Electron Commer Res
DOI 10.1007/s10660-010-9063-z

Experiences from using a UML-based method for trust analysis in an industrial project on electronic procurement

Tormod V. Håvaldsrud · Olav S. Ligaarden · Per Myrseth · Atle Refsdal · Ketil Stølen · Jon Ølnes

Abstract This paper reports on experiences from using a UML-based method for trust analysis in an industrial project. The overall aim of the trust analysis method is to provide a sound basis for making trust policy decisions. The method makes use of UML sequence diagrams extended with constructs for probabilistic choice and subjective belief, as well as the capture of policy rules. The trust analysis method is evaluated with respect to a set of criteria. The industrial project focused on the modeling and analysis of a public electronic procurement (eProcurement) system

The research on which this paper reports has been carried out within the DIGIT project (180052/S10), funded by the Research Council of Norway, and the MASTER project, funded from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement FP7-216917.

T.V. Håvaldsrud · O.S. Ligaarden (✉) · A. Refsdal · K. Stølen
SINTEF ICT, Oslo, Norway
e-mail: olav.ligaarden@sintef.no

T.V. Håvaldsrud
e-mail: tormod.havaldsrud@sintef.no

A. Refsdal
e-mail: atle.refsdal@sintef.no

K. Stølen
e-mail: ketil.stolen@sintef.no

T.V. Håvaldsrud · O.S. Ligaarden · K. Stølen
Department of Informatics, University of Oslo, Oslo, Norway

P. Myrseth · J. Ølnes
DNV Research and Innovation, Høvik, Norway

P. Myrseth
e-mail: per.myrseth@dnv.com

J. Ølnes
e-mail: jon.olnes@dnv.com

Published online: 24 August 2010

making use of a validation authority service for validating electronic certificates and signatures.

Keywords Trust management · Modeling · Electronic certificates · Electronic procurement

1 Introduction

Trust is often linked to the notion of subjective probability. For example, inspired by [5, 10], [11] defines trust as the subjective probability by which an actor, the trustor, expects that another entity, the trustee, performs a given transaction on which its welfare depends. Unless you are a psychologist, subjective probabilities (or beliefs) are not very interesting as long as they are studied in isolation. In computer science we are interested in trust or the more general notion of belief only as long as it has an impact on the factual (or objective) behavior of a computer system or computer-based facility. Moreover, within computer science we are often more interested in the trust of an organization than the individual trust of a human being.

In order to analyze something we need a clear understanding of this "something" (or target) to be analyzed. This target is often captured in the form of a model. Unfortunately, modeling approaches of industrial maturity targeting the computer industry (like UML [13]) do not have the expressiveness required to fully cover the aspects of relevance for a trust analysis. This motivated us to develop a method for trust analysis based on UML sequence diagrams extended with constructs for capturing

1. beliefs of agents (humans or organizations) in the form of subjective probabilities;
2. factual (or objective) probabilities of systems which may contain agents whose behavior is described in terms of subjective probabilities;
3. trust decisions in terms of policy rules with the deontic modalities obligation, prohibition and permission.

This paper reports on experiences from using this method for trust analysis (first proposed in [17]) in an industrial project focusing on the modeling and analysis of a public electronic (eProcurement) system making use of a validation authority service for validating electronic certificates and signatures. The trust analysis was conducted on behalf of Det Norske Veritas (DNV) in the autumn of 2008. DNV's goal was to obtain a better understanding of the potential usefulness of a service they offered for supporting trust-based decisions in systems which rely on electronically signed documents. The performance of the trust analysis is evaluated with respect to a set of evaluation criteria.

The rest of the paper is organized as follows: Sect. 2 introduces our modeling approach which is UML sequence diagrams extended with constructs for probabilistic choice and belief. It also gives a brief introduction on how to specify trust policies to ensure and enforce the desirable behavior of a system. Section 3 presents the method for trust analysis, that builds on the modeling and policy specification approaches introduced in Sect. 2. Section 4 outlines the industrial case we used to test the feasibility of the trust analysis method. It also presents a set of evaluation criteria. Section 5

presents the use of the trust analysis method in the industrial case. Section 6 presents the results from the evaluation based on the criteria identified in Sect. 4. And finally, in Sect. 7 we draw the main conclusion and present related work.

