A UML-based Method for the Development of Policies to Support Trust Management

Atle Refsdal, Bjørnar Solhaug, and Ketil Stølen

Abstract Most of the existing approaches to trust management focus on the issues of assessing the trustworthiness of other entities and of establishing trust between entities. This is particularly relevant for dynamic, open and distributed systems, where the identity and intentions of other entities may be uncertain. These approaches offer methods to manage trust, and thereby to manage risk and security. The methods are, however, mostly concerned with trust management from the viewpoint of the trustor, and the issue of mitigating risks to which the trustor is exposed. This paper addresses the important, yet quite neglected, challenge of understanding the risks to which a whole system is exposed, in cases where some of the actors within the system make trust-based decisions. The paper contributes by proposing a method for the modeling and analysis of trust, as well as the identification and evaluation of the associated risks and opportunities. The analysis facilitates the capture of trust policies, the enforcement of which optimizes the trust-based decisions within the system. The method is supported by formal, UML-based languages for the modeling of trust scenarios and for trust policy specification.

1 Introduction

When the term trust management was introduced in 1996 [3] it basically referred to the management of authorizations and access rights in distributed systems. Since then, trust management has been subject to increased attention and has more re-

Atle Refsdal
SINTEF ICT, e-mail: Atle.Refsdal@sintef.no

Bjørnar Solhaug
Dep. of Information Science and Media Studies, University of Bergen and SINTEF ICT, e-mail: Bjornar.Solhaug@sintef.no

Ketil Stølen
SINTEF ICT and Dep. of Informatics, University of Oslo, e-mail: Ketil.Stolen@sintef.no

Please use the following format when citing this chapter:

Refsdal, A., Solhaug, B. and Stølen, K., 2008, in IFIP International Federation for Information Processing, Volume 263; *Trust Management II*; Yücel Karabulut, John Mitchell, Peter Herrmann, Christian Damsgaard Jensen; (Boston: Springer), pp. 33–49.

cently been described as an activity "in the intersection between sociology, commerce, law and computer science" [8].

Whatever the approach to or domain of trust management, a fundamental issue is to assess the trustworthiness of other entities and to make decisions based on these assessments. Trust is a relationship between two entities, a trustor and a trustee, and is associated with a particular transaction between these entities. The trustor is the stakeholder in the relationship, and the participation in the transaction is motivated by the opportunities involved in the transaction. Trust is, however, inherently related to risk since there always is a chance of deception or betrayal [16].

In this paper we propose a UML-based method for the development of policies to support trust management. The method goes through three main stages: (1) System modeling, (2) trust analysis, and (3) trust policy specification. The trust analysis should result in an overview of the available choices and the associated risks and opportunities. On the basis of this overview, a trust policy is formalized, the enforcement of which ensures that the most beneficial choices are made.

The next section describes the challenges addressed by this paper. By defining the basic concepts of our approach to trust management and explaining the relations between these concepts, we motivate the various steps and the ultimate goal of the proposed method. A motivating example used to illustrate the method throughout the paper is introduced. We also define a set of success criteria that should be fulfilled by the proposed method. An overview of the method is given in Section 3, followed by an example-driven description of the three main steps of the method in the subsequent sections. Firstly, Section 4 shows the use of Subjective STAIRS [12] to model the target of analysis. Secondly, Section 5 employs the models in Subjective STAIRS to analyze and evaluate the relevant trust relationships. Thirdly, Section 6 shows the use of Deontic STAIRS to specify the trust policy resulting from the analysis. Deontic STAIRS, as well as Subjective STAIRS, are based on UML 2.1 [10] sequence diagrams and STAIRS [6]. Subjecitive STAIRS is furthermore also based on Probabilistic STAIRS [11]. The approach is discussed and evaluated against the success criteria in Section 7, before we conclude in Section 8.

2 The Challenge

The overall challenge addressed by this paper is to establish a method to correctly assess trust and analyze trust-based transactions in order to identify, analyze and evaluate the involved risks and opportunities. The evaluation should result in a trust policy the enforcement of which ensures that risks are minimized and opportunities maximized.

A typical target of evaluation is an enterprise, system or organization in which there are actors whose choices of action may be based on trust. As an example, we consider a local bank and the risks and opportunities involved in loan approval to customers. The evaluation is from the perspective of the bank as a stakeholder, where the main asset of the bank is its revenue. In order to properly identify and

assess the involved risks and opportunities, the basis upon which the bank employees grant or reject loan applications must be well understood. To keep the example simple while still illustrating the essential aspects of the approach, we make the following four assumptions: (1) An application for a loan includes the amount a that the customer wants to borrow. Other information, such as the value of the applicant's properties, or other loans the applicant may have, are not considered; (2) The customer either pays back the full loan (including interest), or nothing at all; (3) There is no mortgage securing the loan; (4) If the customer pays back the loan, then the bank's revenue v will increase by a gain g, otherwise it will decrease by a.

In many cases it is obvious whether or not applications should be accepted, typically when the income of the applying customer is very low or very high compared to the loan amount. In this paper we focus on the cases where there might be some doubt or uncertainty with respect to the ability of the customer to repay the loan, and where the decision as to accept an application is made by the individual bank employee. In these cases the level of trust of the employee in the customer may be decisive. Clearly, if the bank employee makes a wrong decision, then money may be lost; either because a loan is granted to a customer who does not pay back, or because a loan is not granted to a customer who would have paid back. The management wishes to develop a policy to ensure that the best possible decisions are made.

