Specifying Legal Risk Scenarios Using the CORAS Threat Modelling Language

Experiences and the Way Forward

Fredrik Vraalsen[1], Mass Soldal Lund[1], Tobias Mahler[2], Xavier Parent[3], and Ketil Stølen[1]

[1] SINTEF, Norway
{fvr, msl, kst}@sintef.no
[2] Norwegian Research Center for Computers and Law, University of Oslo, Norway
tobias.mahler@jus.uio.no
[3] King's College London, UK
xavier@dcs.kcl.ac.uk

Abstract. The paper makes two main contributions: (1) It presents experiences from using the CORAS language for security threat modelling to specify legal risk scenarios. These experiences are summarised in the form of requirements to a more expressive language providing specific support for the legal domain. (2) Its second main contribution is to present ideas towards the fulfilment of these requirements. More specifically, it extends the CORAS conceptual model for security risk analysis with legal concepts and associations. Moreover, based on this extended conceptual model, it introduces a number of promising language constructs addressing some of the identified deficiencies.

1 Introduction

The notion of trust is tightly interwoven with notions like security and usability [1, 2]. Furthermore, it is difficult to separate trust from the expectation of a legal framework that offers protection in the cases where the trust relationship fails [3]. An analysis of trust should therefore encompass a number of issues including technological, legal, sociological and psychological aspects.

Since the trustor may be unable to monitor or control the trustee, it is essential that there are protective measures in place to solve situations that arise from lack of trustworthiness of the trustee. For example, Jones et al. [3] argue that "although businesses and consumers may consider underlying systems to be completely secure, they may not trust these systems with their business or personal interests unless there is a suitable legal framework they can fall back on, should problems arise." In this case, the legal framework is seen as a treatment to potential risks, e.g. economic loss. Hence, the proper foundation of trust is dependent on legal means of protection, as well as the security mechanisms that are employed to protect the system and its data [1]. Security measures, e.g.

P. Herrmann et al. (Eds.): iTrust 2005, LNCS 3477, pp. 45–60, 2005.

logging, may, however, conflict with rules and regulations for data protection and privacy. Hence, the risk of breaking existing legal rules may limit the use of trust-enhancing technologies.

Understanding how to exploit legal risk analysis [4] to achieve well-founded trust and security is seen as an important issue for research. For example, legal risk analysis can be used to do a preliminary identification and prioritisation of legal risks, based on the assets determined by the stakeholders. Risk analysis may in this way help guide the application of conventional legal analysis to the areas which are of highest business importance, thus increasing effectiveness. Some, like Richard Susskind, predict a shift from legal problem solving to legal risk management [5]. Legal risk analysis is based on traditional risk analysis methods. It focuses upon an asset and analyses possible risks to this asset.

The result of a general risk analysis may indicate where a flow of information should be controlled, e.g. by erecting legal or technological barriers. In many cases legal and technological measures can not be stricly separated. Technological measures will often reflect their own set of rules, and can be seen as a *lex informatica*, as indicated by Reidenberg [6]. Hence, risk analysis can be used to identify the necessity for rules, be it binding legal rules or non-binding but effective policies. In particular, a legal risk analysis can be conducted from the perspective of a party who is interested or involved in the flow of information, as an owner, sender, recipient or intermediate. In this case, the involved party can analyze legal risks and bring about a strategy to manage these risks, e.g. through a contract that addresses liability issues. As a final step, risk analysis can be used to analyze and test the contract, in order to control whether previously identified risks are reduced or eliminated and in order to identify additional risks that arise as consequences of the chosen treatment.

Risk analysis requires a clear understanding of the system to be analysed. Normally, this understanding can be obtained only through the involvement of different stakeholders, e.g. legal experts, security experts, system developers and users. In fact, most methods for risk identification make use of structured brainstorming sessions of one kind or another, involving 5-7 stakeholders with different backgrounds. This poses a challenge with regards to communication and understanding between stakeholders. The effectiveness of such sessions depends on the extent to which the stakeholders and analysts involved understand and are understood by each other.

The CORAS language for threat modelling [7, 8, 9] has been designed to mitigate this problem within the security domain. This paper evaluates the suitability of this language to specify legal risk scenarios. It also presents ongoing research focusing on how to increase the expressiveness to provide specific support for the legal domain. The remainder of this paper is structured as follows. In Sect. 2 we give an overview of the existing CORAS language. Models resulting from applying CORAS for the analysis of legal issues are presented in Sect. 3. Section 4 is devoted to ongoing work on extending the language to specifically support legal risk analysis. Finally, Sect. 5 draws the main conclusions and outlines future work.

2 The CORAS Threat Modelling Language

The CORAS language for threat modelling is a graphical language allowing documentation of undesirable behaviour in the form of threat scenarios. The CORAS language covers notions like asset, threat, risk and treatment, and supports communication among particpants with different backgrounds through the definition of easy-to-understand icons (symbols) associated with the modelling elements of the language. The CORAS language is an extension of the UML 2.0 [10] specification language, the de facto standard modelling language for information systems. It is defined as a UML profile [11], and has recently become part of an OMG standard [12].

