Divide and Conquer – Towards a Notion of Risk Model Encapsulation

Atle Refsdal[1], Øyvind Rideng[2], Bjørnar Solhaug[1], and Ketil Stølen[1,3]

[1] SINTEF ICT, Norway
[2] Oilfield Technology Group, Norway
[3] Dep. of Informatics, University of Oslo, Norway
{atle.refsdal,bjornar.solhaug,ketil.stolen}@sintef.no
oyvind.rideng@otg.no

Abstract. The criticality of risk management is evident when considering the information society of today, and the emergence of Future Internet technologies such as Cloud services. Information systems and services become ever more complex, heterogeneous, dynamic and interoperable, and many different stakeholders increasingly rely on their availability and protection. Managing risks in such a setting is extremely challenging, and existing methods and techniques are often inadequate. A main difficulty is that the overall risk picture becomes too complex to understand without methodic and systematic techniques for how to decompose a large scale risk analysis into smaller parts. In this chapter we introduce a notion of risk model encapsulation to address this challenge. Encapsulation facilitates compositional risk analysis by hiding internal details of a risk model. This is achieved by defining a risk model interface that contains all and only the information that is needed for composing the individual risk models to derive the overall risk picture. The interface takes into account possible dependencies between the risk models. We outline a method for compositional risk analysis, and demonstrate the approach by using an example on information security from the petroleum industry.

Keywords: Risk analysis, risk modeling, risk model encapsulation, risk composition, security, ICT

1 Introduction

For most organizations, risk management is an indispensable part of the overall management process. Risk management is coordinated activities to direct and control an organization with regard to risk [11], and the objective is to systematically and proactively identify the current risk picture and to ensure that the necessary controls are in place to maintain risks at an acceptable level.

Risk management may be with respect to many different kinds of risk, such as financial, safety, operational, security and environmental damage. In this chapter we focus on (information) security [12]. The criticality of security is particularly evident when considering the information society of today, and the emergence of Future Internet technologies. Information systems and services become

ever more complex, heterogeneous, dynamic and interoperable. Businesses, enterprises, governments, citizens and many other stakeholders rely more and more on the availability of services and information over the Internet, with Cloud services as a prominent example. Managing risks in such a setting is extremely challenging, and established methods and techniques are often inadequate. The main problems are that the overall risk picture becomes too complex to understand, and that the risk picture quickly and continuously changes and evolves.

Risk analysis is a core part of the risk management process, and should be conducted regularly in order to identify, assess and document risks, as well as identifying controls and means to mitigate risks. For most risk analyses only selected parts or aspects of a system or an organization are addressed. This is because it is infeasible or too costly to conduct a full analysis of the whole system or organization at the same time. For such risk analyses addressing selected parts or aspects we can make use of established methods and techniques (e.g. [1, 2, 6, 11, 15, 16, 19]). Such a traditional approach is fine when we can reach an adequate understanding of the risks by analyzing separate parts of the target in isolation. However, for large, complex systems or organizations we may need to consider all parts of the target in combination in order to adequately understand the full risk picture. Taking into account the infeasibility of addressing the full system or organization at once, we need novel techniques for sound and systematic composition of separate risk analyses in order to deduce an overall risk model.

The challenge we address in this chapter is how to facilitate a compositional [18] approach to risk analysis by applying the principle of *encapsulation*. Following a divide-and-conquer strategy we aim for an approach to risk management where separate parts of a system or organization can be analyzed individually. By risk model we mean any representation of risk information, such as threats, vulnerabilities, unwanted incidents and how they are related, as well as estimates of likelihoods and consequences. Compositional techniques should then enable the systematic and sound composition of the individual risk models in order to derive the overall combined result without having to reconsider the details of the individual models.

A compositional approach has several advantages. First, for systems or organizations that are to be analyzed from scratch, a compositional approach allows the analysis to be split-up top-down in manageable chunks in such a way that the details of each individual analysis do not have to be reconsidered when the results of the individual analyses are aggregated back into an overall risk model for the system or organization as a whole. Second, when there already are several risk analyses of different parts or aspects of some system or organization available, a compositional approach enables the overall risk picture to be derived bottom-up without re-analyzing what has already been analyzed. Third, if the target of one individual analysis is reused in another context, also the risk analysis for the target in question should be reusable in the new context. Fourth, when a system changes due to replacement or introduction of new parts, we should be able to deduce the risk level by re-analyzing only the modified parts.

In the example of this chapter we focus mainly on the first of these usage scenarios, namely the top-down one. The three others are however equally important but only partly addressed by the method presented in this chapter.

The contribution of the presented work is an approach to compositional risk analysis that is based on a new notion of risk model encapsulation. By encapsulation we mean that only the elements that are essential for the composition of risk models are externally observable via its interface. As already mentioned, we outline a method for compositional risk analysis from a top-down perspective where a large target is decomposed into sub-targets that are analyzed individually. We introduce techniques for risk model composition that make use of the risk interface of each individual risk model. We demonstrate the approach by using an example drawn from the petroleum industry.

The structure of the chapter is as follows. In Section 2 we present our notion of risk model encapsulation. In Section 3 we present the petroleum industry example that we use to illustrate our approach and techniques. In Section 4 we outline our method for compositional risk analysis, and in Section 5 to Section 7 we present and exemplify the method in more details. In Section 8 we discuss related work before we conclude in Section 9.

2 Risk Model Encapsulation

In this section we introduce and explain our notion of risk model encapsulation. The objective is to allow different risk models to be composed without having to know or understand all the interior details of the individual models. For this purpose we need to define a notion of risk model interface, where the interface contains all and only the information that is needed for risk model composition. Moreover, the resulting combined risk model should possess all the information that is needed for understanding the risk situation of the overall target.

A further challenge that needs to be tackled is how to take into account possible dependencies between the individual risk models. Each sub-target is analyzed separately, and the other sub-targets belong to the environment of the sub-target being analyzed. This means that the other sub-targets can serve as environmental causes of risk that need to be taken into account for the sub-target being analyzed, and that the sub-target in question can be the cause of risks for the sub-targets in its environment.

In the following we introduce our notion of risk model encapsulation by presenting our underlying conceptual model. The concrete modeling support is presented in Section 5 to Section 7.

In the UML [17] class diagram of Figure 1 the term *target* denotes the target of analysis. The goal of the analysis is to build the *risk model* for the target. The target may be decomposed into a number of more fine-grained targets (which we often refer to as sub-targets). There are two crucial features of our approach to risk model encapsulation. First, for each target we need to understand how it relates to its *environment*. Second, we need a precise notion of *interface* which

consists of the risk information that is needed in order to compose the risk model in question with other risk models.

