

A CORAS diagram in the before-after style may be thought of as a two-state CORAS diagram. It describes the risk picture in the before state as well as in the after state. A CORAS diagram may also focus on only one state, for example the current state. Such a diagram may be thought of as a one-state CORAS diagram. The semantic foundation for one-state diagrams presented below is based on [22].

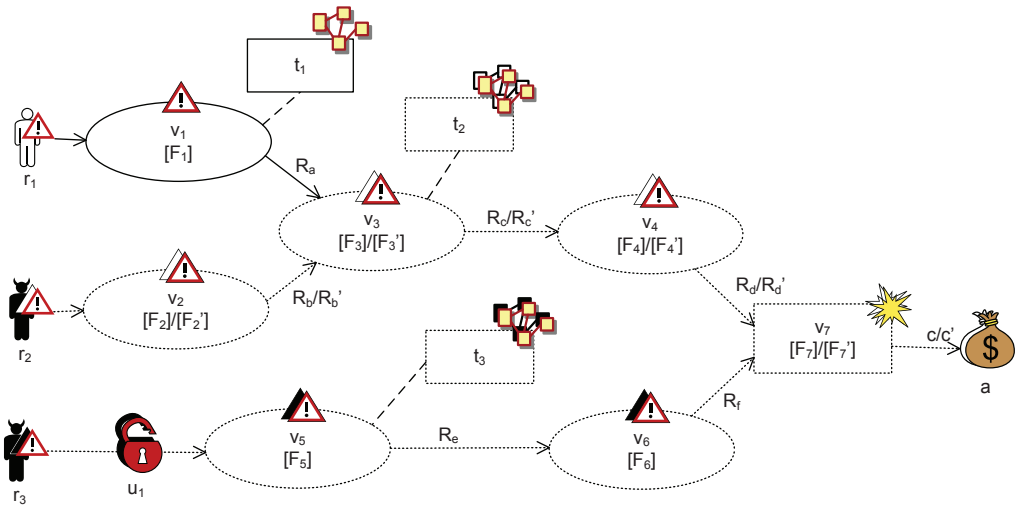

Figure 9. Instantiation of risk graphs in CORAS

4.1 One-state CORAS diagrams as risk graphs

A one-state CORAS diagram may be understood as a risk graph. A risk graph consists of vertices representing threat scenarios (or unwanted incidents) and a finite set of directed relations representing leads-to relationships between them. An example risk graph is shown in Figure 10. Each vertex in a risk graph is assigned a set of frequencies representing the estimated likelihood for the scenario to occur. The assignment of several frequencies, typically a frequency interval, represents underspecification of the frequency estimate. A relation from vertex v to vertex v' means that v may lead to v'. Also the relations can be assigned likelihood sets. These are conditional ratios that specify the likelihood for a scenario leading to another scenario when the former occurs. One threat scenario may lead to several other threat scenarios that are not necessarily mutually exclusive. Moreover, one occurrence of a given threat scenario may lead to several occurrences of another threat scenario. For example, one occurrence of the threat scenario *Misconfiguration of company firewall* may lead to several instances of *An employee's computer is infected by malware*. Hence, a conditional ratio assigned to a relation may be higher than 1, and the conditional ratios on the relations leading from a threat scenario may clearly add up to more than 1.

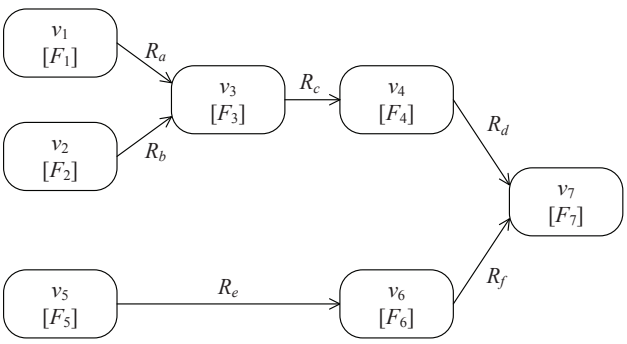

Figure 10. Risk graph

There may of course also be cases where not *all* occurrences of a given threat scenario leads to another. Furthermore, a risk graph is allowed to be incomplete in the sense that there may be threat scenarios not accounted for in the risk graph that can result from or lead to those included in the graph. Hence, the conditional ratios of the relations leading from a threat scenario may also add up to less than 1.

A risk graph may be represented by a set of statements G which contains exactly one statement of the form

$$v \textit{ occurs with a frequency in } F$$

for each vertex $v(F)$, and exactly one statement of the form

$$v_1 \textit{ leads to } v_2 \textit{ with conditional ratio in } R$$

for each relation $v_1 \xrightarrow{R} v_2$. A set of histories (traces of events) H fulfills a risk graph G, written

$$H \vdash G,$$

if every history of H fulfills every proposition of G.

The set of histories represents the behavior of the world (the target of the analysis) by specifying the sequence of occurring events, as well as the time of occurrence of each event. Semantically a risk graph is a set of statements about the frequency of events to occur in H. This means that each statement in G is either true or false given the history H.

4.2 Calculus of risk graph formulas

We now present two examples of rules for reasoning about likelihoods of risk graphs elements. The rules can be used to calculate the likelihood of a vertex based on existing estimates for other vertices in a risk graph, or it can be used for consistency checking of likelihood estimates. For the full calculus and their soundness proofs, we refer the reader to [22].

The following new vertex expressions are introduced here. The expression $v_1 \sqcap\!| \; v_2$ denotes the subset of v_2 corresponding to the instances of v_2 resulting from v_1. For example, if v_1 is the scenario *Misconfiguration of company firewall* and v_2 is the scenario *An employee's computer is infected by malware*, $v_1 \sqcap\!| \; v_2$ denotes the subset of the latter that are preceded by the former. Hence $v_1 \sqcap\!| \; v_2(f)$ states that the frequency of these instances is f. The expression $v_1 \sqcup v_2$ denotes the vertex we get by combining v_1 and v_2. For example, if v_1 is the scenario *The customer database is infected by malware* and v_2 is the scenario *An employee's computer is infected by malware*, $v_1 \sqcup v_2$ denotes the scenario that the computer or the database is infected by malware. Hence $v_1 \sqcup v_2(f)$ states that the frequency of this combined vertex is f. Finally, for any vertex expression v, $s(v)$ yields the set of events associated with v. A formal semantics as well as soundness proofs for the following rules are provided in Appendix B.