Consider first the potential risk involved in loan approval. A risk is defined as the probability of the occurrence of a harmful event [7], i.e. an event with a negative impact on an asset. The harmful event in the bank scenario is that a customer fails to repay the loan, and the impact of this event on the bank's asset is the loss of the loan sum a. The level of risk is given as a function from the consequence (loss) of the harmful event and the probability of its occurrence [1]. If the probability of this event is $p \in [0,1]$, the risk level is $R(p,a)$ for a given risk function R. For sake of simplicity, the risk function is in this paper defined to be multiplication, so $R(p,a) = p \cdot a$.

The dual to a risk is an opportunity, which is defined as the probability of the occurrence of a beneficial event, i.e. an event with a positive impact on an asset. In the bank scenario, the customer having paid all installments represents an opportunity. The positive impact is the gain for the bank, which depends on the loan amount and the interest rate. The opportunity level is given as a function O from the gain, say g, and the probability p for repayment. We use multiplication as the opportunity function also, so $O(p,g) = p \cdot g$.

In cases of doubt, the bank employee must consider the information available on the loan applicant. This may concern job affiliation, age, marital status, previous late settlements of debts, etc. This information may be incomplete or even false, but still a decision has to be made. In such a situation of uncertainty, other factors may also be considered, e.g. the personality of the customer, the impression the customer makes, and even acquaintance if it is a small, local bank. In such cases the trust of the bank employee in the customer may be decisive.

Our notion of trust is based on the definition proposed by Gambetta [5] and defined as the subjective probability by which an actor (the trustor) expects that

another entity (the trustee) performs a given action on which the welfare of the trustor depends.

So trust is a probability estimate that ranges from 0 (complete distrust) to 1 (complete trust). It is subjective, which means that it is a belief that may be wrong. The welfare of a trust relation refers to an associated asset of the trustor. If the trustee performs as expected, it will have a positive outcome for the trustee. There is, however, always the possibility of deception, and in that case there will be a negative impact on the welfare. Hence, trust is related to both opportunity and risk.

For the bank being the stakeholder in our example case, it is important to evaluate the effects of the trust-based decisions of the employees in terms of risks and opportunities for the bank. If, for example, an employee has trust $p \in [0, 1]$ in that a given customer will repay a loan of the amount a with a potential gain g, the opportunity is given by $p \cdot g$ as believed by the bank employee. The employee furthermore believes that the risk is $(1 - p) \cdot a$. Decisions are then made by comparing the risk and opportunity. If there is a difference between the trust value and the actual trustworthiness of the customer, the wrong decision may be made.

By trustworthiness we mean the objective probability by which the trustee performs a given action on which the welfare of the trustor depends. Well-founded trust is the case in which trust equals the trustworthiness, and it is only in this case that the involved risks and opportunities are correctly estimated.

If trust is ill-founded, the subjective estimate is either too high or too low. In the former case we have misplaced trust, which is unfortunate as it means that the believed risk level is lower than the actual risk level. In the latter case we have misplaced distrust, which is also unfortunate since then the believed risk level is higher than the actual one, which may lead to valuable transactions being missed.

In existing literature on the subject, trust management is mostly concerned with approaches and methods aimed to support the trustor in making assessments about trustworthiness of other parties. The challenge addressed by this paper is the analysis of the risks and opportunities to which a system is exposed as a result of choices of behavior of entities within the system, where these choices may be based on trust. Some of the entities within the system are subjective entities, and the challenge is to reach an objective understanding of the system as a whole. Moreover, based on this objective understanding, the challenge is also to gain a correct estimation of the risks and opportunities imposed by subjective decisions. Hence, we aim for a method to identify and analyze trust, and thereby capture a policy to avoid risks and seek opportunities, as further explained by the six success criteria described in the following.

The aim of any analysis is to reach an objective understanding of the target of analysis. In this case the target contains actors of a subjective nature, but the challenge is still to reach an objective understanding of the target as a whole: Does the target of analysis function as it should? What is the impact of subjective decisions on the overall behavior? Therefore: *(C1) The method should facilitate the objective modeling of systems whose overall behavior depends on subjective, trust-based behavior of actors within the system.*

Trust is a subjective probability, and in order to properly specify trust relations, the modeling of trust levels of the relevant trust relations must be supported. Therefore: *(C2) The method should facilitate the specification of the level of trust an actor has in another entity with respect to a given transaction.*

The ultimate goal of trust management is to minimize risks and maximize opportunities related to trust. This gives: *(C3) The method should facilitate the identification, estimation and evaluation of system risks and opportunities imposed by the relevant trust relations.*

Through trust modeling and the analysis of risks and opportunities, trust-based choices of behavior that should be avoided are identified, as well as trust-based choices that should be sought. A policy is a set of rules that govern choices in system behavior, and the method should identify the rules that ensure the most advantageous system behavior. In short: *(C4) The method should support the capturing of an adequate trust policy.*

Policy enforcement is facilitated by precise descriptions of the policy rules. Both obligation and prohibition rules should be supported so as to define absolute choices of behavior. In case of choices between potential alternatives that are considered equivalent with respect to a given purpose, permission rules must be expressible. The rules of a trust policy should be specified with triggers that define the circumstance, as well as the levels of trust, under which the rule applies. Hence: *(C5) The method should have sufficient expressiveness to capture obligations, prohibitions and permissions, as well as triggers where required.*

In order to develop a good policy, it is essential that decision makers, developers, analysts, etc. have a clear and shared understanding of the system, the relevant scenarios, and the alternative policy rules. Moreover, the policy rules must be easily understandable for those who are supposed to adhere to them. *(C6) The method should offer description techniques that are understandable to all relevant stakeholders, including end-users, decision makers and engineers.*

3 Overview of Method

In this section we give an overview of the three main steps of the method and motivate its overall structure. The sections thereafter demonstrate the method on the bank example.