2 Modeling approach

UML 2.1 sequence diagrams [13] are widely used for the modeling and specification of information systems; in particular to capture communication or interaction between system entities.

In this section we give a brief introduction to UML 2.1 sequence diagrams and the constructs for probabilistic choice and belief proposed in [18]. We also explain how sequence diagrams can be enriched to capture policy rules with deontic modalities. We use a running example: Alice purchases items on the Internet. For many of the purchases, Alice needs to send advance payment to the seller of the item. In these cases Alice runs a risk of not receiving the item after paying for it. The challenge is to model the trust considerations made by Alice, as well as their impact on the observable behavior.

2.1 Basic constructs as in the UML standard

The diagram **purchase** in Fig. 1 address the situation where Alice has found an item, with an acceptable price, that she might be interested in purchasing. The keyword `sd` (sequence diagram) in front of the diagram name marks the diagram as a sequence diagram. Each entity modeled by the diagram is represented by a dashed, vertical line called a lifeline, where the box at its top specifies which entity the lifeline represents, its name as well as its type separated by a colon. If one lifeline represents several entities with different names but of the same type, we only specify the type. Entities interact with each other through the transmission and reception of messages, which are shown as horizontal arrows from the transmitting lifeline to the receiving lifeline. For each message we distinguish between two events; a transmission event, represented by the arrow tail, and a reception event, represented by the arrow head. The transmission events occur, of course, before the corresponding receive events. The events on each single lifeline are ordered in time from top to bottom.

According to Fig. 1, Alice starts by requesting a tender from the seller. The seller sends in response a tender, signed with his electronic ID (eID) to Alice. Based on this signed tender, Alice decides whether she trusts the seller to send the item after she has sent the advance payment. Alice may behave in two alternative ways, with respect to this decision. She can either send the advance payment, or she can cancel the deal. This is represented by the outermost `alt` operator, which specifies alternative behavior. A dashed horizontal line separates the alternative behaviors. If Alice trusts the seller and sends the advance payment, the scenario continues in two alternative ways. This is represented by the innermost `alt` operator. Either the seller sends the item to Alice, or the seller does not. In the latter case Alice is forced to write off the money she paid for the item.

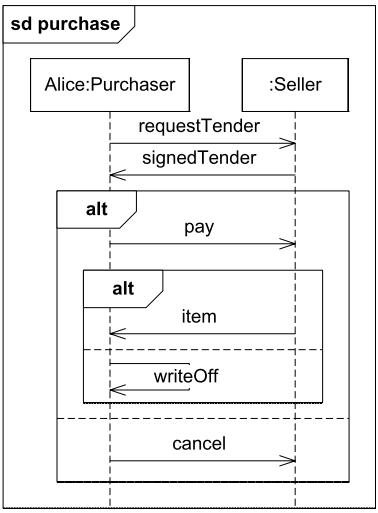

Fig. 1 The sequence diagram **purchase**

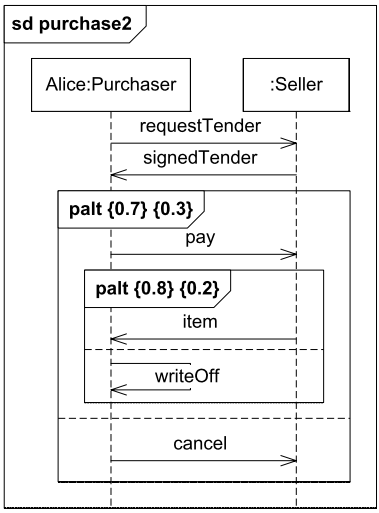

Fig. 2 The probabilistic sequence diagram **purchase2**

2.2 Construct for probabilistic choice

The diagram **purchase** in Fig. 1 specifies behaviors that may occur, but not how often or with what probability. Alice interacts with several sellers, and the scenario in Fig. 1 will therefore be repeated several times. We want to model how Alice behaves with respect to all these sellers. For this purpose we introduce the `palt` construct for probabilistic choice. By using the `palt` construct we can say something about the probability of a specific behavior.