4.2.1 Rule for leads-to

This rule applies to the leads-to relations $v_1 \xrightarrow{r} v_2$ and makes use of the frequency f of vertex v_1 and the conditional ratio r of the leads-to relation to calculate the frequency of v_2. There are therefore two premises in this rule. The one to the left states that the frequency of v_1 in any history in H is f. The other states that for any history in H, each occurrence of v_1 leads to an occurrence of v_2 with conditional ratio r. Hence, the frequency of v_2 in any history in H resulting from occurrences of v_1 is $f \cdot r$.

$$\frac{H \vdash v_1(f) \quad H \vdash v_1 \xrightarrow{r} v_2}{H \vdash v_1 \sqcap\!| \; v_2(f \cdot r)}$$

4.2.2 Rule for separate vertices

This rule is for calculating the frequency that results from combining two vertices. The correct aggregated frequency depends of course on how the two scenarios are related. Such relations may, for example, be that they are statistically dependent or that they have events in common. The rule applies to two vertices that are separate, which means that they have no events in common. There are therefore three premises in this rule. The two to the left state that for any history in H the frequency of v_1 is f_1 and the frequency of v_2 is f_2, respectively. Since the premise to the right states that v_1 and v_2 are separate it follows that the frequency of v_1 and v_2 jointly in any history in H is $f_1 + f_2$.

$$\frac{H \vdash v_1(f_1) \quad H \vdash v_2(f_2) \quad s(v_1) \cap s(v_2) = \varnothing}{H \vdash v_1 \sqcup v_2(f_1 + f_2)}$$

4.3 Two-state CORAS diagrams as pairs of risk graphs

A two-state CORAS diagram may be understood as a pair of risk graphs, G_B, G_A, representing the risk picture respectively before and after the system change. The two risk graphs are sets of statements about histories H_B and H_A, respectively. H_B are the histories before the system change, while H_A are the histories after the system change.

A pair of sets of histories (H_B, H_A) fulfills a pair of risk graphs (G_B, G_A), written

$$(H_B, H_A) \vdash (G_B, G_A)$$

if

$$H_B \vdash G_B \wedge H_A \vdash G_A$$

5 Applying the approach on the petroleum example case

We now demonstrate the application of the approach on the case from Section 2. First we show CORAS diagrams in the before-after style and then we show how the calculus presented in Section 4.2 is used to support reasoning about likelihoods.

5.1 Establishing CORAS threat diagrams in the before-after style

Figure 11 shows the assets before as well as after the change, i.e. before and after the introduction of WPAgent. The stakeholder for this risk analysis is the Operator. Two direct assets apply both before and after the change. *Availability of WP data* refers to the WP data stored in WPDB and accessed through WPManager, while *Availability of weather data* refers to the weather data obtained from WeatherService and also accessed through WPManager. Two new direct assets are identified that are only of relevance after the change: *Availability of WP advice* and *Integrity of WP advice*. Both refer to the advice produced by the newly introduced WPAgent.

In addition to the direct assets, Figure 11 contains three indirect assets, represented by the white asset symbols. Indirect assets are harmed only through harm to other assets. All direct assets may affect the indirect asset *Quality of WP decisions*, which refers to the final decisions w.r.t. work permits made by the human DecisionMaker. Bad decisions may in the worst case lead

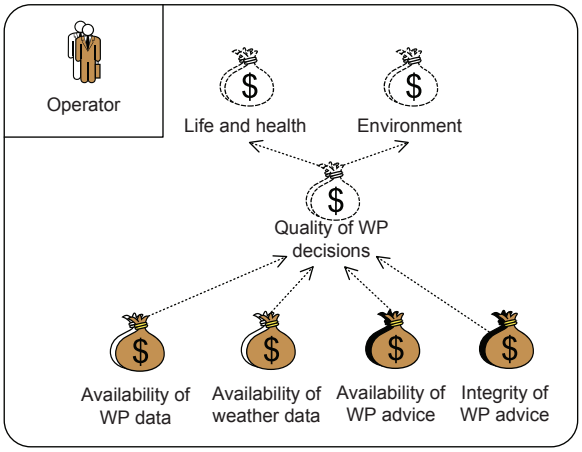

Figure 11. Asset diagram showing assets before and after the introduction of the WPAgent

to loss of life and environmental damage. Therefore, the *Quality of WP decisions* asset affects the indirect assets *Life and health* and *Environment*, which represent the overriding safety concerns for the Operator.

Figure 12 and Figure 13 show the threat diagrams in the before-after style, i.e. they capture the situation before as well as after the change. In this particular case, all the threat diagram elements that apply before the change remain valid after the change. We therefore decided to include all the elements applying both before and after the change in Figure 12, while Figure 13 contains the elements that are relevant only after the change. Note, however, that it is up to the analysts to decide how to split up diagrams in order to keep them at a manageable size. This may be done based on assets, threats or any other criteria deemed suitable for each case.

The upper part of Figure 12 shows that the threat *External communication link* may initiate the threat scenario *Connection with WeatherService goes down* via the vulnerability *No communication redundancy*, which may again lead to the unwanted incident *Weather data cannot be accessed from WPManager*, which impacts the asset *Availability of weather data*. The threat is associated with the communication line between WeatherService and RigSystem in the target model.

Further down in the diagram, the threat *Software error* may initiate the threat scenario *WeatherService goes down*, leading to the same unwanted incident. Here, the threat scenario *WeatherService goes down* is associated with the target model element WeatherService. The dotted lines combined with the white shadow under the elements of Figure 13 show that all the elements of Figure 12 apply both before and after the change. The rest of the diagram should now be self-explanatory.