The underlying assumption when modelling threats with the CORAS language is that the threats are part of the behaviour of a computerized system, whereas vulnerabilities are features of the system, even though these are negative or unwanted features. This basically means that the same modelling techniques may be used for modelling threats as for modelling the desired features and functionality of a system. The CORAS language provides a number of specialized UML diagrams supporting threat modeling, such as asset diagrams, threat & unwanted incident diagrams, and treatment diagrams.

Assets. Figure 1 shows a subset of the CORAS risk analysis conceptual model [11]. This sub-model defines part of the context of a risk analysis. The context consists of the *stakeholders* and *assets* of the system under analysis. A risk analysis is asset-driven, which means that the analysis is carried out relative to the identified assets. A stakeholder is a person or organisation that has interests in the target of evaluation (ToE). In the general case, an asset may be anything that stakeholders of the target find to have value. An *entity* is a physical or abstract part or feature of the target that becomes an asset when assigned value by a stakeholder.

Threats and unwanted incidents. A threat is described in the CORAS language using a *threat agent*, e.g. a disloyal employee or a computer virus. The threat agent initiates a *threat scenario*, which is a sequence of events or activities leading to an *unwanted incident*, i.e. an event resulting in a reduction in the value of the target asset.

Each threat agent is related to one or more threat scenarios, represented by ovals with ignited bombs. The threat scenarios are again related to the assets

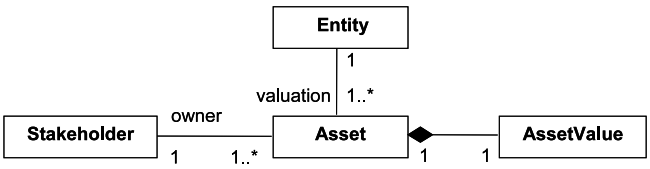

Fig. 1. Risk analysis context sub-model

they threat, represented as stacks of coins. Even though a threat exists, this does not mean that something bad will necessarily happen. The actual incident may also be the result of interplay between a number of threats. We model this by the use of unwanted incidents, represented as ovals with warning signs. The «include» arrow is used for expressing that an unwanted incident includes a threat scenario. The threat scenario pointed at describes a subscenario of the scenario associated with the unwanted incident. Further, an unwanted incident may lead to another unwanted incident, forming a chain of events. This is modelled by the «Initiate» arrow.

Figure 2 shows an example threat & unwanted incident diagram. In this example we have modelled threats to a telemedicine system. The system uses a dedicated network for transmitting medical data, allowing a general practitioner and a cardiology expert to examine a patient together even though they are physically located at different sites. This involves retrieving the patient's health record from a database and transmitting the relevant data to be provided to the application clients (presentation layer) running at both medical doctors' computers.

Taking the patient's view, the target of the threats is his/her medical data, more specifically the health record. There are two threats to this health record (1) input of wrong data, resulting in a health record containing misleading medical information; (2) illegal access to the data when transmitted over the network, with the result that the data get known by unwanted people which again may result in the data being used against the patient's interests.