The diagram **purchase2** in Fig. 2 is a probabilistic version of **purchase** in Fig. 1. The numbers occurring after `palt` in the upper corner of the operator frame specify the probabilities of the various alternatives. In the outermost `palt` the first number

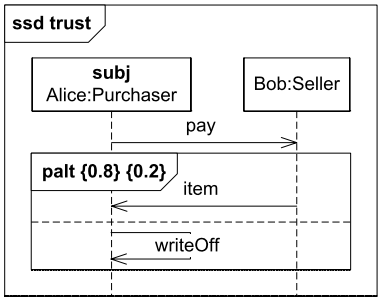

Fig. 3 The subjective sequence diagram **trust**

states that the scenario of the first operand occurs with a probability of 0.7, which means that this behavior occurs in 70% of the cases, while the second number states that the scenario of the second operand occurs with a probability of 0.3, which means that this behavior occurs in 30% of the cases. Furthermore, for the innermost `palt` operator the first operand has a probability of 0.8 of occurring, while the second operand has a probability of 0.2 of occurring. The probabilities in this diagram are factual in the sense that they are meant to reflect observable probabilistic behavior of the system.

2.3 Belief construct

It is clear that Alice behaves based on how much she trusts the seller of the item, but this notion of trust is not really reflected in the diagrams we have seen so far. They capture that she makes a choice, but not whether this choice is based on trust or something else. We want to model explicitly to what degree she needs to trust a seller before she sends advance payment.

Trust is the belief of a trustor that a trustee will perform a specific transaction on which the welfare of the trustor depends. Often the trustor will only expect the trustee to perform the transaction if another event has already occurred. In our example this event would be the sending of advance payment from Alice to the seller. Here, Alice is the trustor, while the seller is the trustee. Alice believes that there is a certain probability that the seller will send the item.

To model trust considerations we use so-called subjective sequence diagrams. Subjective sequence diagrams captures the subjective belief of an actor. Syntactically, subjective sequence diagrams differ from the ordinary sequence diagrams in two respects. Firstly, `ssd` (subjective sequence diagram) is used instead of `sd` to mark the diagram. Secondly, we annotate exactly one lifeline head with the keyword `subj`. This identifies the annotated entity as the subject, meaning that the diagram is used to capture this entity's subjective belief. According to the subjective sequence diagram **trust** in Fig. 3 Alice believes that the probability of receiving the item after sending the payment to the seller Bob is 0.8, and that the probability of not receiving the item is 0.2.

Semantically, a subjective sequence diagram aims to capture the belief of some entity like a person, an organization, or even a computer to the extent a computer may be said to believe. An ordinary sequence diagram, on the other hand, aims to capture