The upper part of Figure 13 shows that the threat *Hacker* may initiate the threat scenario *Communication link hacked* exploiting the vulnerability *Internet communication* referring to the communication link between WPAgent and RigSystem in the target model. This threat scenario may then lead to two unwanted incidents, the first of which is *Erroneous WP advice submitted to WPManager*, which impacts the asset *Integrity of WP advice*. The dotted lines combined with the black shadow under each of these model elements show that they apply only after the change. Again we do not explain the rest of the diagram.

Note that in this particular example case, the change to the target system, i.e. the introduction of WPAgent, results in new risks being introduced while all the original risks remain equally valid. In general, however, a change can result in risks becoming obsolete or that existing risks become more or less severe. The modeling and analysis of all such risk changes are supported by our approach. The interested reader is referred to our previous work for further examples [16,19,20]. At the same time, CORAS is an asset-driven, defensive approach to risk analysis. This means that the risk analysis is concerned with the protection of the things of value that is already there, and not with balancing opportunities and potential gain against the identified risks. Of course, when considering the full picture we would also need to take into account the positive consequences of the change, such as increased quality of decisions which may result from obtaining automated WP advice.

5.2 Application of the risk graph calculus to support consistency checking

In this section we show how the risk graph calculus can be employed to check consistency of likelihood estimates in a CORAS threat diagram. By checking consistency we ensure that the likelihood assignments are meaningful given the semantics of CORAS diagrams, and thereby increase the confidence that the diagram correctly reflects reality.

Figure 14 shows an extract of Figure 13 where frequency estimates have been assigned to the threat scenarios and the unwanted incident, while ratio estimates have been assigned to the leads-to relations. All likelihood values have been given as intervals, as it is often very hard to obtain exact values in practice. In addition, a qualitative consequence level *High* have been assigned to the impacts-relation from the unwanted incident *WPAgent unable to generate WP advice* to the asset *Availability of WP advice*. The risk level is determined by the likelihood of the incident as well as its consequence, and likelihood assessment is therefore a central part of risk estimation. Although we are primarily interested in the likelihood of the unwanted incident, assigning frequency and ratio estimates also to the threat scenarios and relations for paths leading up to the incident will help analysts in establishing the value of interest and/or increase their confidence that the value is correct.

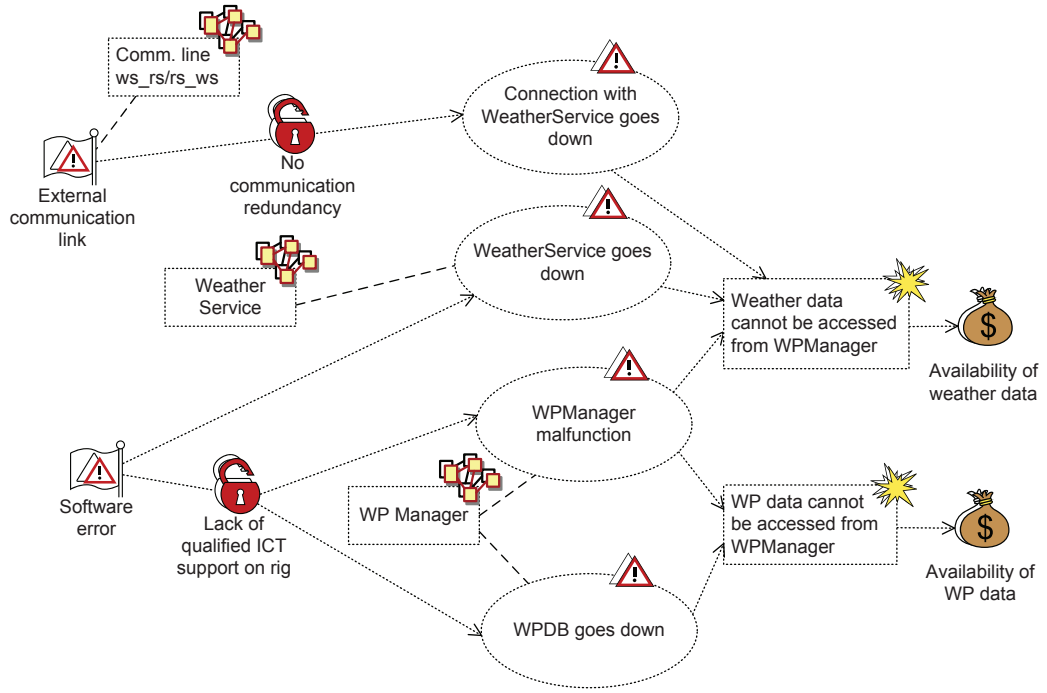

Figure 12. Threat diagram before and after the introduction of the WPAgent, part 1