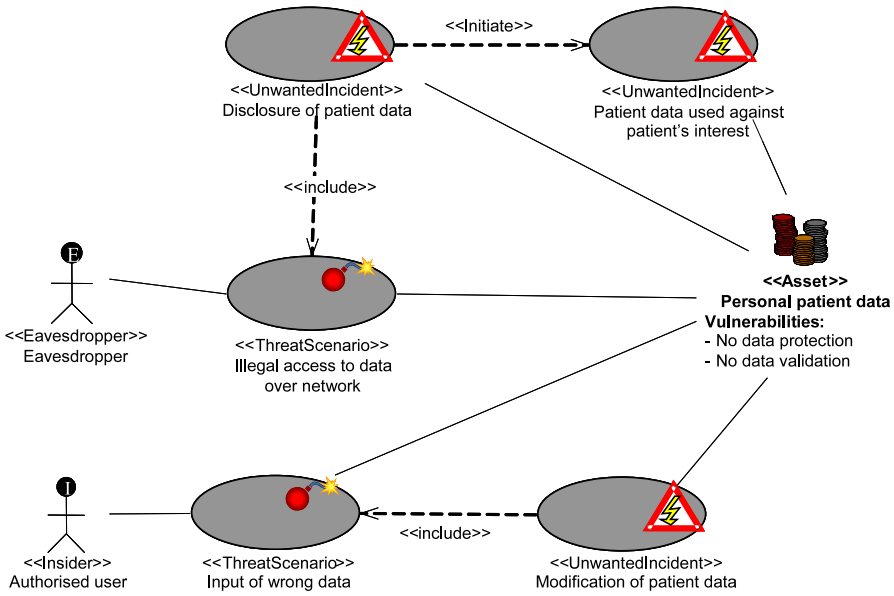

Fig. 2. Threat & unwanted incident diagram

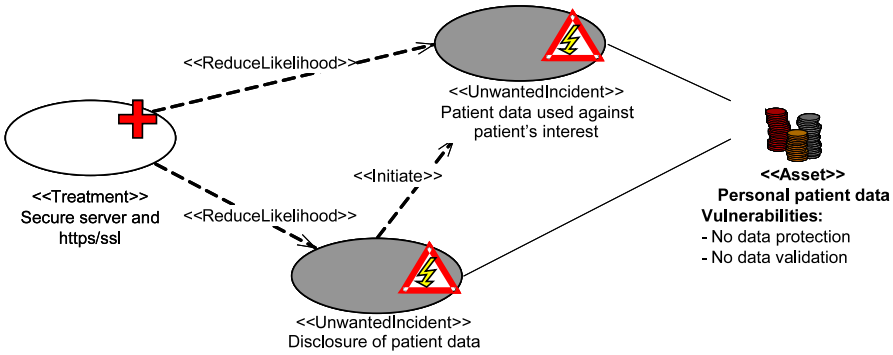

Fig. 3. Treatment diagram

In the example, the unwanted incident "Disclosure of patient data" includes the threat that an eavesdropper gets illegal access to the network, and may lead to the unwanted incident that the patient data is used against the patient's interests. As wee see, both unwanted incidents are related to the asset, meaning that the occurrence of these events have consequences for the asset.

Treatments. Once the threats have been identified and analysed, we go on to identifying treatments in order to reduce the risks. As with threats and unwanted incidents, treatments are modelled by specialized use cases. Treatment options are shown as ovals with a red cross in the corner, and are connected to the threats and unwanted incidents they treat with dashed arrows. One obvious treatment to the threats in the example above is introduce protection mechanisms like encryption of the data being transmitted over the network. This is shown in the example treatment diagram in Fig. 3.

3 Using CORAS Within the Legal Domain

We have evaluated the applicability of the CORAS language for legal risk analysis using two legal risk scenarios defined in the TrustCoM project [13]. The scenarios deal with issues related to data protection law and intellectual property rights (IPR), respectively. Some of the modelling results from the latter scenario are presented in Sect. 3.1. Based on these trials, we have come up with requirements to an extended language, which are discussed in Sect. 3.2.

3.1 Analysis Results

The scenario forming the basis for the following results deals with a group of engineering companies who are forming a Virtual Organisation (VO) in order to collaborate on a project to design an aircraft. To accomplish this, they of course need to share information, such as design drawings for the various aircraft components. A number of challenges related to intellectual property rights arise from this scenario, such as who has the right to access and distribute the data,

as well as possible remedies when non-disclosure agreements are broken. These issues are of course also tightly connected to technical issues, such as security infrastructure and policies.

Before we are able to start identifying threats, we need to define the target of evaluation through e.g. UML models and natural text documenting the parts of the virtual organisation and its processes we are interested in. Consider a small example of company A and company B collaborating in the above mentioned project. Both companies hold intellectual property, "A's information" and "B's information" respectively, that they want to keep confidential (trade secrets) but which they agree to share with each other in the project in order to reach a common goal. Furthermore, they need to discuss the project idea with a potential customer C. Figure 4 shows a UML diagram documenting some of the activities that A may perform in relation to B's intellectual property.

Assume that company B is the client of a risk analysis. That is, B is regarded as stakeholder, and threats are identified from the viewpoint of B. In order to protect their interests, the involved parties set up an agreement that deals with the use of confidential information and its non-disclosure to third parties. Through the risk analysis, the probabilities and consequences, e.g. monetary loss, of the various risks are determined. These results may form the basis for determining which aspects should be included in e.g. non-disclosure agreements, such as who may receive the information and the required level of security, as well as appropriate legal sanctions if the agreement is broken.

Hazard and Operability (HazOp) analysis [14] was employed to identify threats. HazOp is a form of structured brainstorming session where the knowledge and expertise of all the participants is exploited in order to find as many relevant threats as possible. There are two broad classes of risks in this example as seen from B's perspective: (1) The disclosure of B's confidential information to a competitor, caused by either A or a third party. (2) A possible liability for B himself disclosing A's confidential information in breach of the non-disclosure agreement. Figure 5 shows a threat & unwanted incident diagram documenting some of the threats and unwanted incidents which were identified in relation to the disclosure of confidential information.

An example of a legal threat is the possibility of a lawsuit against B if B is responsible for A's information being leaked to a third party (competitor),

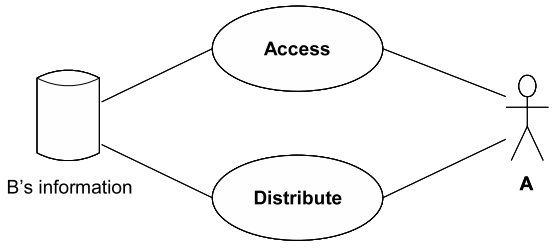

Fig. 4. Target of Evaluation (ToE)