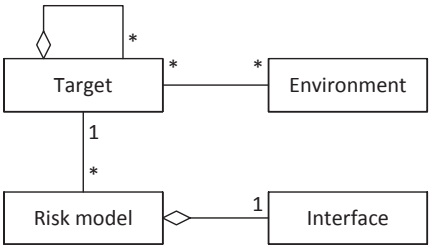

Fig. 1. Risk model

Figure 2 depicts the interface for risk model encapsulation in further detail. The interface consists of three sets of ingredients. The first one is a set of *threat relations* originating *from* the *environment* and impacting the target. These relations represent ways in which the environment may influence the risk model of the target.

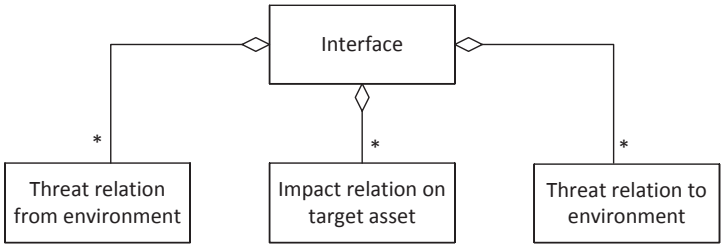

Fig. 2. Interface for risk model encapsulation

The second ingredient is a set of *impact relations* describing potential harm *on target assets*. A target asset is something of value inside the target that must be protected from unacceptable risk. For example, if the target is a database, a target asset could be the integrity of the information on the database. This in contrast with an environment asset that is something of value in the environment. Such an asset could, for example, be the reliability of a web service that uses the database.

The third ingredient is a set of *threat relations* from the target *to* the *environment* that represent ways in which the target may influence the risk model of the environment.

Before demonstrating the application of these concepts we next introduce our example.

3 The Petroleum Work Permit Example

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 is available for each decision.

In the following we assume that a petroleum operator has initiated a project in collaboration with a software tool and service provider to update their ICT system for work permit management. In addition to functionality for registering, releasing and rejecting WP applications, the system will provide 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 or considered extra carefully, accompanied by an explanation, or simply an empty message. Human decision makers will still be fully responsible for the final decision.

The UML collaboration diagram of Figure 3 shows an overall view of the system. The class RigSystem represents all ICT infrastructure related to WPs that are installed on the rig itself. WPAgent represents the automated agent. This will be developed and maintained by the software provider, as represented in Figure 3 by WPAgent maintainer. WeatherService is an Internet-based meteorological service offering weather forecasts. The small boxes on the borders represent communication ports. The port ui on RigSystem represents the user interface of RigSystem, while the port ma on WPAgent represents the interface through which the WPAgent maintainer performs maintenance. All other ports represent technical interfaces.

The WPAgent will need information from WeatherService. It will also need to interact with the components of RigSystem, which explains the lines that are included between WPAgent and each of these entities. The communication between WPAgent and RigSystem goes via an encrypted Internet connection, while the communication with WeatherService uses an open line.

The internal details of RigSystem are shown in the UML internal structure diagram of Figure 4. Each of the internal components of RigSystem is available to WPAgent through the port wa on RigSystem. We have not assigned names to the internal communication ports. WPManager handles WP applications and release/reject decisions. All communication with users goes through WPMan-

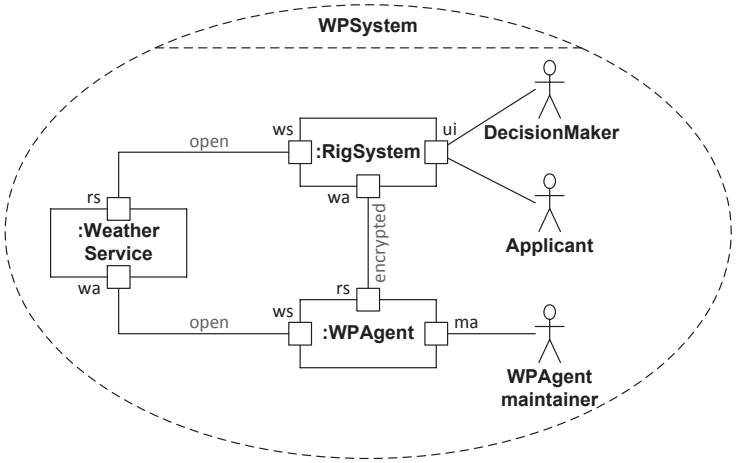

Fig. 3. Overall view of the system

ager, 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. 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 the work 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 Decision-Maker might want to obtain information from these databases before deciding whether to release or reject a new WP.