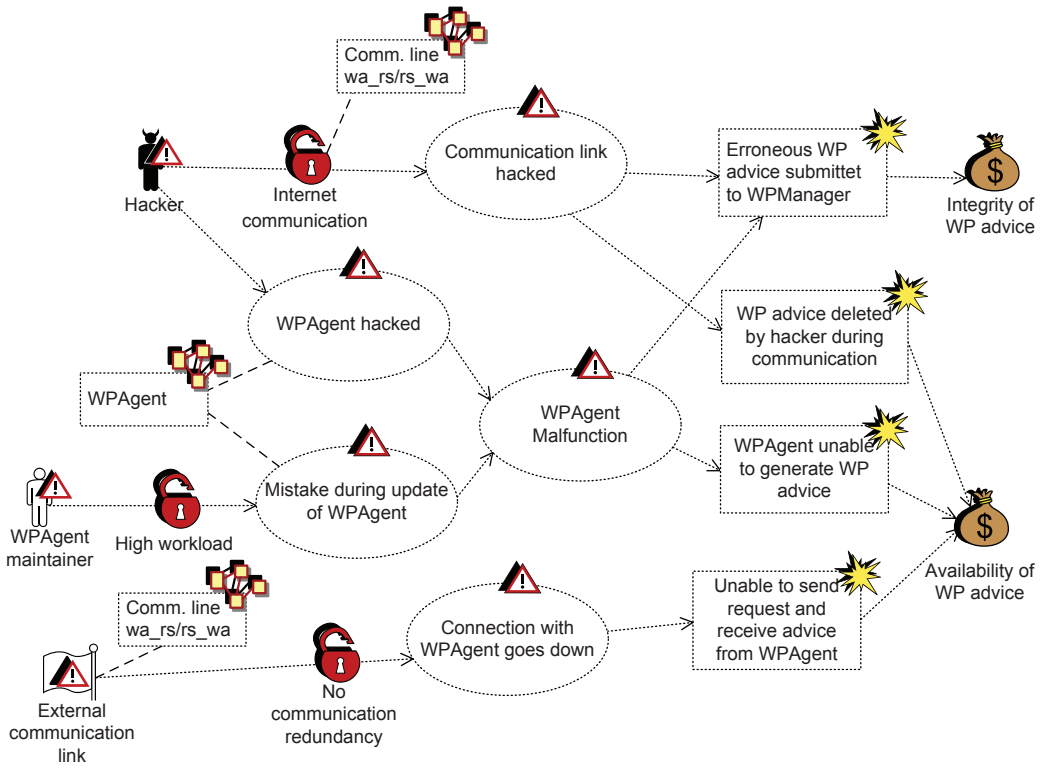

Figure 13. Threat diagram before and after the introduction of the WPAgent, part 2

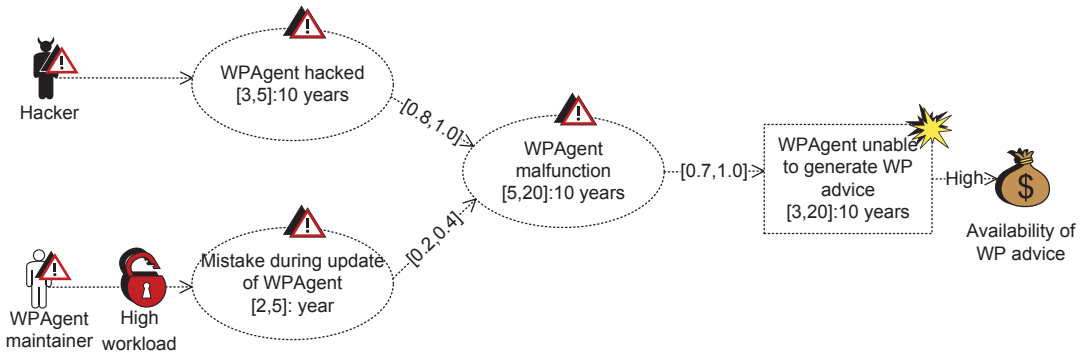

Figure 14. Extract of threat diagram with likelihood estimates

In the following we assume that a group of analysts have arrived at the frequency and conditional ratio assessments of Figure 14 during a risk analysis meeting. Our task is to check if the estimates are mutually consistent. We assume that the diagram is complete in the sense that there are no other paths leading to the unwanted incident than the ones shown that need to be taken into consideration.

Before applying the calculus we need to translate the CORAS threat diagram into a risk graph. Vulnerabilities are mere annotations on relations, and can be ignored in the formal representation of the diagrams. Other elements of threat diagrams can be translated to risk graph elements as follows:

- A threat scenario or unwanted incident of a threat diagram is replaced with a vertex of a risk graph.
- A leads-to relation of a threat diagram is preserved in the risk graph.
- A set of threats $t_1, \ldots, t_n$ with initiates relations to the same threat scenario s is interpreted as follows: The threat scenario s is decomposed into n parts, where each resulting sub-scenario $s_j, j \in \{1, \ldots, n\}$, corresponds to the part of s that is initiated by threat t_j. The threat t_j with initiates relation of likelihood l_j to sub-scenario s_j is then combined into the risk graph scenario *Threat* t_j *initiates* s_j and the scenario is assigned likelihood l_j.
- An impacts relation from unwanted incident u to asset a with consequence c in a threat diagram is interpreted as follows: The impacts relation is interpreted as a risk graph relation with likelihood 1; the asset a is interpreted as the risk graph scenario *Incident* u *harms asset* a *with consequence* c.

In Figure 14 we see that no likelihood assignments have been added to the initiates-relations from the threats to the threat scenarios. The threats and the initiates relations are therefore not relevant for checking consistency of likelihood assessments for this diagram. Moreover, impacts relations and assets do not carry information about likelihoods, and can therefore always be ignored w.r.t. likelihood calculations. Hence, for our purpose, we only need to preserve the threat scenarios, unwanted incidents and leads-to relations of Figure 14 when translating the diagram. This gives us the risk graph in Figure 15. In order to avoid having to repeat the full node names, we have used the following names: v_1=*WPAgent hacked*, v_2=*Mistake during update of WPAgent*, v_3=*WPAgent malfunction*, v_4=*WPAgent unable to generate WP advice*. We have also used 10 years as a common frequency period for all nodes. In the following all frequency values will be given per 10 years, without stating this explicitly.

We will check consistency of the diagram by comparing the values assigned to the nodes v_3 and v_4 with the values we obtain by applying the rules on the paths leading to these nodes. As the likelihood estimates are given as intervals we need to define multiplication and addition of intervals:

$$[min_1, max_1] \ op \ [min_2, max_2] = \\ [min_1 \ op \ min_2, max_1 \ op \ max_2]$$

where $op \in \{\cdot, +\}$.

We start by calculating the contributions from v_1 and v_2 to v_3 via the leads-to relations. Applying Rule 4.2.1 we deduce $v_1 \sqcap\!| \ v_3([3,5] \cdot [0.8,1.0])$ from $v_1([3,5])$ and $v_1 \xrightarrow{[0.8,1]} v_3$. Hence, we get $v_1 \sqcap\!| \ v_3([2.4,5])$. Intuitively, this means that v_3 (i.e. *WPAgent malfunction*) results from v_1 (i.e. *WPAgent hacked*) from 2.4 to 5 times per 10 years. Similarly, applying Rule 4.2.1 on $v_2([20,50])$ and $v_2 \xrightarrow{[0.2,0.4]} v_3$ we get $v_2 \sqcap\!| \ v_3([4,20])$.