Step 1. Modeling of Target. In order to analyze something, we need an understanding of this "something" at a level of abstraction that is suitable for the analysis. Abstraction is necessary since most systems are extremely complex when all details are taken into consideration. For example, our bank example involves human beings, and nobody would even consider to describe a human being in full detail. In order to document the sought understanding and validate it on others, it seems reasonable to make use of a modeling language.

This description of the target should not only show how the actors and components behave, but also what decisions and choices are made by actors in the system, and *why* those decisions and choices are made. More specifically, as our purpose here is to develop a trust policy, we are interested in understanding what decisions are taken on the basis of trust, and what considerations lie behind such decisions.

Subjective STAIRS, which is a language based on UML 2.1 sequence diagrams, has been selected for this purpose. It allows us to specify subjective beliefs about scenarios, as well as actual (objective) scenarios, and also to show how the subjective beliefs influence the choices made by actors. Subjective STAIRS distinguishes between subjective and objective diagrams, and probability may be expressed in both kinds. Trust with respect to a transaction is represented by a probability in a subjective diagram.

We use objective diagrams to capture the actual behavior of the target, while subjective diagrams are employed to express the belief of an actor with respect to a scenario. In Section 4 we demonstrate the use of Subjective STAIRS to describe the banking system as the target of analysis.

There are two main reasons for basing the modeling language on UML sequence diagrams. Firstly, sequence diagrams are well suited to express interactions between entities. As trust is relevant in the context of an interaction between the trustor and the trustee, this makes sequence diagrams a suitable choice. Secondly, sequence diagrams allow systems to be described at a high level of abstraction, in a simple and intuitive manner that can be understood by stakeholders with different background and level of training. These qualities are important in the context of the risk and opportunity analysis that we must conduct in order to develop trust policies.

Step 2. Analysis of Target. After obtaining a suitable description of the target, the next task is to perform an analysis. The analysis proceeds in four sub-steps as described in the following and demonstrated in Section 5 with the banking system as target.

Step 2.1. Identify critical decision points. First, the critical decision points that need to be looked into are identified. Typically, this will be points where decisions are made based on trust. But it may also be points where one could potentially benefit from introducing new trust-based decisions, if the resulting opportunities outweigh the risks.

Step 2.2. Evaluate well-foundedness of trust. Second, we need to evaluate the well-foundedness of the trust on which decisions and choices are based. As trust is a subjective probability estimate, this amounts to finding out to what degree the subjectively estimated probabilities correspond to the actual (objective) probabilities.

Step 2.3. Estimate impact of alternative behavior. Of course, it may well be that the current way of making choices is not optimal. Therefore, the third sub-step is to estimate what would be the impact of other, alternative choices of behavior.

Step 2.4. Evaluate and compare alternative behavior. The final sub-step consists of an evaluation and comparison of alternative behaviors, with the aim to identify the behaviors that should be sought or avoided.

Step 3. Capturing a Policy to Optimize Target. Having identified the desirable behavior, we are finally ready to specify a policy the enforcement of which ensures the optimal choices of behavior.

A policy is a set of rules that determines choices in the behavior of a system [14], and is used in policy based management. Typical domains are security management and the management of networks and services. The method proposed by this paper is an approach to policy based trust management. Each rule determines a system choice of behavior, where a given trust level is a decisive factor for each choice. Enforcement of the given rules ensures the optimal level of the risks and opportunities that are imposed by trust based decisions within the system. As for the target model, it is essential that the policy is unambiguous and understandable for all involved stakeholders. To formalize the policy we use Deontic STAIRS, which is a language for expressing policies, and based on UML sequence diagrams. Employing sequence diagrams as the basis for all modeling, analysis and specification in the method is desirable, both because it facilitates use and understandability, and because specifications can be reused.

Deontic STAIRS has the expressiveness to specify constraints in the form of obligations, prohibitions, and permissions, corresponding to the expressiveness of standard deontic logic [9]. Such constraints are normative rules that describe the desired system behavior. This reflects a key feature of policies, namely that they "define choices in behavior in terms of the conditions under which predefined operations or actions can be invoked rather than changing the functionality of the actual operations themselves" [15]. Furthermore, Deontic STAIRS supports the specification of triggers that define the circumstances under which the various rules apply. In particular, the policy triggers can specify the required trust levels for a particular choice of behavior to be constrained.

4 Modeling the Bank System

We now show how the system to be analyzed is modeled in Subjective STAIRS, focusing only on the case where the bank employee makes a decision based on trust. Subjective STAIRS builds on Probabilistic STAIRS, which has a formal semantics. However, for our purposes here, an intuitive explanation of the diagrams suffices.