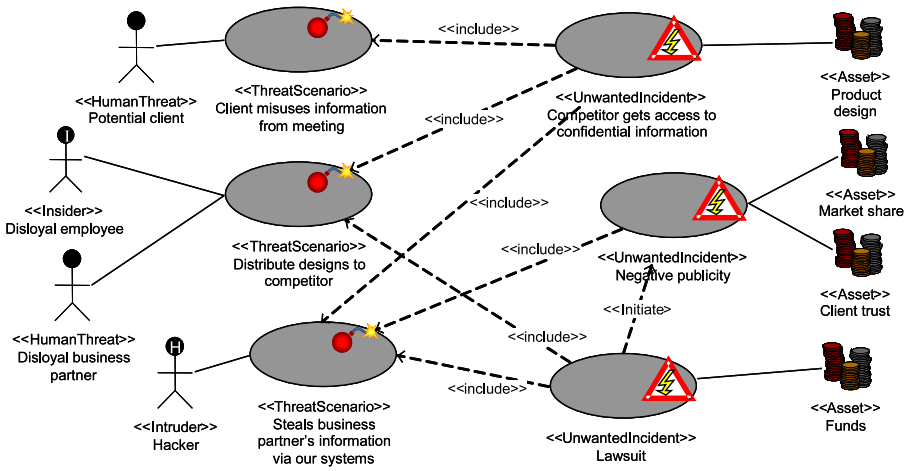

Fig. 5. Threat & unwanted incident diagram

either on purpose, e.g. a disloyal employee, or through negligence, e.g. insufficient security measures. One benefit of legal risk analysis is that legal risks may be incorporated into a larger risk management picture. Non-legal consequences of legal risks, e.g. negative media exposure as a result of a lawsuit, may turn out to be equally important from a business perspective but are typically not considered in conventional legal analysis.

Once the risks have been identified and evaluated, one can identity and assign treatments to reduce the risks. Treatments may for example be to establish a non-disclosure agreement with the potential client C to regulate what they may do with the information provided to them in the meeting. Alternatively, the companies may agree to not disclose any confidential information to each other in the meeting. Legal treatments may be used both against legal threats, e.g. contract clauses limiting liability to reduce the consequence of a potential lawsuit, or against non-legal threats, e.g. hackers may be prosecuted by law. Similarly, treatments to legal threats may also be of a non-legal nature, e.g. improvements to the security infrastructure to reduce the likelihood of a lawsuit. These treatments are shown in the treatment diagram in Fig. 6.

3.2 Requirements to an Extended Language

As the examples above have shown, we are able to use the CORAS language to document and analyse some aspects of legal risks and the current language seems to be a good starting point for legal risk analysis. However, we wish to be able to model and analyze more legal aspects, for example whether the act of disclosing certain information, as shown in Fig. 4, infringes upon a non-disclosure agreement. To enable this, we need facilities for:

- specifying ownership, which is highly relevant when determining e.g. the rights and obligations of an actor,

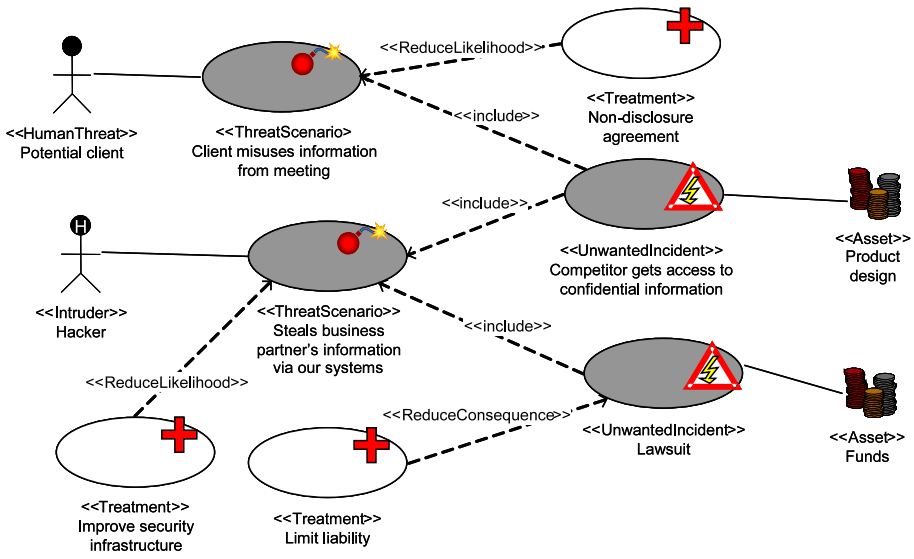

Fig. 6. Treatment diagram

- specifying legal effects on different roles and activities, and
- correlating these effects with the relevant legal sources, e.g. which contract clause is the source of the legal effect in question.

We thus see the need to incorporate more information relevant to legal aspects into the graphical language. Furthermore, to facilitate the use of the graphical modelling language for documentation and communication of legal risk analysis results, the users of the language need a clear understanding of what the graphical models express. A graphical language for legal risk analysis should on the one hand be easily understandable for practitioners and on the other hand be sufficiently precise to allow in-depth analysis. We must be able to explain the meaning of the diagrams as well as how they can be combined and refined. Furthermore, to support automated analysis of the graphical models, tools must be able to extract and process relevant information. To enable this, the semantics of the graphical language need to be defined, with particular emphasis on the notions of trust, security, privacy, data protection and intellectual property rights.

4 Towards a More Expressive Language

To meet the requirements stated above, we are extending the CORAS language with concepts and relationships relevant to legal analysis. A central conjecture is that modal logic [15] may be an important source of inspiration with respect to what kind of language constructs are required. In particular, we are looking at deontic logic, a form of modal logic which deals with the notion of obligation.