In the next step we want to calculate the frequency of v_3 from the above contributions. We assume that all occurrences of v_3 are due to either v_1 or v_2, but not both. This means that v_3 can be divided into two separate vertices v_{3a} and v_{3b} such that $v_3 = v_{3a} \sqcup v_{3b}$ and

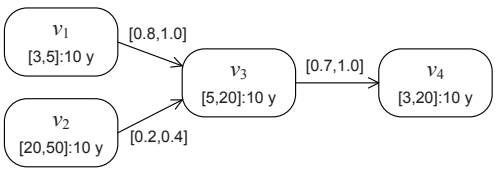

Figure 15. Risk graph representation of Figure 14 for likelihood calculations

$s(v_{3a}) \cap s(v_{3b}) = \varnothing$, where v_{3a} represents the malfunctions due to v_1 and v_{3b} represents the malfunctions due to v_2, as illustrated in Figure 16. The frequency of v_{3a} therefore equals that of $v_1 \sqcap\!| \, v_3$, and the frequency of v_{3b} equals that of $v_2 \sqcap\!| \, v_3$. As $s(v_{3a}) \cap s(v_{3b}) = \varnothing$, we may now apply Rule 4.2.2 to deduce $v_{3a} \sqcup v_{3b}([6.4, 25])$ and therefore $v_3([6.4, 25])$.

In this case we see that the frequency interval for v_3 that we arrived at through the calculation is quite well in line with the original estimate from Figure 14, as there is a large degree of overlap between the two intervals. In case of significant inconsistencies, members of the analysis team will need to resolve the issue through further discussion, data/information collection and analysis.

The next step is to calculate the frequency interval for v_4. For the purpose of this demonstration, we continue to use the calculated interval $[6.4, 25]$ for v_3. Applying Rule 4.2.1 on the relation from v_3 to v_4 we get $v_4([4.48, 25])$, which again corresponds fairly well with the interval originally assigned in Figure 14. This concludes the demonstration of the application of the calculus.

All relevant model elements from Figure 14 concern only the *after* perspective, which means that only one likelihood interval is assigned to each vertex and relation. In the general case we may have a different interval for the *before* and *after* perspective. This can be straightforwardly dealt with by applying the calculus rules on each perspective separately.

6 Related work

Our approach to the management of changing and evolving risk is based on the ISO 31000 standard [14], and consists of a method, a language and tool support. Some of the important features of the approach are the systematic traceability of changes from the target models to the risk models, the support for modeling and visualizing risk changes, as well as the underlying formalism and the techniques for reasoning about likelihoods. Some other works have similar features, but—to the best of our knowledge—none of these provide support for managing change throughout the whole risk analysis process.

Model Versioning and Evolution (MoVE) [4, 17] is an approach to build an infrastructure to maintain the validity, mutual consistency and interdependencies between models as they evolve over time within model-driven engineering. The approach does not target security and risk in particular, but rather builds a tool-supported infrastructure for versioning of several inter-dependent models, for example for software architecture and design, business processes, services, security and risk.

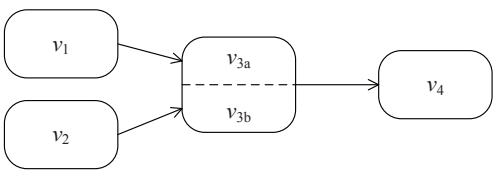

Figure 16. Splitting v_3 in two. Frequencies have been omitted to reduce clutter in the diagram

In [11] a model-based approach to risk analysis with support for dependency identification and modeling is proposed. The approach relates risk elements to elements of a functional model of the target of analysis. Moreover, the model elements are related to security objectives and security requirements, and risks are related to threats and security controls. Although risk assessment is supported, there is no risk modeling support other than a simple, high-level description of incidents and their likelihood and consequence.

Thales Research & Technology has developed their own industrial model-based approach to risk assessment, supported by the Rinforzando tool [3]. The security risk assessment and modeling can be performed as stand-alone, but is also designed to serve as an integrated part of their mainstream system engineering workbench [23]. For this purpose, dynamic links can be built and maintained between the risk models and the system engineering models, the latter specified using a service oriented architecture (SOA) modeling suit. When any model changes are implemented during the system development process, either on the risk model or the system model, the changes are immediately propagated via the links to trigger updates and maintain the mutual consistency between the modeling domains. The approach is similar to our in the sense of using traceability links between the risk model and the system model to maintain validity and mutual consistency. However, the integration with their system engineering process is hard-coded and much tighter than what is offered by CORAS. This is at the cost of general applicability, though, as Rinforzando is tied to the Thales engineering workbench, whereas CORAS allows any notation to be used for target modeling.

Considering modeling support, some of the established techniques for risk and threat modeling facilitate automatic updating of the values that are annotated on the diagrams; by changing input values to capture changes in the target of analysis, the derived output values can be generated. These techniques include fault trees [12], Markov models [9, 13], and Bayesian networks [2]. Influence diagrams [10] were originally a graphical language designed to support decision making by specifying the factors influencing a decision. In [5], such diagrams are connected to the leaf nodes of fault trees supporting the propagation of influence to the unwanted incidents specified at the root of the tree. Similar, but simpler, are the risk influence diagrams [1] where influencing factors are connected to the nodes in event trees. Although these techniques could serve as alternatives to

our use of CORAS for handing traceability, they do not come with methodological support for how to manage change and dependencies in the overall risk management process, and they do not provide support for explicitly modeling and analyzing the risk changes.