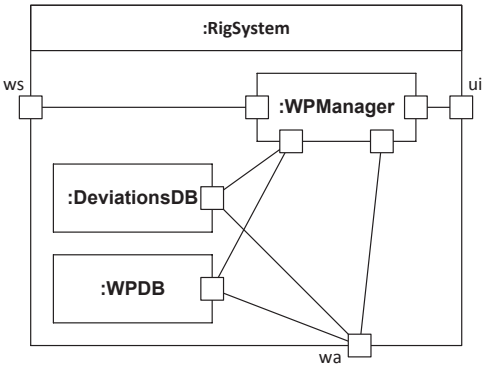

Fig. 4. Internal structure of RigSystem

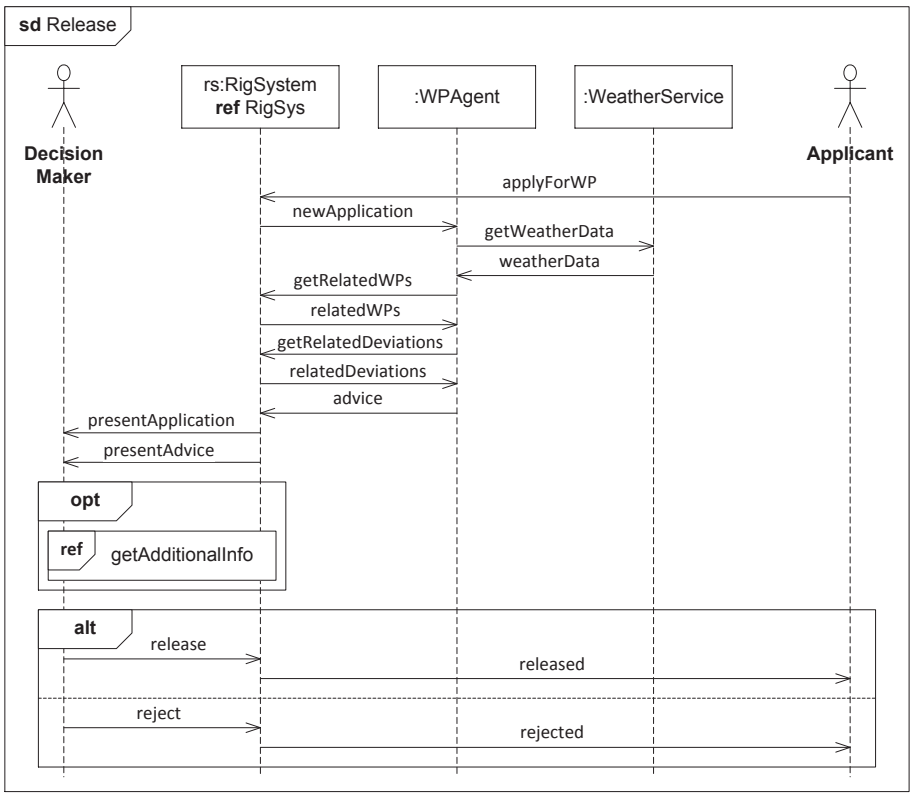

Fig. 5. Message exchange for the WP application process

The WP application process is shown in the UML sequence diagram of Figure 5. Note that the update of weather data from WeatherService to RigSystem/WPManager is a continuous process that is independent from the WP application process and has therefore not been included. The process starts with the Applicant registering a new application for a WP, represented by the applyForWP message. 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 components of) 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. 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 5 this is represented by the reference getAdditionalInfo, which has not been detailed further as its content

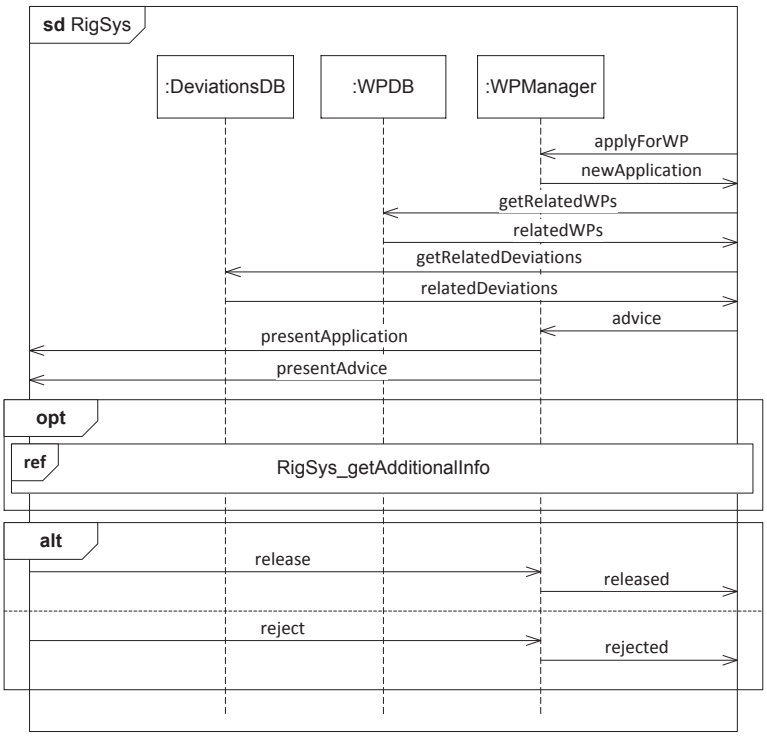

Fig. 6. Details of message exchange within RigSystem

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.

The UML sequence diagram in Figure 6 shows a decomposition of the RigSystem lifeline of Figure 5. All communication with external components go to/from WPManager, except the requests from WPAgent to WPDB and DeviationsDB.

4 Outline of a Method for Compositional Risk Analysis

In this section we outline a method for compositional risk analysis that makes use of target decomposition and risk model encapsulation. The method follows a top-down approach where we start with a high-level view of the target as a whole. The target is then decomposed before a risk analysis is conducted for each sub-target separately.

Our method is closely based on the risk analysis process as defined by the ISO 31000 standard on risk management. The process consists of five consecutive steps described as follows. 1) Establish the context involves defining the external and internal parameters to be accounted for when managing risk, and to set the scope and risk criteria for the risk management policy. 2) Risk identification is

to find, recognize and describe risks. 3) Risk estimation is to comprehend the nature of risk and to determine the risk level. 4) Risk evaluation is to compare the risk estimation results with the risk criteria to determine whether each risk and its magnitude are acceptable or tolerable. 5) Risk treatment is the process of modifying the risk. Step 2–4 are referred to as risk assessment.

The main novelties of our compositional method are the target decomposition, the sub-target risk assessment, and the risk model composition. The remaining activities mainly follow the standardized process. The target decomposition happens during the context establishment, whereas the risk model composition happens at the end of the risk assessment. In the following method overview we focus on the steps that are specific for our method, omitting details that are explained in the ISO 31000 standard.

- **Context establishment**
  - Model and document the overall target of analysis
  - Identify the assets of the overall target of analysis
  - For each asset, identify the part of the target to which the asset belongs
  - Decompose the target (and possibly the assets) such that each asset belongs to exactly one sub-target
- **Compositional risk assessment**
  - Conduct risk assessment for each sub-target separately
  - Specify the risk model interface for each sub-target
  - Build the overall risk model by composing the sub-target risk models using their interfaces

A part of the context establishment in any risk analysis consists of describing and documenting the target of analysis at an adequate level of detail. In our top-down approach to compositional risk analysis we start by modeling the whole target of analysis at a level that is suitable for providing a high-level overview and for identifying the system level assets that should be the focus of the overall analysis. For each of the assets we next identify to which part of the target it belongs, i.e. where it is located. This means that the assets must be sufficiently specific. For example, if confidentiality of health records is an asset and the records are stored at different places, we may need to split this asset up and rather specify assets like confidentiality of health data as stored on a specific database. The target is then decomposed according to the location of assets. Note that while an asset can belong to one sub-target only, one sub-target can have several assets.

In addition to taking the asset location into account, the target decomposition should ensure that each sub-target is of a size and complexity that can be handled in one analysis. If the complexity of one sub-target is too high, it must be decomposed further.

Once the target is decomposed into adequate sub-targets separate risk assessments are conducted for each sub-target individually. This basically follows the standard risk assessment process, but we also need to take into account environment threats and environment assets. Once the sub-target risk models are

completed, the respective encapsulated risk models are created. This is done by a straightforward mapping from the sub-target risk model that easily can be automated. Finally, the overall risk model is built by composing the sub-target risk models using their interfaces.