This will enable us to specify e.g. which activities are permitted, obligated and forbidden.

Formal conceptual modelling is employed in Sect. 4.1 to provide a semantic basis for the graphical language. Some examples of how the extended conceptual model may be used to improve the expressiveness of the graphical language are presented in Sect. 4.2.

4.1 Conceptual Model for Legal Risk Analysis

In this section we propose a conceptual model for legal risk analysis, described using UML 2.0 [10]. The following introduces the various parts of our conceptual model.

Central to legal risk analysis are *legal norms*, which describe legal requirements and consequences. Legal norms have the general structure of an *antecedent* and a *consequent*: if A then B.

The antecedent describes the *criteria* for the norm to apply, i.e. which factual *circumstances* have to be present. The *consequent* indicates the *legal effects* of the norm being applied. Often, an effect is only a link to further norms, which chained together will represent the norms governing the case at hand. When law is applied to a case or situation, the legal criteria are compared with the circumstances of the case. Figure 7 shows a UML diagram depicting the concepts outlined above and the relationships between them.

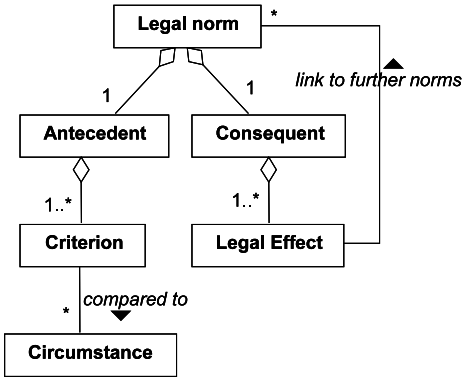

Fig. 7. Legal norm

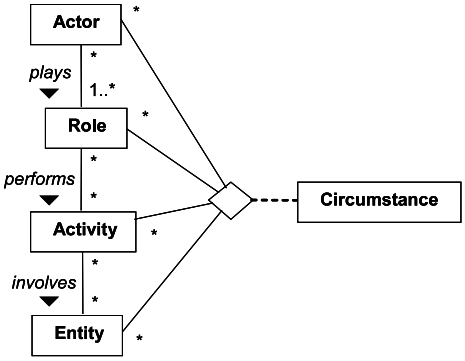

Fig. 8. Circumstance

The legal effect a particular norm has on an actor depends on the activity being performed as well as the roles that actor plays, e.g. student, employer, (system) owner, etc. A circumstance thus consists of an *actor*, an *activity* being performed by that actor, and the *role* which the actor is in while performing the activity. Another *entity* may also be involved in the activity. For example, a circumstance may be that a person (*actor*) who is an employee of company A (*role*) accesses (*activity*) some information (*entity*) which belongs to company B. Figure 8 shows the concepts and relationships related to circumstance.

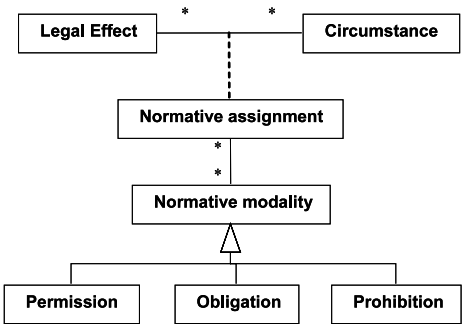

Fig. 9. Normative modalities and effects of legal norms

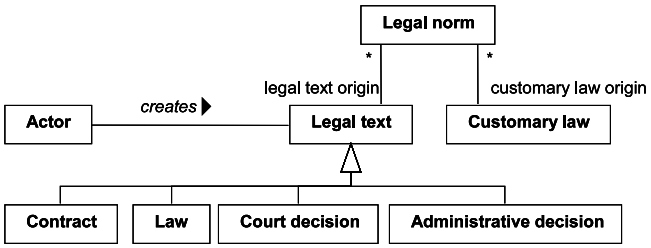

Fig. 10. Legal sources

In general, an actor may play several roles and perform many activities, each involving a number of other entities. However, by looking at each combination of these as separate circumstances, this allows us to assign different legal effects to each circumstance. For example, the person in the example above may be permitted to access the information, whereas another person employed by company C may be forbidden to access the same information.

Normative modalities are used in deontic logic to describe the normative status (*permitted*, *obligatory*, *forbidden*, and so on) assigned to a state of affairs A [15]. *Obligation* may be expressed as $\mathrm{O}A$, meaning "it is obligatory that A." The agency operator, E_i, is used to express propositions such as $\mathrm{OE}_i A$, meaning "agent i is obliged to bring it about that A" [16]. *Permission* is the dual of obligation, i.e. $\mathrm{PE}_i A = \neg\mathrm{O}\neg\mathrm{E}_i A$ ("agent i is permitted to bring it about that A"), and *prohibition* (forbidden) is simply the negation of permission, e.g. $\neg\mathrm{PE}_i A$ ("agent i is forbidden/not permitted to bring it about that A"). We assign normative modalities to the relationship between legal effect and circumstance to specify which circumstances are permitted, obligatory and forbidden by the legal norm in question, as depicted in Fig. 9. The legal criteria are derived from legal reasoning based on the relevant source material, which may include statutes, regulations, court or administrative decisions, etc. The identification of such sources itself is an essential part of a legal decision making process. Figure 10 shows an example of some legal sources. Figure 11 shows how the various concepts from