7 Conclusion

Change due to technological progress, quickly evolving markets, or other factors, is an important characteristic of society today. Changes to systems and organizations typically imply changes also in the risks to which the systems and organizations are exposed. Understanding these changes is crucial for those who need to make decisions. We have presented an approach to analyze risk with respect to change. The approach provides specialized support for establishing explicit links between elements of the target of analysis and the related parts of the risk model, as well as for explicitly capturing the changes in the risk picture. This facilitates a deeper understanding of the relation between the target of analysis and the risks w.r.t. change and makes it easier to decide which part of a risk analysis needs to be reconsidered when a new change is introduced.

The approach is underpinned by a formal calculus for likelihood estimation that is proved to be sound. Likelihoods for threat scenarios and unwanted incidents are given in terms of natural frequency. The use of frequency, rather than probability, ensures that the calculus is easily applicable in a practical setting, where our experience indicates that it is much easier to obtain frequency estimates than probability estimates from logs, historical data, expert opinion etc. Moreover, the calculus allows the use of intervals rather than exact values, which is again essential in a practical setting where uncertainty is always present.

The approach has been demonstrated on a case involving introduction of new technology to support decision making in a system for handling work permit applications in the oil & gas domain. Information security is a central concern for this system, as incidents related to information security may escalate to safety incidents that may harm life, health and the environment. Avoiding such incidents is of course a major concern for the industry.

There are several interesting directions for further research. One is to develop more comprehensive methodological support, with detailed guidelines for different kinds of usage scenarios and different types of change. Moreover, there is scope for including more rules in the calculus, as well as guidelines for their application. Finally, we would of course like to apply the approach on a wide range of realistic cases from different domains, and to refine it further based on the experiences we would gain.

Acknowledgment. This work has been conducted as a part of the NESSoS network of excellence (256980) funded by the European Commission within the 7th Framework Programme, and the CONCERTO project (333053) funded by the ARTEMIS Joint Undertaking and the Research Council of Norway (232059), as well as the Dynamic Risk Assistant project (217213) funded by the Research Council of Norway.

References

1. Terje Aven, Snorre Sklet, and Jan Erik Vinnem. Barrier and operational risk analysis of hydrocarbon releases (BORA-Release). Part I. Method description. *J. Haz. Mat.*, A137:681–691, 2006.
2. Irad Ben-Gal. Bayesian networks. In Fabrizio Ruggeri, Ron S. Kenett, and Frederick W. Faltin, editors, *Encyclopedia of Statistics in Quality and Reliability*. John Wiley & Sons, 2007.
3. Franco Bergomi, Stéphane Paul, Bjørnar Solhaug, and Raphaël Vignon-Davillier. Beyond traceability: Compared approaches to consistent security risk assessments. In *Proc. Eighth International Conference on Availability, Reliability and Security (ARES'13)*, pages 814–820. IEEE Computer, 2013.
4. Michael Breu, Ruth Breu, and Sarah Löw. MoVEing forward: Towards an architecture and processes for a Living Models infrastructure. *International Journal On Advances in Life Sciences*, 3(1-2):12–22, 2011.
5. EUROCONTROL. *Methodology report for the 2005/2012 integrated risk picutre for Air Traffic Management in Europa*, 2006. EEC Technical/Scientific Report No. 2006-041.
6. Gerd Gigerenzer. *Calculated Risks – How to Know When Numbers Deceive You*. Simon & Schuster, 2002.
7. Ida Hogganvik and Ketil Stølen. Risk analysis terminology for IT-systems: Does it match intuition? In *4th International Symposium on Empirical Software Engineering (ISESE'05)*, pages 13–23. IEEE Computer Society, 2005.
8. Ida Hogganvik and Ketil Stølen. A graphical approach to risk identification, motivated by empirical investigations. In *9th International Conference on Model Driven Engineering Languages and Systems (MoDELS'06)*, volume 4199 of *LNCS*, pages 574–588. Springer, 2006.
9. Ronald A. Howard. *Dynamic Probabilistic Systems, Volume I: Markov Models*. John Wiley & Sons, 1971.
10. Ronald A. Howard and James E. Matheson. Influence diagrams. *Decis. Anal.*, 2(3):127–143, 2005.
11. Frank Innerhofer-Oberperfler and Ruth Breu. Using an enterprise architecture for IT risk management. In *Information Security South Africa Conference (ISSA'06)*, 2006.
12. International Electrotechnical Commission. *IEC 61025 Fault Tree Analysis (FTA)*, 1990.
13. International Electrotechnical Commission. *IEC 61165 Application of Markov Techniques*, 1995.
14. International Organization for Standardization. *ISO 31000 Risk management – Principles and guidelines*, 2009.

15. Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. *Model-Driven Risk Analysis – The CORAS Approach*. Springer, 2011.
16. Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. Risk analysis of changing and evolving systems using CORAS. In *Foundations of Security Analysis and Design VI (FOSAD VI)*, volume 6858 of *LNCS*, pages 231–274. Springer, 2011.
17. MoVE – Model Versioning and Evolution. `http://move.q-e.at/` [Accessed 27. August, 2014].
18. Object Management Group. *OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.2*, 2009. OMG Document: formal/2009-02-02.
19. Fredrik Seehusen and Bjørnar Solhaug. Tool-supported risk modeling and analysis of evolving critical infrastructures. In *Multidisciplinary Research and Practice for Information Systems*, volume 7465 of *LNCS*, pages 562–577. Springer, 2012.
20. Bjørnar Solhaug and Fredrik Seehusen. Model-driven risk analysis of evolving critical infrastructures. *Journal of Ambient Intelligence and Humanized Computing*, 5(2):187–204, 2014.
21. Bjørnar Solhaug and Ketil Stølen. The CORAS language – Why it is designed the way it is. In *Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures, proc. of 11th International Conference on Structural Safety and Reliability (ICOSSAR'13)*, pages 3155–3162. CRC Press, 2013.
22. Le Minh Sang Tran, Bjørnar Solhaug, and Ketil Stølen. An approach to select cost-effective risk countermeasures exemplified in coras. Technical report A24343, SINTEF ICT, 2013.
23. Jean-Luc Voirin. Method & tools for constrained system architecting. In *18th Annual International Symposium of the International Council on Systems Engineering (INCOSE'08)*, pages 775–789. Curran Associates, Inc., 2008.