We demonstrate and exemplify the method and our techniques for compositional risk analysis over the next three sections using the petroleum work permit system. The examples illustrate essential aspects of our approach, and also serve to elaborate and further explain our notion of risk model encapsulation as introduced in Section 2.

The initial modeling and documentation of the overall target of analysis that is part of the context establishment was presented in Section 3. Before proceeding with the risk assessment, the assets need to be identified, and the target needs to be decomposed.

There are of course a number of critical information and service assets in the WP scenario. For the purpose of the example we select only a few that we focus on. Considering the rig system, it is obvious that availability of the WP data and availability of the WP advice are essential for both WP manager and for the decision maker. The availability of WP data is also essential for the WP agent that needs data for creating the advice. Considering the WP system as a whole, it is also critical to ensure the dependability of the WP agent. Because the WP agent is a software for automated decision support, the integrity of the software—including the implemented algorithms—needs to be protected. In the WP system analysis we are concerned about information security risks with respect to these assets.

Based on the identified assets we have decomposed the target into two sub-targets as indicated in Figure 7. Two of the assets are associated with the rig system, and two of them with the WP agent and its communication line to the rig system. In the remainder of the chapter we refer to the former as *sub-target A* and to the latter as *sub-target B*. Note that in this analysis the Internet weather service is part of the environment of the overall target of analysis.

In Section 5 and Section 6 we do the risk assessment and modeling for sub-target A and sub-target B, respectively. Subsequently we do the composition of the results in Section 7.

5 Risk Modeling for Sub-Target A

In this section we give a stepwise introduction to how we do the risk assessment for sub-target A by describing three different cases. We start with the simple situation where all threats and assets are internal, i.e. Case I is the identification of risks with respect to threats and assets only within sub-target A. Then we also consider external threats, i.e. Case II takes into account also environment threats, namely the external causes that can stem for other sub-targets or from the environment of the overall target. Finally, we address the general situation, i.e. Case III considers also environment assets, namely the assets of other sub-

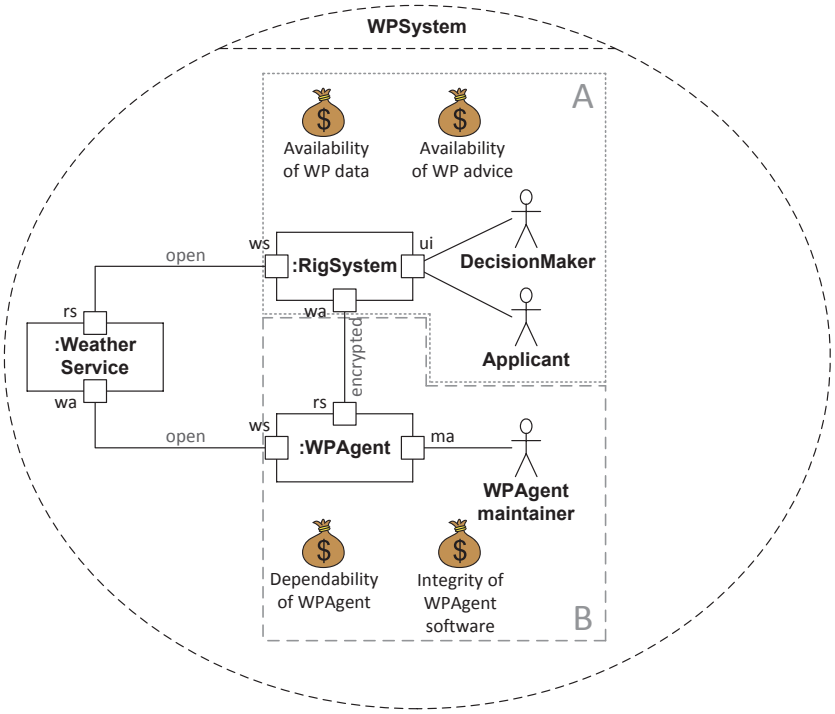

Fig. 7. Target assets and target decomposition

targets for which sub-target A can act as a source of risk. Note, importantly, that this stepwise introduction is only for pedagogical reasons, and does not indicate a specific order of what to consider during the risk assessment.

5.1 Case I: Internal Threats and Assets Only

The main purpose of our compositional approach to risk analysis is to allow individual parts of the target of analysis to be analyzed separately. In our example we have used the CORAS approach [15] for the risk assessment and risk modeling. CORAS is based on the ISO 31000 risk analysis process and comes with a language for specifying, assessing and documenting the identified risks by using so-called threat diagrams. However, our principles for risk model encapsulation and composition can be applied using also other notations for risk modeling.

Figure 8 shows our format for compositional risk modeling. It consists of three compartments, where the middle compartments includes all the threats, vulnerabilities, assets, etc. that are internal to the sub-target in question, i.e. to sub-target A in Figure 7. In the compartment to the left we model environment threats, and in the compartment to the right we model environment assets, neither of which are relevant when we restrict our attention to internal threats

and assets only. The use of the latter two compartments are exemplified and further explained in the next two sub-sections.

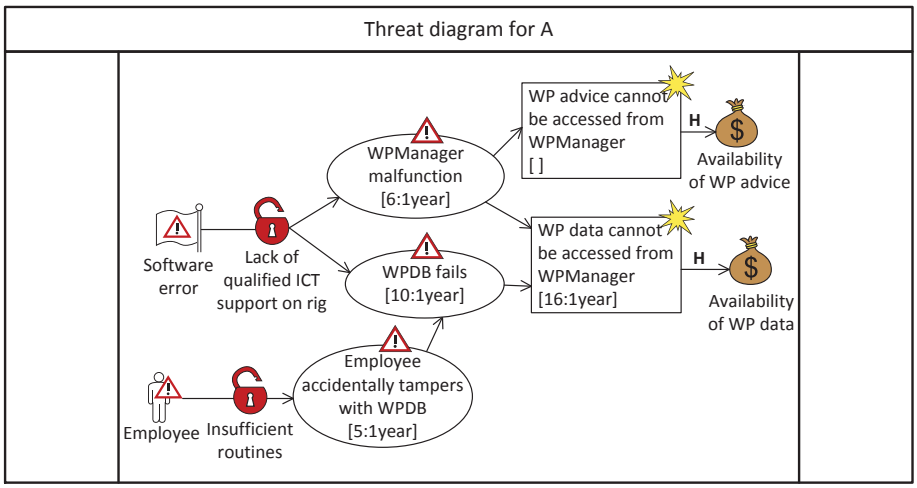

Fig. 8. Internal threats and assets only

Our example diagrams are rather small as the purpose is only to illustrate the approach. While they are based on a real industrial scenario we do not show here the actual results of a real risk analysis.