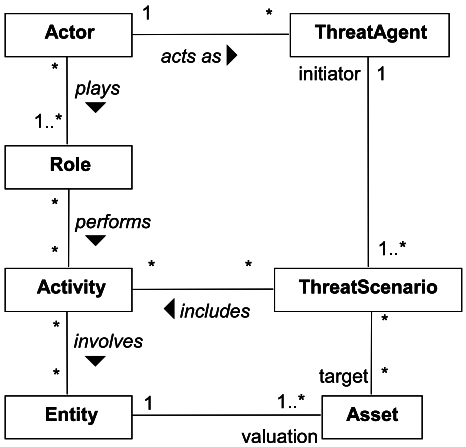

Fig. 11. Integrating threat scenario

the CORAS risk analysis conceptual model are integrated with the concepts described above. On the left hand side are the concepts related to circumstance, and on the right hand side are the concepts from the CORAS model. An actor may act as a threat agent, i.e. initiating the threat scenario. The threat scenario may include a number of activities. Entity and asset are related in the CORAS conceptual model, as shown in Fig. 1.

The client of the risk analysis may not be interested in trust as such, but rather in assets such as market share and income. *Trust* and *reputation*, however, are clearly important factors that affect customers' behaviour, and may therefore be viewed as assets of their own [17]. As mentioned in Sect. 2, an *entity* is a physical or abstract part or feature of the target that becomes an asset when assigned value by a stakeholder. We therefore also view trust and reputation as subtypes of entities. Of particular interest in the context of legal risk analysis of data protection issues and intellectual property rights (IPR) are *information* assets, as well as the notion of *ownership*.

Studies of trust distinguish between the trustor, that is, the agent that trusts another agent, and the trustee; the agent being trusted. Trust can be modelled as a binary relationship from the trustor to the trustee, denoted as *source* and *target* of the trust, respectively. Ownership is modelled as a relationship between an actor, the *owner*, and an entity. These concepts and relationships are shown in Fig. 12. The actor also plays the role of stakeholder with regards to an asset. This is the same as the stakeholder shown in Fig. 1. Actor is another specialisation of entity. An actor may be a natural person or a juristic person, e.g. an organization, as well as other types of behavioral entities, e.g. a software agent.

4.2 Exploiting the Improved Expressiveness

In legal risk analysis, parts of the target of evaluation will originate from legal issues formulated in legal texts such as laws and contracts. This motivates that

ToE descriptions not only cover technical and organizational aspects but also the legal issues. Legal texts have a tendency to be complex and also hard to read by laymen. Since participants of legal risk analyses will include people that are not legal experts, we claim that standard modelling techniques can be applied to give these participants a better understanding of the legal issues.

As discussed in Sect. 4.1, legal norms have effects that bind the roles involved in certain activities by normative modalities, such as "it is permitted for A to do X" and "it is forbidden for B to do Y". When modelling legal issues, we concentrate on modelling the *effects* of the relevant legal texts (not the legal texts themselves), and we do this by introducing the normative modalities as modelling elements in the form of UML stereotypes.

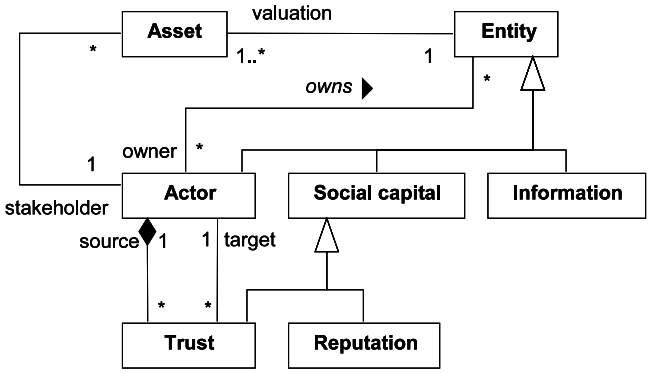

Fig. 12. Incorporating trust

Consider the example from Sect. 3, where company A and company B decide to cooperate on a shared project to reach some goal. An extract of the legal effects of the contract between A and B are modelled in the diagram of Fig. 13, which is a specialisation of Fig. 4. In this example, one of the effects of the contract is that A has permission to access B's information, modelled using the stereotype «permitted» on the association between A and the use case "access". Similarly, the effect that A is forbidden to distribute B's information is modelled using the stereotype «forbidden». The diagram also shows where these effects originate from (e.g. contract clause 3) by attaching a constraint to the association. Ownership is modelled using the «owns» stereotype on the association between B and B's information.