A Formal foundation

In the following we introduce the formal machinery.

A.1 Basics

$\mathbb{N}$ and $\mathbb{R}$ denote the sets of natural numbers and real numbers, respectively. We use $\mathbb{N}_0$ to denote the set of natural numbers including 0, while $\mathbb{R}^+$ denotes the set of nonnegative real numbers. This means that:

$$\mathbb{N}_0 \stackrel{\mathsf{def}}{=} \mathbb{N} \cup \{0\}, \quad \mathbb{R}^+ \stackrel{\mathsf{def}}{=} \{r \in \mathbb{R} \mid r \geq 0\}$$

For any set of elements, we use $\mathbb{P}(A)$ to denote the powerset of A.

A tuple is an element of a Cartesian product. We use π_j to extract the jth element of a tuple. Hence, if

$$(a, a') \in A \times A'$$

then $\pi_1.(a, a') = a$ and $\pi_2.(a, a') = a'$.

A.2 Sequences

By A^∞, A^ω and A^* we denote the set of all infinite sequences, the set of all finite and infinite sequences and the set of all finite sequences over some set of elements A, respectively. Hence, we have that

$$A^\omega = A^\infty \cup A^*$$

We define the functions

$$\#\_ \in A^\omega \to \mathbb{N}_0 \cup \{\infty\}, \quad \_[\_] \in A^\omega \times \mathbb{N} \to A$$

to yield the length and the nth element of a sequence. Hence, $\#s$ yields the number of elements in s, and $s[n]$ yields the nth element of s if $n \leq \#s$.

We also need functions for concatenation and filtering:

$$\_ \frown \_ \in A^\omega \times A^\omega \to A^\omega, \quad \_ \circledS \_ \in \mathbb{P}(A) \times A^\omega \to A^\omega$$

Concatenating two sequences implies gluing them together. Hence, $s_1 \frown s_2$ denotes a sequence of length $\#s_1 + \#s_2$ that equals s_1 if s_1 is infinite, and is prefixed by s_1 and suffixed by s_2, otherwise.

The filtering operator is used to filter away elements. $B \circledS s$ denotes the subsequence obtained from s by removing all elements in s that are not in the set B.

A.3 Timed events

$\mathbb{E}$ denotes the set of all events, while the set of all timestamps is defined by

$$\mathbb{T} \stackrel{\mathsf{def}}{=} \mathbb{R}^+$$

A timed event is an element of

$$\mathbb{E} \times \mathbb{T}$$

A.4 Histories

A history is an infinite sequence of timed events that is ordered by time and progresses beyond any finite point in time. Hence, a history is an element of:[1]

$$\begin{aligned}\mathbb{H} \stackrel{\mathsf{def}}{=} \{ \quad & h \in (\mathbb{E} \times \mathbb{T})^\infty \mid \\ & \forall\, n \in \mathbb{N} : \pi_2.h[n] \leq \pi_2.h[n+1] \\ & \forall\, t \in \mathbb{T} : \exists\, n \in \mathbb{N} : \pi_2.h[n] > t \quad \}\end{aligned}$$

The first conjunct requires the timestamp of a timed event to be at least as great as that of its predecessor. The second conjunct makes sure that time will always progress beyond any finite point in time. That is, for any timestamp t and history h there is a timed event in h whose timestamp is greater than t.

We also need a function for truncating histories

$$\_|\_ \in \mathbb{H} \times \mathbb{T} \to (\mathbb{E} \times \mathbb{T})^*$$

The truncation operator captures the prefix of a history up to and including a certain point in time. Hence, $h|_t$ describes the maximal prefix of h whose timed events all have timestamps less than or equal to t.

[1] We often use indentation to represent conjunction.

A.5 Frequencies

As explained above, we use the nonnegative real numbers to represent time. The time unit is equal to 1. For simplicity, we assume that all frequencies are per time unit. The set of frequencies F is therefore defined as follows:

$$\mathbb{F} \stackrel{\mathsf{def}}{=} \mathbb{R}^+$$

Hence, $f \in \mathbb{F}$ denotes the frequency of f occurrences per time unit.

B Risk graphs

B.1 Syntax of risk graph formulas

B.1.1 Risk graph definition

A risk graph G is a pair of two sets (V, R) where

$$V \subseteq \mathbb{P}(\mathbb{E}) \times \mathbb{F}, \quad R \subseteq V \times \mathbb{R}^+ \times V$$

We refer to the elements of V as vertices and to the elements of R as relations. We use $v(f)$ to denote a vertex, while $v \xrightarrow{r} v'$ denotes a relation.

B.1.2 Vertex expressions

The set of vertex expressions is the smallest set X_V such that

$$\mathbb{P}(\mathbb{E}) \subseteq X_V, \quad v, v' \in X_V \Rightarrow v \sqcup v' \in X_V \wedge v \sqcap\!| \, v' \in X_V$$

We need a function

$$s \in X_V \to \mathbb{P}(\mathbb{E})$$

that for any vertex expression yields its set of events. Formally, s is defined recursively as follows:

$$s(v) \stackrel{\mathsf{def}}{=} \begin{cases} v & \text{if } v \in \mathbb{P}(\mathbb{E}) \\ s(v_1) \cup s(v_2) & \text{if } v = v_1 \sqcup v_2 \\ s(v_2) & \text{if } v = v_1 \sqcap\!| \, v_2 \end{cases}$$

B.1.3 Risk graph formulas

A risk graph formula is of one of the following forms

- $H \vdash v(f)$
- $H \vdash v \xrightarrow{r} v'$
- $H \vdash v \sqcup v'(f)$
- $H \vdash v \sqcap\!| \, v'(f)$

where

- $H \in \mathbb{P}(\mathbb{H}) \setminus \varnothing$
- $v, v' \in X_V$
- $f \in \mathbb{F}$
- $r \in \mathbb{R}^+$