Briefly stated, the scenario is as follows: First the customer applies for a loan. The bank employee then grants the loan if she or he believes that the probability of the loan being paid back is sufficiently high. Otherwise, the application is rejected. In the cases where the loan is granted, one of two things may happen: either the customer pays back the loan (with interest), so that the bank's asset value increases, or the bank has to write off the loan, in which case the asset value decreases. The model is given in Fig. 1. The main diagram is `loan1`, which is an objective diagram showing the actual behavior of the system. Each of the entities taking part in the interaction is represented by a dashed, vertical line called a lifeline, where the box at the top of the line contains the name of the entity, in this case the bank employee

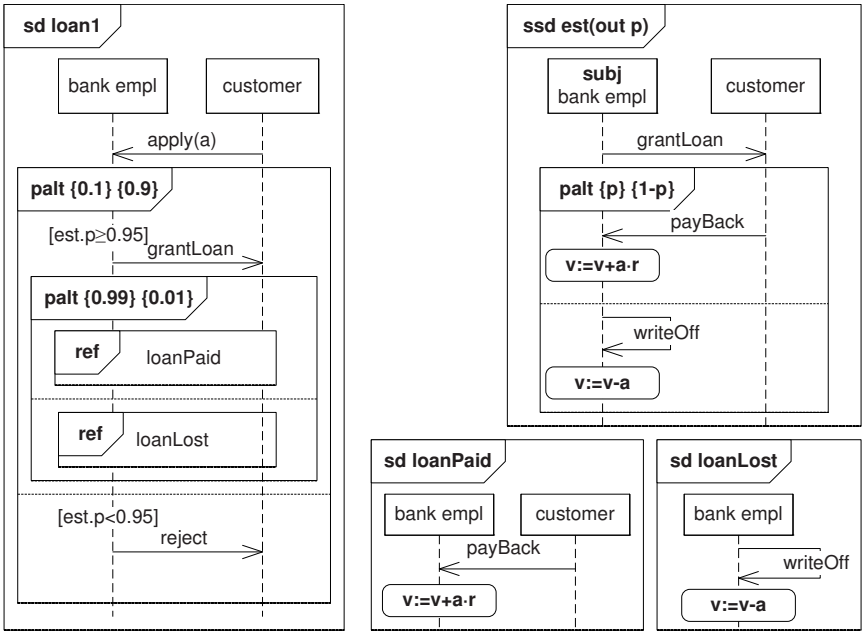

Fig. 1 System model

(bank empl) and the customer (customer). The interaction between the entities are represented by messages, which are shown as horizontal arrows from one lifeline to another (or to itself). Each message defines two events: the transmission of the message, represented by the arrow tail, and the reception of the message, represented by the arrow head. Transmission naturally occurs before reception, and events are ordered from the top on each lifeline, so the first thing that happens is that the customer sends the apply(a) message to the bank employee. This message represents the application, where a is the amount applied for.

At this point, the scenario may continue in one of two alternative ways, as the application may be either granted or rejected. These alternatives are represented by the outermost palt operator. The palt operator expresses alternatives with probabilities. The operator is shown as a frame with the operator name (palt) in the upper left corner. Its operands are separated by a horizontal dashed line. Each operand represents an alternative. The numbers occurring after palt in the upper corner of the operator frame show the probabilities of the operands; the probability for the first (upper) alternative to occur is 0.1, while the probability for the second (lower) is 0.9. As this is an objective diagram, we may imagine that the probabilities have been obtained for example by observing the system for a while and registering frequencies for the alternative behaviors.

At the beginning of each operand, we find a Boolean expression enclosed in square brackets. These constructs, which are called guards, constrain the conditions under which the alternatives may occur; an alternative occurs only if its guard eval-

uates to true. The expression est.p in the guards of both operands of the outermost palt represents the probability subjectively estimated by the bank employee that the customer will pay back the loan if the loan is granted, as will be further explained below. This means that the first alternative, where the loan is granted, occurs only if the bank employee believes that the probability that the loan will be paid back is at least 0.95. The fact that the first alternative has probability 0.1 means that the bank employee holds this belief in 10% of the cases where a loan is applied for.

Assuming the bank employee estimates that the probability of the loan being paid back is at least 0.95, she or he grants the loan, as represented by the grantLoan message. Then we have again two possible alternatives[1]. The first alternative, which has probability 0.99, is that the customer pays back the loan, represented by the payBack message. Notice that there is no assumption about the time interval between events on a lifeline, so the interval between granting the loan and being paid back may be much longer than the time between receiving the application and granting the loan. After the loan is paid back, the bank's asset value *v* increases by the amount *a* multiplied by the interest rate *r*. This is represented in the diagram by the assignment statement v:=v+a·r.

The second alternative, which has probability 0.01, is that the bank employee decides that the money will have to be written off, as the customer will not pay back the loan. This decision is represented by the writeOff from the bank employee to her/himself. In this case the bank's asset value decreases by the amount *a*, as represented by the assignment statement v:=v-a.

As noted above, the expression est.p represents the probability subjectively estimated by the bank employee that the customer will pay back the loan if the loan is granted. This can be seen from the diagram est in the upper right-hand corner of Fig. 1, which is a subjective diagram representing the belief of the bank employee. Subjective diagrams have the keyword ssd (for subjective sequence diagram) instead of sd in front of the diagram name. In addition, exactly one lifeline is decorated with the keyword subj, indicating that this is the actor (subject) whose belief is represented by the subjective diagram in question. For the est diagram, the bank employee is the subject. The probabilities in the est diagram are given in terms of the symbolic value p rather than a number, as the probability estimate will vary depending on the customer. The statement out p after the diagram name means that the symbolic value p can be referred to from an objective diagram by the expression est.p, as is done in the guards of loan1.