The CORAS language has proved useful in threat modelling [18]. In order to model threats related to legal issues, we integrate the approach above with the threat modelling features of the CORAS language. Figure 14 shows a threat and how this is related to one of the legal effects in the previous figure. Company B acts as stakeholder and company A acts a threat agent in this example, modelled using the «acts as» stereotype. Furthermore, B's information is viewed as one of B's assets, and the threat scenario includes an activity that is forbidden for A,

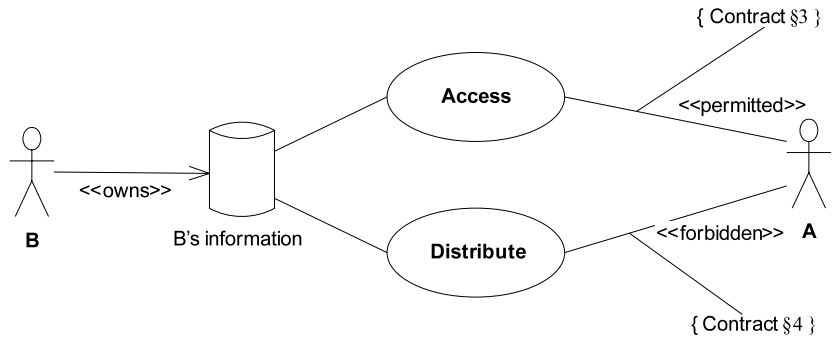

Fig. 13. Modelling legal effects

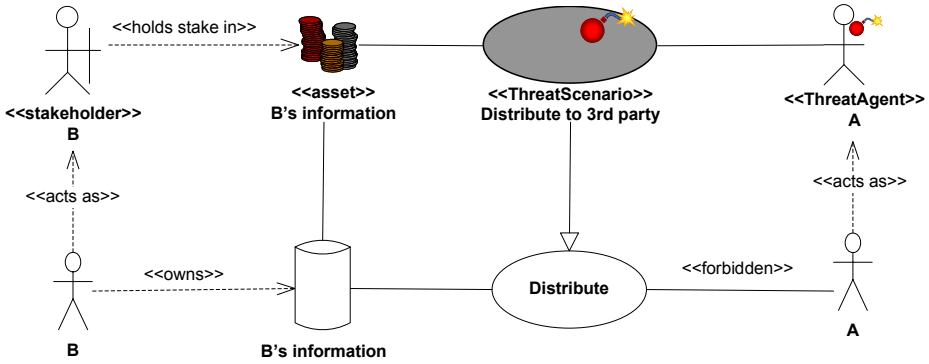

Fig. 14. Modelling legal threats

modelled using the standard UML «includes» stereotype. The structure of this diagram is similar to the one of Fig. 11, which shows the relationship between the CORAS conceptual model for risk analysis and the new concepts presented in this paper. Here, A is the actor, performing the "Distribute" activity involving the entity "B's information".

5 Conclusion

Our preliminary results from using the CORAS language for security risk analysis in legal risk scenarios show that this language may also be utilised successfully for the specification of legal risk scenarios. Furthermore, conventional legal analysis focuses primarily on the purely legal consequence of legal risks, i.e. legal sanctions such as fines. However, these legal sanctions may lead to a number of other consequences which may turn out to be equally important from a business perspective, e.g. negative media exposure and loss of market share. Technical aspects, such as protection of electronic information, will often also have legal

implications, and vice versa. Legal risk analysis enables us to integrate the legal aspects into the overall risk management picture, thus providing a better foundation for decisions with regards to which legal risks to tackle.

We have argued that the normative modalities are highly relevant to legal risk analysis and shown some examples of how these may be integrated into the language. However, a number of other notions might also prove to be relevant, e.g., the notion of exception plays an important role in agreements. We are thus investigating whether there is a need to further increase the expressiveness. This work is carried out in parallel with a revision of the CORAS language where both the meta model and the icons are revised based on experience from use and experimentation with the language. We are also currently working on various approaches for assigning semantics to our graphical language, e.g., by mapping it to a logical language which would provide an unambiguous reference point. Our approach is to formalize the conceptual model in a first order logic framework [19] and to formalize the use case based language in the STAIRS semantics [20] for sequence diagrams. A mapping between these two formalizations and an interpretation in the model theoretic foundations of deontic logic will also be provided.

The language facilities for specifying obligations and permissions may also prove useful for e.g. specification of policies. The normative modalities in Sect. 4.1 are used as the basis for policy specifications described in the ODP enterprise viewpoint [21]. Ponder [22] is a declarative, object-oriented language for specifying security and management policies for distributed systems. Unlike the graphical CORAS language, Ponder uses a textual language for specification of policies. An interesting topic for future study may be whether the extended CORAS language can be used for graphical modelling of (a subset of) the Ponder language. A number of other text-based policy languages and frameworks exist, such as XACML [23] for specification of access control policies, REFEREE [24] for specification and evaluation of trust policies, the KeyNote [25] trust and public key based authorisation system, as well as various other PKI- and credential-based approaches [26, 27].

Tropos [28] is a methodology and graphical modelling language for development of agent-oriented systems. Like our extended language for legal risk analysis, actors and roles are central concepts in the Tropos language. However, unlike the CORAS language, which is asset-oriented and targeted at threat modelling, the Tropos language is goal-oriented and is targeted at modelling agents and their goals, plans, capabilities, beliefs, etc. In our future work we will investigate the need for incorporating concepts from languages such as TROPOS as well as other conceptual models such as the one presented in [29].