B.2 Semantics of risk graph formulas

We use the brackets $[\![\ ]\!]$ to extract the semantics of a risk graph formula. If $v \in \mathbb{P}(\mathbb{E})$ we define:

$$[\![H \vdash v(f)]\!] \stackrel{\mathsf{def}}{=} \forall h \in H : f = \lim_{t\to\infty} \frac{\#((v \times \mathbb{T}) \circledS (h|_t))}{t}$$

The semantics of any other risk graph formula is defined recursively as follows:

$$[\![H \vdash v_1 \sqcup v_2(f)]\!] \stackrel{\mathsf{def}}{=} \exists f_1, f_2, f_3 \in \mathbb{F} : [\![H \vdash v_1(f_1)]\!] \; [\![H \vdash v_2(f_2)]\!] \; [\![H \vdash s(v_1) \cap s(v_2)(f_3)]\!] \; f_1 + f_2 - f_3 \le f \le f_1 + f_2$$

$$[\![H \vdash v_1 \sqcap\!| \, v_2(f)]\!] \stackrel{\mathsf{def}}{=} \exists f_2 \in \mathbb{F} : [\![H \vdash v_2(f_2)]\!] \; f \le f_2$$

$$[\![H \vdash v_1 \xrightarrow{r} v_2]\!] \stackrel{\mathsf{def}}{=} \exists f_1, f_2 \in \mathbb{F} : [\![H \vdash v_1(f_1)]\!] \; [\![H \vdash v_2(f_2)]\!] \; f_2 \ge f_1 \cdot r$$

For a risk graph $G = (V, R)$ the semantics of $H \vdash G$ is defined as follows.

$$[\![H \vdash (V, R)]\!] \stackrel{\mathsf{def}}{=} \bigwedge_{e \in V \cup R} [\![H \vdash e]\!]$$

B.3 Calculus of risk graph formulas – proofs of soundness

We now prove soundness of three example rules. The three rules below correspond to rules 13.10, 13.11 and 13.12 in the CORAS book [15], respectively. There are some minor differences. In the CORAS book the real number decorating a leads-to relation is restricted to $[0, 1]$. The statistical independence constraint in Rule 13.12 of the CORAS book is not needed.

B.3.1 Rule for leads-to

$$\frac{H \vdash v_1(f) \quad H \vdash v_1 \xrightarrow{r} v_2}{H \vdash v_1 \sqcap\!| \, v_2(f \cdot r)}$$

Soundness
Assume

(1) $[\![H \vdash v_1(f)]\!]$
(2) $[\![H \vdash v_1 \xrightarrow{r} v_2]\!]$

Then

(3) $[\![H \vdash (v_1 \sqcap\!| v_2)(f \cdot r)]\!]$

Proof: (2) implies there are $f_1, f_2 \in \mathbb{F}$ such that

(4) $[\![H \vdash v_1(f_1)]\!]$
(5) $[\![H \vdash v_2(f_2)]\!]$
(6) $f_2 \geq f_1 \cdot r$

(1) and (4) imply

(7) $f = f_1$

(6) and (7) imply

(8) $f_2 \geq f \cdot r$

(5) and (8) imply (3).

B.3.2 Rule for mutually exclusive vertices

$$\frac{H_1 \vdash v_1(f) \wedge v_2(0) \quad H_2 \vdash v_2(f) \wedge v_1(0)}{H_1 \cup H_2 \vdash v_1 \sqcup v_2(f)}$$

For simplicity we have merged four premises into two using logical conjunction.[2]

Soundness
Assume

(1) $[\![H_1 \vdash v_1(f) \wedge v_2(0)]\!]$
(2) $[\![H_2 \vdash v_2(f) \wedge v_1(0)]\!]$

Then

(3) $[\![H_1 \cup H_2 \vdash v_1 \sqcup v_2(f)]\!]$

Proof: (1) and (2) imply

(4) $H_1 \cap H_2 = \varnothing \vee f = 0$

(1) and (2) imply

(5) $[\![H_1 \vdash v_1 \sqcup v_2(f)]\!]$
(6) $[\![H_2 \vdash v_1 \sqcup v_2(f)]\!]$

(4), (5) and (6) imply (3).

B.3.3 Rule for separate vertices

$$\frac{H \vdash v_1(f_1) \quad H \vdash v_2(f_2) \quad s(v_1) \cap s(v_2) = \varnothing}{H \vdash v_1 \sqcup v_2(f_1 + f_2)}$$

[2] Hence, $H \vdash X \wedge Y$ means $H \vdash X$ and $H \vdash Y$.

Soundness
Assume

(1) $[\![H \vdash v_1(f_1)]\!]$
(2) $[\![H \vdash v_2(f_2)]\!]$
(3) $s(v_1) \cap s(v_2) = \varnothing$

Then

(4) $[\![H \vdash v_1 \sqcup v_2(f_1 + f_2)]\!]$

Proof: (3) implies

(5) $[\![H \vdash s(v_1) \cap s(v_2)(0)]\!]$

(1), (2), (5) and the fact that $f_1 + f_2 - 0 \leq f_1 + f_2 \leq f_1 + f_2$ imply (4).

C Binary risk graphs

C.1 Syntax of binary risk graph formulas

A binary risk graph formula is of the form

$$(H_1, H_2) \vdash (G_1, G_2)$$

where $H_1 \vdash G_1$ and $H_2 \vdash G_2$ are risk graph formulas. A pair of histories (H_1, H_2) fulfills a pair of risk graphs (G_1, G_2), written

$$(H_1, H_2) \vdash (G_1, G_2)$$

if

$$H_1 \vdash G_1 \wedge H_2 \vdash G_2$$

C.2 Semantics of binary risk graph formulas

$$[\![(H_1, H_2) \vdash (G_1, G_2)]\!] \stackrel{\mathsf{def}}{=} [\![H_1 \vdash G_1]\!] \wedge [\![H_2 \vdash G_2]\!]$$