[1] For these alternatives we have made use of the UML ref construct. This construct is a reference to the diagram whose name occurs in the frame. Its meaning is the same as if the content of the referenced diagram was inserted in place of the ref construct. The ref construct allows a modular presentation of diagrams, as well as reuse of diagrams.

5 Analyzing the Bank System

In this section we demonstrate the analysis method presented in Section 3 on the bank example by going through the four (sub-)steps of the analysis step.

Step 2.1. Identify critical decision points. For this example, we assume that the only critical decision point we want to consider is the choice between granting or rejecting the application. Other decision points could also have been considered. For example, the bank employee could decide whether further investigations into the customer's previous history are needed before deciding whether to grant the loan, or whether another attempt at demanding payment should be made before writing off the money.

Step 2.2. Evaluate well-foundedness of trust. For the second analysis step, we need to find out whether the trust is well-founded, i.e. to what degree the subjectively estimated probabilities correspond to the objective probabilities. In order to do this, we need a model that describes what happens if the bank employee grants *all* applications. This is necessary in order to evaluate the correctness of the probability estimates also for the cases where the application would normally be rejected. The diagram loan2 in Fig. 2 provides such a model[2]. How the model is obtained is up to the analysis team. It could for example be based on some expert's opinion, historical data, simulation, or be the result of an experiment where all applications are granted for a certain period of time (although the latter is perhaps not likely for this particular example).

In addition to assuming that all applications are granted, we have chosen to distinguish between four different intervals of subjectively estimated probability, as can be seen from the guards in loan2. The number of intervals can be chosen freely by the analysis team, depending on the desired granularity of the analysis. For each interval we have a separate palt operand. This allows us to compare the actual (objective) probability with the interval in which the subjective estimate lies. The first palt operand in loan2 represents the cases where the estimate lies within the interval $[0.95, 1]$, which according to the probability of the first operand happens in 10% of the cases (as the probability for this operand is 0.1). From granted1 referred to in the first operand, we see that the probability of being paid back in this case is 0.99. The second palt operand in loan2 represents the cases where the estimate lies within the interval $[0.9, 0.95\rangle$, which happens in 20% of the cases. The probability of being paid back in these cases is 0.96, as seen from granted2. The third palt operand in loan2 represents the cases where the estimate lies within the interval $[0.8, 0.9\rangle$. For these cases, the probability of being paid back is in fact 0.92, which is slightly outside the estimated interval. Finally, the fourth palt operand in loan2 represents the cases where the estimate is lower than 0.8, and the probability of being paid back in these cases is 0.6 according to granted4.

Step 2.3. Estimate impact of alternative behavior. In addition to showing the correspondence between subjective estimates and objective probabilities, loan2 also describes the overall system behavior resulting from using alternative thresholds

[2] The references loanPaid and loanLost are to the diagrams in Fig. 1.

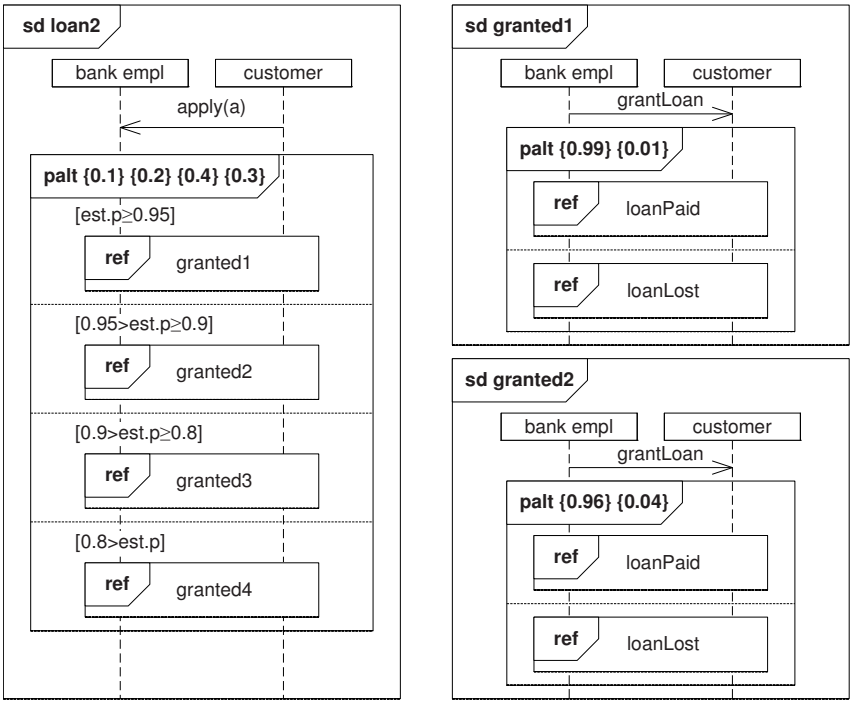

Fig. 2 System model obtained in the analysis in order to evaluate alternatives. The diagrams granted3 and granted4 are omitted. They are identical to the granted1 and granted2 diagrams, except from the probabilities in the palt operand. For granted3, the probabilities are 0.92 for the first operand, and 0.08 for the second operand. For granted4, the probabilities are 0.6 for the first operand, and 0.4 for the second operand.