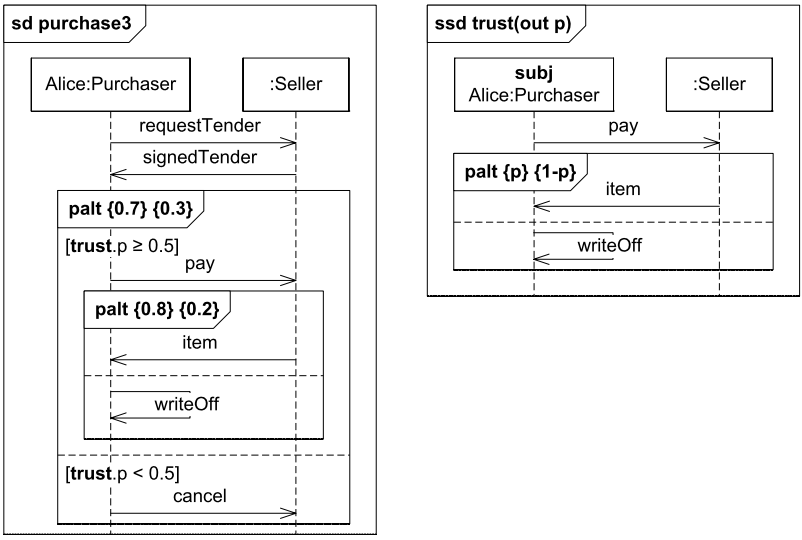

Fig. 4 The objective sequence diagram **purchase3** and the subjective sequence diagram **trust**

the factual reality, i.e. how things really are, independent of subjective beliefs, and such diagrams are in the following often referred to as objective sequence diagrams.

2.4 Combining objective and subjective diagrams

In the previous section we showed how we can model Alice's trust in a seller using subjective sequence diagrams. In this section we explain how subjective diagrams relate to the objective ones. Alice will only send payment if her trust in the seller is sufficiently high, i.e. if it reaches a certain threshold. The threshold says how much trust Alice needs to have in the seller in order to send him advance payment. In the diagram **purchase3** in Fig. 4, the threshold is represented as guards. A guard is a Boolean expression within square brackets. It constrains the choice of operand. An operand can only be chosen if its guard evaluates to true. We can see that the two guards refer to the variable `trust.p`. Here, `trust` refers to the subjective diagram **trust** in Fig. 4. Unlike the diagram in Fig. 3, which captures the trust with respect to one specific seller, this diagram uses the variable `p` for the probability of the `palt` operands. This variable can be referred to in the objective diagram, since it is an out parameter of the subjective diagram. The `trust.p` expression in the objective diagram refers to this output value. The guards in the objective diagram specify that Alice sends advance payment if she believes that the probability of receiving the item is greater than or equal to 0.5. This will happen in 70% of the cases, since the operand where this guard holds has the probability of 0.7.

2.5 Policy specification

A policy is a set of rules that determines choices in the behavior of a system [19], and is used in policy based management. Each rule determines a system choice of

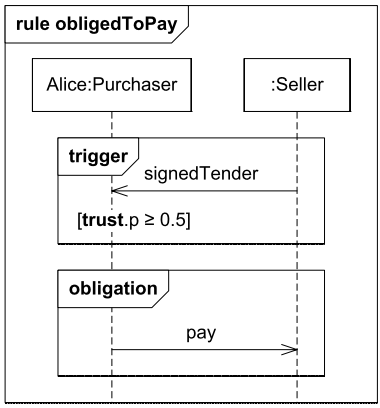

Fig. 5 Example of a policy rule

behavior, where a given trust level is a decisive factor for each choice. Enforcement of the given rules aims to ensure the optimal balance of the risks and opportunities that are imposed by trust based decisions within the system.

To formalize the policy the trust analysis method, proposed in [17], uses Deontic STAIRS [21], which is a language for expressing policies, and based on UML sequence diagrams. Deontic STAIRS has the expressiveness to specify constraints in the form of obligations, prohibitions, and permissions, corresponding to the expressiveness of standard deontic logic [12]. 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" [20]. 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.

Figure 5 shows an example of a policy rule in Deontic STAIRS for the scenario described in this section. The keyword `rule` in the upper left corner indicates that the diagram specifies a policy rule, while **obligedToPay** is the name of the rule. The diagram consists of two parts, a trigger and an interaction that is the operand of a deontic modality.

The first operator with keyword `trigger` specifies the circumstances under which the rule applies and consists of an interaction and a condition. The former refers to a scenario such that when it occurs, the rule applies. In this case the scenario is the reception by Alice of a signed tender. 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 0.5 or higher.