The threat diagram in Figure 8 identifies risk with respect to sub-target A. One of the identified unwanted incidents is that *WP advice cannot be accessed from WPManager*, which could be due to a software error that leads to malfunction of the WP manager. This incident harms the asset of *Availability of WP advice*. Another incident is that *WP data cannot be accessed from WPManager*, which may be due to software error or an employee that accidentally tampers with the WPDB. The asset that is harmed is *Availability of WP data*.

After the risk identification and modeling, the risk assessment proceeds with the risk estimation. This includes estimating the likelihood of the unwanted incidents to occur, as well as their consequences for the assets they harm. In the diagram the consequences are annotated on the impacts relations from unwanted incidents to assets. In our example we have used frequencies for the likelihood estimation, and we have used a scale of the three consequence levels high (H), medium (M) and low (L) for the consequences. The consequence values must be precisely defined for each asset, but this is omitted here as it is not important for the purposes of the chapter.

When estimating the frequencies for incidents to occur, we make use of likelihood estimates also for the threats and threat scenarios that lead to the incidents. The reader is referred to existing literature on CORAS for the calculus to reason about likelihoods and to do consistency checking [15, 20]. In Figure 8 we

have estimated that *WP data cannot be accessed from WPManager* occurs 16 times per year. The likelihood of the other incident, however, is not estimated at this point. This is because the analysts know that the availability of the WP advice depends also on the WP agent. Hence, we need to take into account also environment threats.

5.2 Case II: Also Considering Environment Threats

For a given sub-target the environment threats are the causes or origins of risks that are external to the sub-target. In Figure 9 we see that one such external threat to sub-target A is that *WPAgent fails to deliver advice*. Importantly, because this threat occurs outside of A, the estimation of its likelihood is not part of the risk assessment of A. Instead the variable x_1 is used such that we get a parameterized specification of the likelihoods of the scenarios and incidents that this threat may cause.

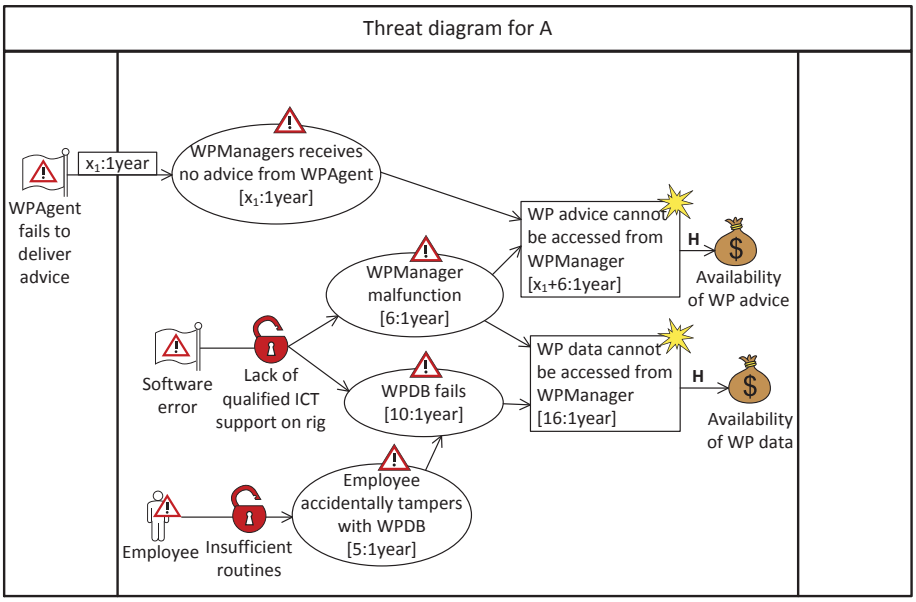

Fig. 9. Also considering environment threats

The environment threat in question may lead to the threat scenario *WPManager receives no advice from WPAgent*. Assuming that the identified threat is the only cause of this scenario, the estimated frequency is x_1 per year as annotated in the diagram. The estimation of the frequency of the resulting incident is done on the basis of the two scenarios that lead to it. As specified in Figure 9 the estimated frequency is the sum $x_1 + 6$ occurrences per year.

As we will see later the estimation of x_1 is done as part of the assessment of sub-target B, and this input is used when composing the threat diagrams to generate the risk picture for the overall target.

5.3 Case III: Also Considering Environment Assets

As we explained in the previous sub-section, compositional risk assessment must take into account also environment threats. In order to understand and analyze how one sub-target can act as an environment threat for another sub-target, we need a way to systematically consider all the other sub-targets.

Our approach to do this is to take into account all assets of the overall target in each individual risk assessment. However, while considering all assets, we still distinguish between the internal assets and the environment assets. This is illustrated in the threat diagram for sub-target A shown in Figure 10. One of the

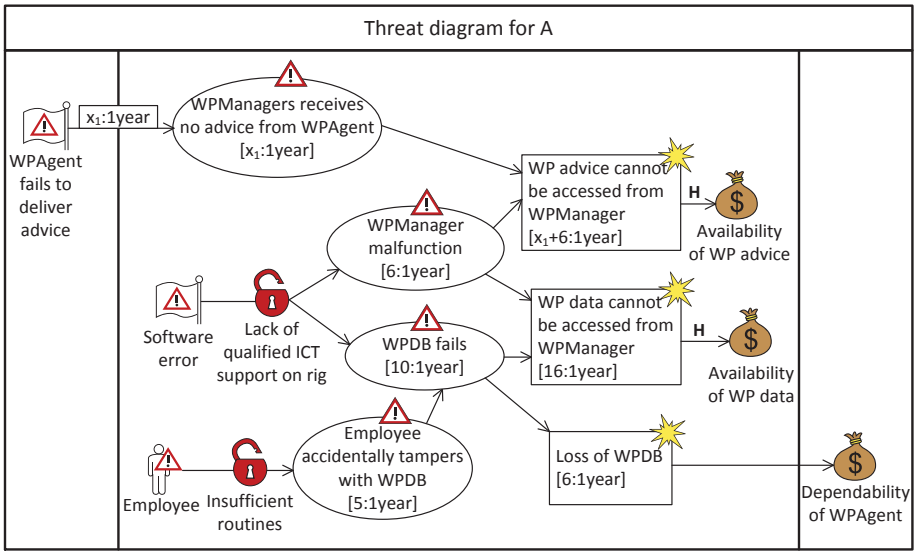

Fig. 10. Also considering environment assets

assets of the analysis that do not belong to A is *Dependability of WPAgent*. In the diagram this asset is placed in the environment compartment to the right. As part of the risk assessment of A we identify all incidents that may have an impact on any of the environment assets. In the example diagram, one such incident is *Loss of WPDB*. We use the environment impacts relation to specify this potential impact from A to the environment asset in question.