(against which the estimated probability is compared) when deciding whether to grant loans. Therefore, loan2 already represents an estimate of the impact of some alternative behavior, i.e. what is the impact of using different thresholds. The impact of other alternative behavior could also be considered in this step of the analysis. For example, what would be the impact of always performing investigations into the customer's previous history before deciding whether to grant the loan? However, due to lack of space, we consider only the alternative behavior already described in Fig. 2.

Step 2.4. Evaluate and compare alternative behavior. Table 1 presents the risks and opportunities on the basis of loan2. The numbers are calculated from an interest rate of 8.7%, i.e. $r = 0.087$. The "Threshold" column shows the decision threshold that is analyzed in the corresponding row. For example, the second row in the table shows the results of only granting a loan in cases where the estimated probability of being paid back is at least 0.9.

The "Granted" column shows the percentage of the received applications that will be granted when the relevant decision threshold is used. For example, if a loan

Table 1 The result of using alternative decision thresholds

Threshold	Granted	Paid back	Opp.	Risk	Opp.−Risk
≥ 0.95	10%	99%	0.0086	0.001	0.0076
≥ 0.9	30%	97%	0.025	0.009	0.016
≥ 0.8	70%	94%	0.057	0.041	0.016
≥ 0	100%	84%	0.073	0.16	−0.087

is granted if the estimated probability of being paid back is at least 0.9, then 30% of all applications will be granted. This number is obtained by adding the probabilities of all operands whose guard ensures that the estimated probability is not lower than the decision threshold. For the case where the threshold is 0.9, we add up the probabilities of the two first operands of the palt in the loan2 specification. Thus we obtain 0.3, which corresponds to 30%.

The "Paid back" column shows the percentage of the granted loans that will be paid back. This number is obtained by, for each operand where a loan will be granted, taking the product of the probability of the operand and the probability of being paid back according to this operand. The sum of these products is divided by the percentage of applications being granted. For example, for the second row we get $(0.1 \cdot 0.99 + 0.2 \cdot 0.96)/0.3$, which gives 0.97, i.e. 97%.

The "Opp." column shows the opportunity value we get by choosing the relevant threshold value. The opportunity value is obtained from the formula $0.087 \cdot \sum(p_1 \cdot p_2)$, where $\sum(p_1 \cdot p_2)$ is the sum of the product of the probability p_1 of the operand and the probability p_2 of being paid back for all operands where a loan will be granted when the threshold in the left-hand column is used. For example, for the second row in the table, we get $0.087 \cdot (0.1 \cdot 0.99 + 0.2 \cdot 0.96) = 0.025$. The loan amount a has been factored out, as this would occur only as a common factor in all rows, and we are only interested in the relative values.

The "Risk" column shows the accumulated risk value we get by choosing the relevant threshold value. It is obtained in a similar way as the accumulated opportunity value, except that we use the probability of *not* being paid back for each operand, and that the interest rate is not taken into account. For the second row in the table, we get $(0.1 \cdot 0.01 + 0.2 \cdot 0.04) = 0.009$.

Finally, the "Opp.−Risk" column shows the difference between the two previous columns. From Table 1 we are now able to compare the alternative thresholds in order to decide which of them should be used in order to achieve the most desirable system behavior. The goal of the bank here is to maximize the difference between opportunity and risk, i.e. to achieve the highest possible value for "Contr.-Risk". This goal may or may not coincide with the goals of the bank employee, which has different assets from the bank. However, as the analysis is performed on behalf of the bank, we do not consider the goals of the bank employee.

From the two first rows in the "Opp.-Risk" column we see that the value increases if the threshold is lowered from ≥ 0.95 to ≥ 0.9, i.e. if a loan is granted also in cases where the estimated probability of being paid back is between 0.9 and 0.95, instead of only when the estimated probability is at least 0.95. Even if slightly more of the

customers that are granted a loan will not be able to pay back, this is more than made up for by the increased number of customers that will pay back their loan with interest. In other words, the bank loses money by not granting loans in cases where the subjective estimate lies between 0.9 and 0.95. Therefore, the bank should enforce the following trust policy rule: *(R1) It is obligated to grant a loan for the cases in which the trust level is at least 0.9*.

The second and third rows show that the net effect of lowering the threshold one step further, to ≥ 0.8, is zero, as the "Opp.-Risk" value is the same for the thresholds ≥ 0.9 and ≥ 0.8. This means that the bank will neither lose nor gain from granting loans also in the cases where the estimated probability of being paid back is between 0.8 and 0.9. Therefore, the bank may decide to leave the choice up to the individual bank employee. This gives the following two policy rules: *(R2) It is permitted to grant a loan for the cases in which the trust level is lower than 0.9 but at least 0.8*; *(R3) It is permitted to reject a loan for the cases in which the trust level is lower than 0.9 but at least 0.8*.

The fourth row in the "Opp.−Risk" column, however, shows that granting loans to all applicants will give a risk that is higher than the opportunity. Therefore, the following rule should be enforced: *(R4) It is prohibited to grant a loan for the cases in which the trust level is lower than 0.8*.

6 Policy to Optimize the Bank System

We now show how we may use Deontic STAIRS to formalize the optimized strategies we arrived at in Section 5 as a policy. Deontic STAIRS is an extension of UML 2.1 sequence diagrams and is underpinned by a formal semantics based on the denotational semantics of STAIRS. For the purpose of this paper it suffices to explain the semantics informally.

The analysis of the bank system in the previous section revealed that loan application should be granted in the cases where the trust of the bank employee in that the customer will repay the loan is 0.9 or higher, as captured by rule *(R1)* above. This is expressed by the obligation rule r1 specified to the left in Fig. 3.