Acknowledgements

The research on which this paper reports has partly been carried out within the context of the EU-projects TrustCoM (IST-2003-01945) and iTrust (IST-2001-34910).

References

1. Jøsang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems (to appear) http://security.dstc.edu.au/papers/JIB2005-DSS.pdf.
2. Egger, F.N.: Towards a model of trust for e-commerce system design. In: CHI 2000: Workshop Designing Interactive Systems for 1-to-1 E-commerce. (2000) `http://www.zurich.ibm.com/~mrs/chi2000/contributions/egger.html`.
3. Jones, S., Wilikens, M., Morris, P., Masera, M.: Trust requirements in e-business. Communications of the ACM **43** (2000) 81–87
4. Wahlgren, P.: Juridisk riskanalys - Mot en säkrare juridisk metod. Jure, Stockholm (2003) (In Swedish).
5. Susskind, R.: The Future of Law. Clarendon Press, Oxford (1996)
6. Reidenberg, J.: Lex Informatica: The Formulation of Information Policy Rules Through Technology. In: Texas Law Review. Volume 76. (1998) 553–593
7. CORAS: The CORAS project (2005) http://coras.sourceforge.net/ (visited February 2005).
8. Dimitrakos, T., Ritchie, B., Raptis, D., Aagedal, J.Ø., den Braber, F., Stølen, K., Houmb, S.H.: Integrating model-based security risk managament into eBusiness systems development: The CORAS approach. In: I3E2002, Kluwer (2002) 159–175
9. Raptis, D., Dimitrakos, T., Gran, B.A., Stølen, K.: The CORAS approach for model-based risk management applied to e-commerce domain. In: CMS-2002, Kluwer (2002) 169–181
10. OMG: UML 2.0 Superstructure Specification. (2004) OMG Document: ptc/2004-10-02.
11. Lund, M.S., Hogganvik, I., Seehusen, F., Stølen, K.: UML profile for security assessment. Technical Report STF40 A03066, SINTEF Telecom and informatics (2003)
12. OMG: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, Draft Adopted Specification (2004) OMG Document: ptc/2004-06-01.
13. TrustCoM: Trust and Contract Management in Virtual Organisations (2005) http://www.eu-trustcom.com/ (visited February 2005).
14. Redmill, F., Chudleigh, M., Catmur, J.: HazOp and software HazOp. Wiley (1999)
15. Chellas, B.F.: Modal Logic - An Introduction. Cambridge University Press, Cambridge, UK (1980)
16. Elgesem, D.: The Modal Logic of Agency. Nordic Journal of Philosophical Logic **2** (1997)
17. Brændeland, G., Stølen, K.: Using risk analysis to assess user trust - a net-bank scenario. In: Proceedings of 2nd International Conference on Trust Management (iTrust 2004). Volume 2995., LNCS, Springer (2004) 146–160
18. den Braber, F., Lund, M.S., Stølen, K.: Using the CORAS Threat Modelling Language to Document Threat Scenarios for several Microsoft relevant Technologies. Technical Report STF90 A04057, SINTEF ICT (2004)
19. Berardi, D., Calì, A., Calvanese, D., De Giacomo, G.: Reasoning on UML Class Diagrams. Technical Report 11-03, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza" (2003)
20. Haugen, Ø., Husa, K.E., Runde, R.K., Stølen, K.: Why timed sequence diagrams require three-event semantics. To appear in LNCS (2005)
21. ISO/IEC: FCD 15414: Information Technology - Open Distributed Processing - Reference Model - Enterprise Viewpoint. JTC1/SC7 N2359, ISO/IEC (2000)

22. Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The Ponder Specification Language. In: Workshop on Policies for Distributed Systems and Networks, Bristol, UK (2001)
23. OASIS: eXtensible Access Control Markup Language (XACML) Version 1.0. Technical report, OASIS (2003)
24. Chu, Y.H., Feigenbaum, J., LaMacchia, B., Resnick, P., Strauss, M.: Referee: Trust management for web applications. In: Sixth International World Wide Web Conference, Santa Clara, CA, USA (1997)
25. Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.D.: The KeyNote Trust Management System, Version 2. Request For Comments (RFC) 2704, AT&T Labs and University of Pennsylvania (1999)
26. Biskup, J., Karabulut, Y.: A Hybrid PKI Model with an Application for Secure Mediation. In: 16th Annual IFIP WG 11.3 Working Conference on Data and Application Security, Cambridge, England, Kluwer Academic Press (2002) 271–282
27. PERMIS: Privilege and Role Management Infrastructure Standards Validation (2004) http://sec.isi.salford.ac.uk/permis/ (visited December 2004).
28. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A.: TROPOS: An Agent-Oriented Software Development Methodology. In: Journal of Autonomous Agents and Multi-Agent Systems. Volume 8., Kluwer Academic Publishers (2004) 203–236
29. Sagri, M.T., Tiscornia, D., Gangemi, A.: An ontology-based model for Representing "Bundle-of-rights". In: The second International Workshop on Regulatory Ontologies (WORM 2004). LNCS, Larnaca, Cyprus, Springer (2004)