Note importantly that the consequence estimation for the environment assets is not done as part of the risk assessment of the sub-target in question. Exactly how incidents of the sub-target in question may impact assets belonging to other

sub-targets needs to be analyzed as part of the risk assessment of each of the impacted sub-targets. This includes the estimation of the consequences.

6 Risk Modeling for Sub-Target B

In Figure 11 we exemplify a completed threat diagram for sub-target B. We see here that the incident *WPAgent fails to deliver advice* may impact the external asset *Availability of WP advice*. This asset belongs to A, which is why this incident occurs as an external threat in the threat diagram for A shown in Figure 10. From the diagram in Figure 11 we also see that incidents of one sub-target may impact its own assets as well as environment assets.

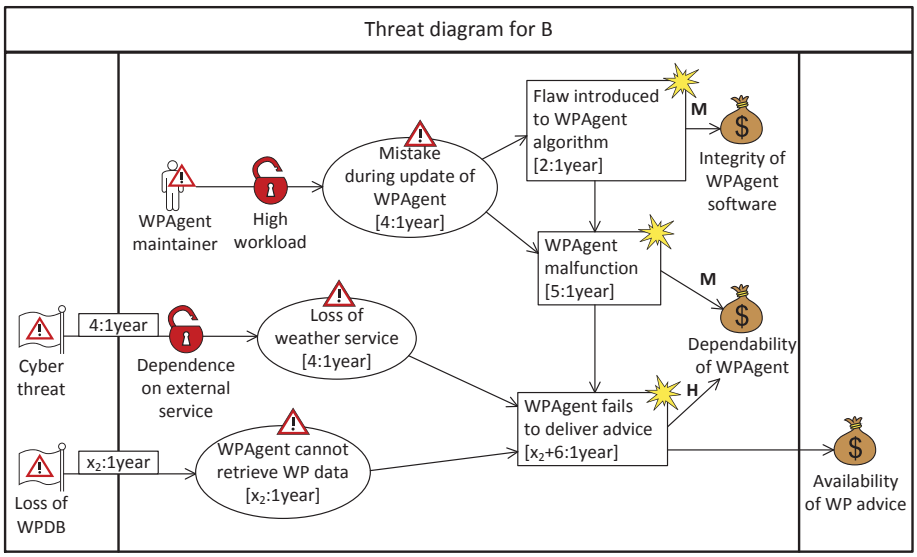

Fig. 11. Threat diagram for sub-target B

In the threat diagram for B there are two environment threats, namely *Cyber threat* and *Loss of WPDB*. The latter stems from A, while the former stems from the environment of the overall target. More specifically, in this case the cyber threat initiates an attack on the weather service that is provided over the Internet. Such a threat could, for example, be denial of service or malware. For the assessment of B it suffices to take into account the potential loss of the weather service and to estimate the likelihood.

Recall from the previous section that in principle we do not estimate the likelihoods of environment threats. This is why *Loss of WPDB* is assigned the variable x_2 in Figure 11. However, for environment threats that are part of the environment of the overall target, we can choose to make an estimate. This is

exemplified for the cyber threat where we have specified the frequency $4 : 1$ *year*. Such an estimate can be based, for example, on logs or historical data. Alternatively these estimates can be done during the risk composition. In that case the risk assessment for the sub-target in question gives a parameterized specification of also these kinds of environment threats.

The frequency estimation of the incident *WPAgent fails to deliver advice* is based on the estimates of the two scenarios and the incident that lead to it. Using x_2 as input variable, the estimate for this incident is $x_2 + 6$ occurrences per year.

7 Risk Composition

The threat diagrams introduced in the previous sections give the white-box view of the risk model for each sub-target; their purpose is to support the full risk assessment of the sub-targets, including all the internal threats, vulnerabilities and threat scenarios. To facilitate their composition, however, we create their corresponding interface diagrams.

The interface diagrams for A and B are depicted in Figure 12 and Figure 13, respectively. The interface diagrams contain the information that is needed to compose the different diagrams to yield the overall risk picture, and to document all of the risks with their risk levels.

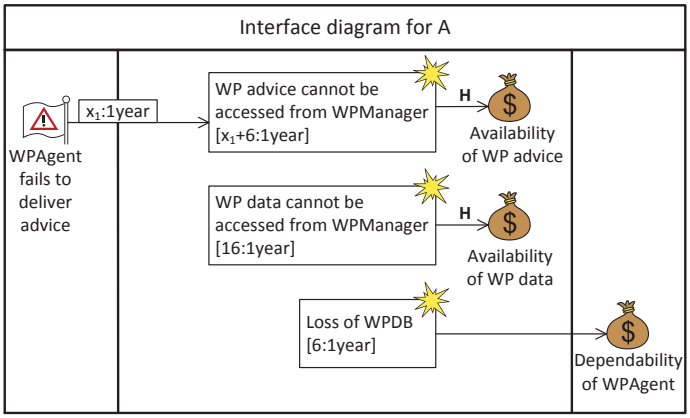

Fig. 12. Interface diagram for A

When composing the threat interface diagrams the variable x_2 in Figure 13 is instantiated with the value 6 from the incident *Loss of WPDB* in Figure 12. The likelihood of the unwanted incident *WPAgent fails to deliver advice* is then calculated by $x_2 + 6$, which gives $12 : 1$ *year*.

The resulting threat interface diagram for A and B composed, and hence for our overall target of analysis, is depicted in Figure 14. Since the diagram

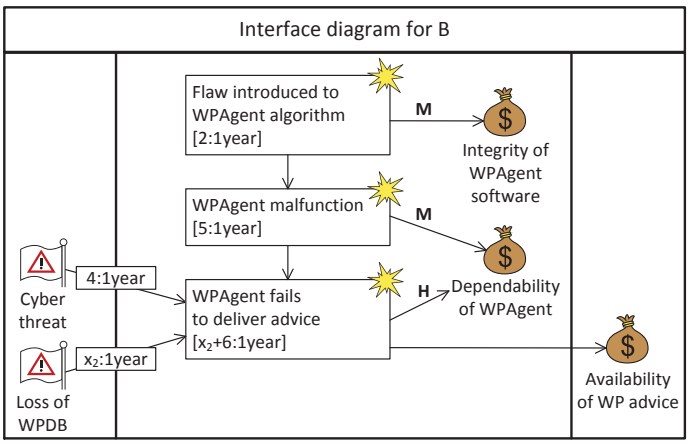

Fig. 13. Interface diagram for B

covers the whole target the set of environment assets is empty. Moreover, the only environment threat is the one that belongs to the environment of the overall target.