The keyword rule in the upper left corner indicates that the diagram specifies a policy rule. The diagram consists of two parts, a trigger and an interaction that is the operand of a deontic modality. The trigger specifies the circumstances under which the rule applies and consists of an event and a condition. The former refers to an event such that when it occurs, the rule applies. In this case the event is the reception by the employee of a loan application. The condition of the trigger limits the applicability of the rule to a set of system states. In this case it refers to the states in which the relevant trust level is 0.9 or higher.

The keyword obligation shows the modality of the rule. It is an operator the operand of which specifies the behavior that is constrained by the rule. In this case, the relevant behavior is the granting of a loan. For simplicity, we model this by a

single message only, but in the general case the behavior can be described in any detail with all the expressiveness of UML 2.1 sequence diagrams.

A further result of the analysis in Section 5 is rule *(R4)*, namely that loan should not be granted to customers whose trust value is lower than 0.8, since in that case the risk is higher than the opportunity for the bank. This is captured by diagram r4 to the right in Fig. 3, where the keyword prohibition indicates the modality of the rule.

Obligations and prohibitions specify behavior that must and must not occur, respectively. Permissions, on the other hand, define choices of behavior that should be offered potentially, without disallowing alternative choices to be made instead. That is to say, permissions specify behavior that may occur. The above analysis of the bank system showed that for trust levels in the interval from 0.8 to 0.9, the involved risk and opportunity even out. This means that from the point of view of the bank, both loan approval and rejection are acceptable choices of behavior, i.e. both choices may be permitted.

As indicated by the modality, the rule r2 to the left in Fig. 4 is a permission. It specifies *(R2)* as captured in the analysis, and states that for the mentioned interval of trust levels, the bank employee is permitted to grant the applied loan to the customer. This means that the bank system must always allow the employee to make this choice in case the trigger holds. The employee is, however, free to make any other choice that does not conflict with other policy rules. In order to ensure that a set of potential alternatives should be available, other permissions with the same trigger can be specified. This is exemplified with the permission to the right in Fig. 4, expressing the rule *(R3)* captured in the analysis.

7 Discussion

Ever since trust management became a topic of research, the focus has mostly been on assessing trustworthiness of other entities and establishing trust between entities, as witnessed by a recent survey of state-of-the-art approaches to trust management

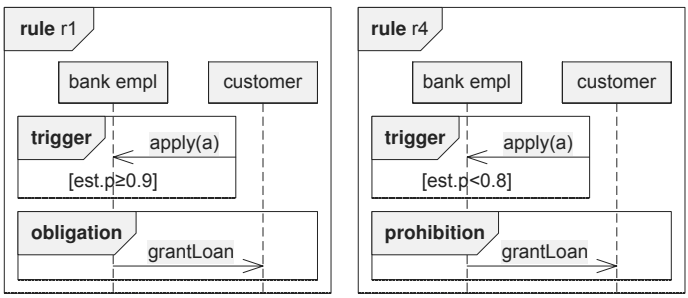

Fig. 3 Obligation and prohibition

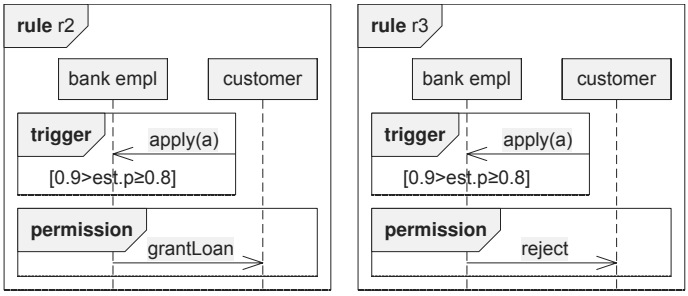

Fig. 4 Permissions

[13]. This is also put forward in [8], where trust management is discussed as an independent research topic and existing methods are reviewed.

Trust management is particularly relevant for distributed systems where the identity and/or intentions of other entities may be uncertain, and it is crucial to develop methods for managing trust, and hence managing risk and security. Importantly, most of the existing approaches focus on trust management from the point of view of the trustor, and the issue of mitigating risks to which the trustor is exposed. In this paper we have focused on the important challenge of understanding the risks to which a system is exposed, where some of the entities within the system are actors that make risk critical decisions based on subjective trust estimates. Moreover, whereas trust management traditionally focuses mostly on risk in relation to trust, we address also the challenge of identifying and evaluating the dual concept, namely opportunity. In the same way as risk is inevitably related to trust, so is opportunity, and in order to derive the most optimal trust policy for a given system, both risks and opportunities must be estimated.

Game theory [4] addresses strategies for describing rational choices in situations in which the outcome of a choice of one actor depends on the subsequent choice of another actor. A payoff structure describes the loss and gain to which the various players are exposed in each of the potential outcomes, and each player seeks the outcome with the most beneficial payoff for itself. Game theory can also be applied to analyze trust, as explained by e.g. Bacharach and Gambetta [2]. They show that the trustor's choice to trust or not, and the trustees subsequent choice to deceit or not, can be modeled in terms of this rational choice theory. The method presented in this paper captures aspects of game theory by the identification and modeling of choices of (trust-based) decisions, as well as the modeling of the associated payoff structure in terms of risks and prospects.