The second operator with keyword `obligation` shows the modality of the rule, while its operand specifies the behavior that is constrained by the rule. In this case, the relevant behavior is that Alice sends payment to the seller. According to **obligedToPay**, she is obliged to do so, given that the trigger is fulfilled. On the other hand, if the keyword had been `prohibition` then Alice would have been prohibited from sending payment, while if the keyword had been `permission` then Alice could choose whether or not to send payment.

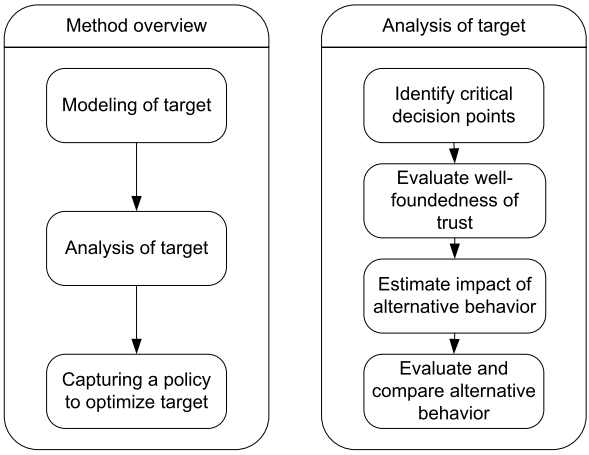

Fig. 6 Overview of the method proposed in [17]. The *right-hand side of the figure* shows the sub-steps of the second step

3 The trust analysis method

In this section we give a brief overview of the trust analysis method that was introduced in [17]. For further details we refer to [17] and of course Sect. 5 of this paper which describes how the method was used in the industrial project on which this paper reports.

Figure 6 shows an overview of the method. There are three major steps. The first step is to model the target, the second step is to analyze the target and the third step is to capture policies to optimize the behavior of the target based on the knowledge acquired in the first two steps.

Step 1. Modeling of target. In order to analyze a system, we first need to understand the system under analysis, including the behavior of its users. A major goal of the first step and the resulting models is to provide such an understanding. However, as most systems are highly complex, it is neither feasible nor desirable to take every detail into account. Therefore the target should be modeled at a level of abstraction suitable for the analysis to come. Thus, the models should only capture the aspects of the system that enhances our understanding of the decisions that are taken on the basis of trust and the considerations that lie behind these decisions, as well as the resulting system behavior and outcomes that are relevant.

As explained in Sect. 2, the modeling approach is based on UML sequence diagrams. The reason is that trust is mainly of relevance in the context of interactions between different entities, and sequence diagrams are well suited for modeling interactions. Moreover, UML sequence diagrams are fairly easy to understand at an intuitive level. This is important, as the models developed in the first step should serve as a point of focus for discussions and as an aid in communication between the analysts and participants throughout the analysis. The extensions of UML sequence diagrams provided by subjective STAIRS [18] ensures that the trust considerations behind decisions can be captured in the models, as well as the resulting system behavior.

Step 2. Analysis of target. After a suitable model of the target has been established, the next step is to conduct the actual analysis. This involves investigating the current system behavior and the way in which trust-based decisions are being made, as well as potential alternative behaviors. The aim is to obtain a good understanding of the risks and opportunities involved. The analysis is divided into four sub-steps.

Step 2.1. Identify critical decision points. In this sub-step critical decision points that will be further investigated are identified. This will typically be points where actors in the system make trust-based decisions. But it may also be points where one could benefit from introducing new trust-based decisions. For example, if time is a critical factor, it may be more important to make a quick decision than to make the optimal decision. In such cases, it may be better to allow actors to make decisions based on trust than to insist on more time-consuming decision procedures.

Step 2.2. Evaluate well-foundedness of trust. Trust involves a subjective estimate of the potential behavior of another entity. The second sub-step of Step 2 consists of evaluating to what degree the subjective estimates reflect reality. In the industrial project on which this paper reports this step was not relevant since our task was not to evaluate an existing trust solution, but rather to develop a policy from scratch.