The interface diagram for the full target shows all unwanted incidents with respect to the assets we identified during the context establishment. It also shows the likelihood and consequence estimates for each of the incidents. Because a risk is defined as an unwanted incident with its likelihood and consequence, we have in our example identified five risks. The risk levels are calculated by using a risk function such as a risk matrix.

In this paper we have focused on risk model encapsulation and compositional risk assessment. The steps of our outlined method cover the first four steps of the risk analysis process as defined by the ISO 31000 standard. The last step is the risk treatment, which is outside the scope of this chapter. Deciding which risks that need to be considered for possible treatment is done by comparing the resulting risk levels with the risk evaluation criteria that are defined during the context establishment.

8 Related Work

Few approaches to risk management and security assessment provide support for modularity, decomposition and compositionality. Similar to [18], by compositionality we mean that risk models can be composed without considering their internal details.

Traditional risk assessment methods typically do not take into account that the risk level towards component-based systems may change given changes in the environment of the systems [21]. Instead, they rely on analyzing systems as a whole [14], without providing means for deducing the effect of composition with respect to risk. However, there also exist approaches that provide some degree

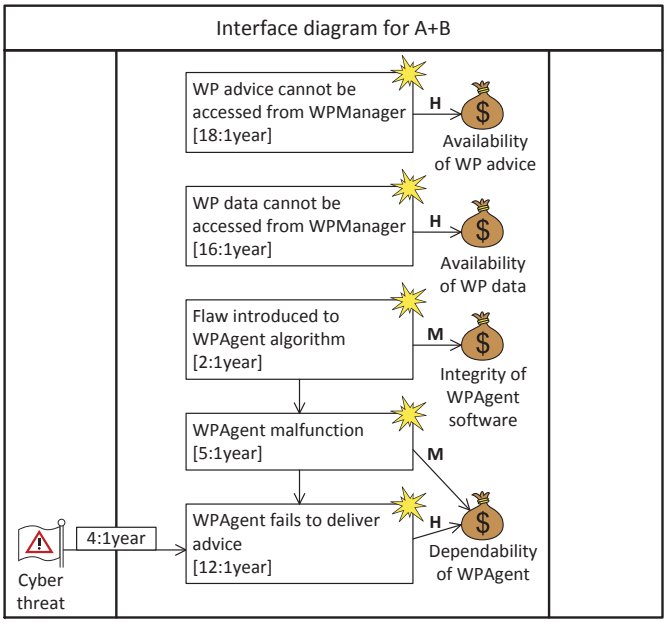

Fig. 14. Interface diagram for the composition of A and B

of support for a modular and compositional approach. In the following we give an overview of these.

Some approaches to hazard analysis address the propagation of failures in component-based systems by matching ingoing and outgoing failures of individual components. In [7, 8] UML [17] component diagrams and deployment diagrams support a method for compositional hazard analysis. Fault trees [10] are used to describe hazards and the combination of component failures that can cause them. For each component, the method is used to describe a set of incoming failures, outgoing failures, local failures (events) and the dependencies between the former two. Failure information of components can be composed by combining their failure dependencies. Likelihood of failure can be analyzed in terms of probability. In the case of AND ports, this is done by multiplication, which means that there is an assumption about independence between incoming elements. This differs from our approach, which allows the use of frequencies rather than probabilities for threat scenarios and unwanted incidents in order to facilitate better understanding [9]. Furthermore, the CORAS approach makes no assumptions about independence or overlap between threat scenarios and does not impose strong restrictions on the propagation of likelihood values, although a number of rules for likelihood reasoning and checking consistency of diagrams are offered [15].

A technique for compositional fault tree analysis (FTA) is proposed in [13]. Component failures are described by specialized component fault trees that can

be combined into system fault trees via input and output ports. Similar to our approach, different component fault trees can be developed by different user groups, composed without considering internal details, and reused. However, as usual for FTA-based approaches, likelihood analysis is performed in terms of probability and makes independence assumptions. Moreover, there is no specific support for risk analysis concepts such as unwanted incidents, threats and vulnerabilities, or links to an overall risk management process.

A denotational model for component-based risk analysis is presented in [4]. Here, a component model is provided that integrates the explicit representation of risks as part of the component behavior. Similar to our notion of encapsulation, a hiding operator is defined which allows partial hiding of internal interactions. However, interactions affecting the component risks are not hidden. Unlike our approach, the intention is to provide a theoretical foundation. Hence, the focus is on formal representation and analysis rather than direct support for practitioners. Component behavior is represented by probability distributions over communication histories, and the use of frequencies is not supported. The model is aimed exclusively at component-based systems.

In [3] dependent risk graphs are introduced as a technique to support modular risk modeling and assessment. Dependent risk graphs provide support for documenting and reasoning about assumptions and dependencies. The approach uses an assumption-guarantee style by dividing a risk graph into an assumption part and a target part. Typically, the assumptions concern the environment of the target. This facilitates modular risk assessment by the support for decomposing the target of analysis and later combining the assessment results. For example, when decomposing a target system into two, the target in one may serve as the assumptions in the other and vice versa. Once the two separate risk assessments are completed, a calculus provides rules for combining the results into one risk graph. However, no notion of risk model encapsulation is provided.

The use of risk graphs as the basis facilitates instantiation in other graph-based risk modeling approaches. In [3] this is demonstrated by the instantiation in CORAS. In [5] this modular and component-based approach to risk assessment using CORAS is integrated into a component-based system development process to support risk assessment in the development process. The instantiation in CORAS is further elaborated in [15], resulting in an extension referred to as Dependent CORAS.

In [22] an extension of CORAS is suggested that explicitly supports components by representing them with reusable threat interfaces. Threat composition diagrams representing more complex systems can then be composed from the threat interfaces, although the approach is not fully compositional. Unlike our approach, threat interfaces have (only) vulnerabilities as input ports and unwanted incidents as output ports. In addition, relations between input ports and output ports show propagation of likelihood. Even if the original CORAS method is asset-driven, assets are not included in the threat interface for a component, and there is no distinction between internal and external assets. Likelihood calculations are done in terms of probability in a similar way as for fault trees,

although [22] allows directed acyclic graphs, rather than just trees. To this end, AND/OR gates and dependency sets are introduced. The dependency sets distinguish between different occurrences of an unwanted incident depending on triggering conditions and their dependencies. These additions facilitate detailed analysis of probability at the cost of significantly increasing the complexity of the approach.