Subjective STAIRS has the expressiveness to model the alternative behaviors of a system, as well as the probabilities of these alternatives. Moreover, the notation supports the objective specification of the subjective probability estimates made by entities within the system, and thereby the trust on which these entities base their decisions. The trust management method proposed in this paper hence fulfills success criteria *C1* and *C2* as formulated in Section 2.

As demonstrated in Section 5, system modeling with Subjective STAIRS facilitates the identification, estimation and evaluation of both the risks and opportunities to which a system is exposed. Criterion *C3* is then also fulfilled by the method, thus allowing the relevant system analysis to be conducted. Through the analysis, the preferable choices of behavior are identified, which in turn facilitates the capture of an adequate trust policy. Hence, also criterion *C4* is fulfilled by the method.

Deontic STAIRS is a customized notation for policy specification, and as shown in Section 6, this notation supports the specification of trust policies. Behavior that should be sought and behavior that should be avoided may be expressed by obligations and prohibitions, respectively, whereas alternative choices of behavior that should be offered by the system may be formalized as permissions. Additionally, the circumstances under which a given rule applies can be expressed by the specification of a trigger consisting of an event and the level of trust of relevance for the policy rule in question. As demonstrated in Section 6, the proposed trust management method then fulfills success criterion *C5*.

The UML has the last decade or so emerged as the *de facto* standard for the specification of information systems. Both Subjective STAIRS and Deontic STAIRS are conservative extensions of the UML 2.1 sequence diagram notation, so people that are already skilled in the UML should be able to understand and use the languages employed by our method. Moreover, contrary to textual, more mathematical notations, the UML should be understandable also for people of non-technical background, at least with some guidance. Arguably, success criterion *C6* is then at least partly fulfilled by the proposed method.

8 Conclusion

This paper contributes by addressing the quite neglected challenge of understanding the risks and opportunities to which a system is exposed in cases where actors within the system make choices based on their trust in other entities. The proposed method offers languages and techniques for the modeling and analysis of the subjective notion of trust in an objective manner, facilitating the identification of risk and opportunity critical decision points. By identifying the set of behavioral alternatives of the target system and precisely estimating the involved risks and prospects, a policy that optimizes system behavior may be captured. The method furthermore contributes by facilitating the formalization of trust policies through an appropriate policy specification language.

Although refinement has not been an issue in this paper, it is certainly of relevance in relation to trust modeling and policy development, as it allows a system to be described, and hence analyzed, at different levels of abstraction. In the future we will address refinement in relation to the trust management method proposed in this paper by utilizing refinement techniques that have already been developed for Subjective STAIRS and Deontic STAIRS.

Acknowledgements The research on which this paper reports has partly been funded by the Research Council of Norway through the ENFORCE project (164382/V30) and partly by the European Commission through the S3MS project (Contract no. 27004) under the IST Sixth Framework Programme.

References

1. AS/NZS. *Australian/New Zealand Standard, AS/NZS 4360:2004, Risk Management*, 2004.
2. M. Bacharach and D. Gambetta. Trust in Signs. In K. S. Cook, editor, *Trust in Society*, volume II of *The Russel Sage Foundation Series on Trust*, pages 148–184. Russel Sage Foundation, 2001.
3. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 164–173, Oakland, CA, 1996.
4. D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.
5. D. Gambetta. Can We Trust Trust? In *Trust: Making and Breaking Cooperative Relations*, chapter 13, pages 213–237. Department of Sociology, University of Oxford, 2000. Electronic edition.
6. Ø. Haugen, K. E. Husa, R. K. Runde, and K. Stølen. STAIRS towards formal design with sequence diagrams. *Journal of Software and Systems Modeling*, 4:355–367, 2005.
7. ISO/IEC. *ISO/IEC 13335, Information technology – Guidelines for management of IT security*, 1996–2000.
8. A. Jøsang, C. Keser, and T. Dimitrakos. Can We Manage Trust? In *In Proceedings of the 3rd International Conference on Trust Management (iTrust)*, volume 3477 of *LNCS*, pages 93–107. Springer, 2005.
9. P. McNamara. Deontic Logic. In D. M. Gabbay and J. Woods, editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, pages 197–288. Elsevier, 2006.
10. Object Management Group. *Unified Modeling Language: Superstructure, version 2.1.1*, 2007.
11. A. Refsdal, R. K. Runde, and K. Stølen. Underspecification, inherent nondeterminism and probability in sequence diagrams. In *Proceedings of the 8th IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS)*, volume 4037 of *LNCS*, pages 138–155. Springer, 2006.
12. A. Refsdal and K. Stølen. Extending UML sequence diagrams to model trust-dependent behavior with the aim to support risk analysis. In *Proceedings of the 3rd International Workshop on Security and Trust Management (STM)*. ENTCS, to appear.
13. S. Ruohomaa and L. Kutvonen. Trust Management Survey. In *In Proceedings of the 3rd International Conference on Trust Management (iTrust)*, volume 3477 of *LNCS*, pages 77–92. Springer, 2005.
14. M. Sloman. Policy Driven Management for Distributed Systems. *Journal of Network and Systems Management*, 2:333–360, 1994.
15. M. Sloman and E. Lupu. Security and Management Policy Specification. *Network, IEEE*, 16(2):10–19, 2002.
16. B. Solhaug, D. Elgesem, and K. Stølen. Why Trust is not proportional to Risk. In *Proceedings of The 2nd International Conference on Availability, Reliability and Security (ARES)*, pages 11–18. IEEE Computer Society, 2007.