Step 2.3. Estimate impact of alternative behavior. In this sub-step the impact of various alternative behaviors that the system may potentially perform is investigated with respect to risks and opportunities. The goal is to get an understanding not only of the current "as-is" system behavior, which may not be optimal, but also of potential alternatives. Typically, this involves asking "what if" questions about the system, and capturing the answers in models. For example: what would be the overall effect on the system behavior if a certain actor was more (or less) willing to engage in interactions with other entities? What happens if a different policy is applied when making a certain decision?

Step 2.4. Evaluate and compare alternative behavior. In the final sub-step, the different alternative behaviors that were identified and investigated in the previous step are evaluated and compared. The purpose is to identify behaviors that should be sought or avoided.

Step 3. Capturing a policy to optimize target. The final step of the method proposed in [17] consists of using the obtained knowledge about preferred behavior to form policies to ensure and enforce the desirable behavior.

4 The industrial project on electronic procurement

We now present the industrial project in which the trust analysis method outlined in Sect. 3 was applied. The trust analysis method was used to model and analyze a public eProcurement system, which makes use of a Validation Authority (VA) service for validating electronic certificates and signatures. We first present the public eProcurement system, before describing how this system can make use of the VA service. Then we present criteria for evaluating the trust analysis method.

4.1 Public electronic procurement

Public eProcurement is used by public authorities within the EU to award public work contracts, public supply contracts, and public service contracts to economic operators [16]. We consider only the open procedure for individual contracts as specified in [3]. In the open procedure, any interested economic operator may submit a tender. The procedure consists of three phases: eNotification, eTendering, and eAwarding. In the eNotification phase a procurement officer[1] creates a call for tenders. This call specifies the requirements of the contracting authority for the goods/services/works to be procured. In the eTendering phase, interested economic operators will create tenders containing legal, financial, and technical information. Before submitting the tender electronically, one or more persons representing the economic operator need to sign the tender with their electronic IDs (eIDs), issued by Certificate Authorities (CAs). When received by the system, the system will examine whether the tender is compliant with the requirements defined in the call, including examining whether the digital signatures in the tender are valid. The eAwarding phase begins after the deadline for submission has expired. In this phase the contract is awarded based on an evaluation of the received tenders.

4.2 The validation authority service

For the eProcurement system it is important to be able to accept electronically signed tenders from electronic operators from all over Europe, regardless of the eID used by the operator. Due to the potential large number of CAs, the technical validation of eIDs and digital signatures has some challenges with respect to scaling [14], but the real problem is the assessment of the risk implied by accepting a digital signature. Here, one particular concern is that an economic operator can refute the validity of the offer stated in the submitted tender, if awarded the contract. The eProcurement system can ensure that this risk is acceptable by making an assessment of the signature quality and accepting only those of a certain quality. The higher the quality is, the harder it would be for an economic operator to refute the validity of a submitted tender. The quality of a signature [15] can be decided from the quality of the eID, which is derived from the certificate policy of the CA, and the cryptography used. A certificate policy may be written in a foreign language and may refer to a foreign legislation, so with a large number of CAs, the contracting authorities will have a hard time determining the quality of digital signatures. Thus, it will be hard if not impossible for the contracting authorities to have agreements with all the CAs on which it may want to rely, which again limits the number of economic operators that can submit tenders. A solution to this, as proposed in [15], is to use a VA as the single trust anchor, as shown in Fig. 7. In the figure we can see that the VA supports a number of CAs. For each CA that it supports, the VA is able to assess the quality of the eIDs issued by this CA and the signatures produced with those eIDs. A relying party, in this case the eProcurement system, can then validate and assess the quality of the

[1] Representative for the contracting authorities.