While some of the above works share certain characteristics with our approach, we are not aware of existing approaches similar to the one we propose. It is designed to be compositional, simple and general. The approach is simple in the sense that no new constructs are added to the modeling language except from the diagram frames. It is general in the sense of being applicable not only for component-based systems, but also for other settings where a partitioning of risk models is appropriate, for example based on aspects or business concerns. As illustrated above, most methods and techniques focus primarily on failure rate, likelihood or risk level assessment in a component-based setting. While this is an important ingredient of component-based risk analysis, the lack of an encapsulation mechanism for many existing techniques complicates composition and means that composed models may become very large and complex, and thus hard to understand and work with.

9 Conclusion

We have presented a top-down approach to compositional risk analysis where the target of analysis is decomposed in such a way that each identified asset belongs to exactly one sub-target. A separate risk model is then developed for each sub-target, and the individual risk models are eventually combined to arrive at a risk model for the whole target. The approach follows ISO 31000, but provides additional support for the context establishment and risk assessment phases specifically aimed to facilitate decomposition and composition.

At the core of the approach is a novel notion of risk model encapsulation, where only the elements that are essential for composition are exposed through an explicitly defined *risk model interface*, while internal details are hidden. All one needs to know in order to compose risk models is the contents of their interfaces. By hiding the internal details we make it easier for practitioners to compose risk models, while at the same time reducing the size and complexity of the resulting model. An added benefit is that a risk model interface contains the information that would typically be of interest for managers and decision makers who often have little time and have not themselves taken part in the risk assessment.

Encapsulation is a key reason for the success of object-oriented programming. We believe that significant benefits can be achieved by introducing this concept into risk management and analysis. We are not aware of any other approach offering a clear encapsulation concept for risk analysis allowing compositional reasoning.

The work presented here opens up a number of interesting directions for further research that we hope to pursue. In particular, a more complete method with detailed techniques and guidelines for practitioners should be developed. We would also like to explore how our notion of encapsulation could be applied in a bottom-up approach. The added challenge here is that we cannot assume that the environment of a target is known at the time when the corresponding risk model is developed. Finally, we would of course like to validate and refine our results by applying them on a variety of case studies.

Acknowledgments. The research presented in this chapter was partially funded by the European Commission via the FP7 projects NESSoS (256980) and RASEN (316853), by the ARTEMIS Joint Undertaking and the Norwegian Research Council via the CONCERTO project (333053 and 232059), and by the Norwegian Research Council via the Dynamic Risk Assistant project (217213).

References

1. Agence nationale de la sécurité des systèmes d'information: EBIOS 2010 – Expression of Needs and Identification of Security Objectives (2010), in French
2. Alberts, C.J., Dorofee, A.J.: OCTAVE Criteria. Tech. Rep. CMU/SEI-2001-TR-016, CERT (December 2001)
3. Brændeland, G., Refsdal, A., Stølen, K.: Modular analysis and modelling of risk scenarios with dependencies. Journal of Systems and Software 83(10), 1995–2013 (2010)
4. Brændeland, G., Refsdal, A., Stølen, K.: A denotational model for component-based risk analysis. In: Proc. International Symposium on Formal Aspects of Component Software (FACS'11). LNCS, vol. 7253, pp. 12–41. Springer (2012)
5. Brændeland, G., Stølen, K.: Using model-driven risk analysis in component-based development. IGI Global pp. 330–380 (2011)
6. CRAMM – The total information security toolkit. `http://www.cramm.com/`, accessed 13 June 2012
7. Giese, H., Tichy, M.: Component-based hazard analysis: Optimal designs, product lines, and online-reconfiguration. In: Proc. Computer Safety, Reliability and Security (SAFECOMP). LNCS, vol. 4166, pp. 156–169. Springer (2006)
8. Giese, H., Tichy, M., Shilling, D.: Compositional hazard analysis of UML component and deployment models. In: Proc. Computer Safety, Reliability and Security (SAFECOMP). LNCS, vol. 3219, pp. 166–179. Springer (2004)
9. Gigerenzer, G.: Calculated Risks – How to Know When Numbers Deceive You. Simon & Schuster (2002)
10. International Electrotechnical Commission: IEC 61025 Fault Tree Analysis (FTA) (1990)
11. International Organization for Standardization: ISO 31000 Risk management – Principles and guidelines (2009)
12. International Organization for Standardization / International Electrotechnical Commission: ISO/IEC 27001 – Information technology – Security techniques – Information security management systems – Requirements (2005)

13. Kaiser, B., Liggesmeyer, P., Mäckel, O.: A new component concept for fault trees. In: Proc. 8th Australian workshop on Safety critical systems and software (SCS). vol. 33, pp. 37–46. Australian Computer Society (2003)
14. Lund, M.S., Solhaug, B., Stølen, K.: Evolution in relation to risk and trust management. Computer 43(5), 49–50 (2010)
15. Lund, M.S., Solhaug, B., Stølen, K.: Model-Driven Risk Analysis – The CORAS Approach. Springer (2011)
16. Microsoft Solutions for Security and Compliance and Microsoft Security Center of Excellence: The Security Risk Management Guide (2006)
17. Object Management Group: OMG Unified Modeling Language (OMG UML), Superstructure. Version 2.3 (2010), OMG Document: formal/2010-05-03
18. de Roever, W.: The quest for compositionality – A survey of assertion-based proof systems for concurrent programs, part 1: Concurrency based on shared variables. In: Proc. IFIP Working Conference on the Role of Abstract Models in Computer Science. North-Holland (1985)
19. Stoneburner, G., Goguen, A., Feringa, A.: Risk management guide for information technology systems. Tech. Rep. 800-30, NIST (2001)
20. Tran, L.M.S., Solhaug, B., Stølen, K.: An approach to select cost-effective risk countermeasures exemplified in CORAS. Tech. Rep. A24343, SINTEF ICT (2013)
21. Verdon, D., McGraw, G.: Risk analysis in software design. IEEE Security & Privacy 2(4), 79–84 (2004)
22. Viehmann, J.: Reusing risk analysis results – An extension for the CORAS risk analysis method. In: Proc. 4th International Conference on Information Privacy, Security, Risk and Trust (PASSAT). pp. 742–751. IEEE (2012)