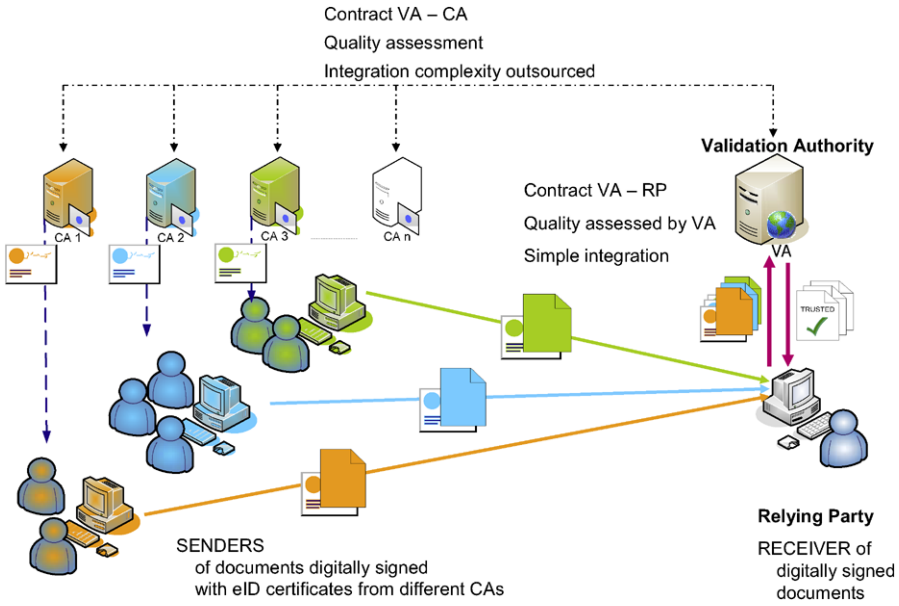

Fig. 7 Figure from [15]. The figure shows how a relying party (in this case a public eProcurement system) may use a VA service to validate and assess the quality of signatures in signed documents (in this case signed tenders)

signature[2] in a signed tender by issuing a request to the VA. If the eID used to create the signature has been issued by a supported CA, the VA will use the CA to validate the eID, while the quality of the signature is computed by the VA by applying the formula

$$\begin{aligned}\text{Signature Quality} = {} & \text{eID Quality} + \text{Hash Quality} \\ & + \text{Public Key Crypto Key Length Quality},\end{aligned}$$

which will assign a Signature Quality value from 0 to 20 according to criteria further specified in [15]. This value is then compared to the minimum required quality level requested by the eProcurement system. If the signature is valid, meaning that the technical validation of the eID and the signature was successful, and the signature has sufficient quality, the VA will give the tender a trusted verdict. Otherwise, the VA will give the tender a not trusted verdict.[3] By trusting the VA and its assessments, the eProcurement system is able to trust any CA that the VA handles. Hence, the eProcurement system can support a large number of CAs and it gets a one-stop shopping service for verification of digital signatures and eIDs and quality assessment of digital signatures.

[2]It is possible to sign a tender with more than one signature. The VA is able to make an overall quality assessment of all these signatures.

[3]The VA can also give an inconclusive verdict. This will happen in the cases when the VA cannot validate the eID and/or signature and/or cannot assess the quality of the signature.

4.3 Evaluation criteria

A major objective with applying the trust analysis method in the industrial project was to get an idea of how well the method performs in a practical setting. To do so we need a set of evaluation criteria and they are characterized and motivated in the following. The criteria are general in the sense that they do not target the eProcurement system or VA service specifically; they are equally valid for other kinds of trust-related infrastructures on which the trust analysis method may be applied.

When evaluating a method for trust analysis there are of course many concerns. To make sure that we covered the most important concerns we started by identifying groups of stakeholders of relevance for the method. What is important for one group may of course be less so for another group. We identified three main groups of stakeholders, and the evaluation criteria are based on the point of view for each of these groups. First, the *customers* are those who pay for the analysis. Typically, this will be managers and decision makers. They do not necessarily take part in the analysis process themselves, but will use the results of the analysis as a basis for making policy decisions. Second, the *analysts* are those who will conduct the analysis (process) and document results. They know the analysis method, but cannot be assumed to know the particular target system at the start of the analysis. Third, the *participants* are people such as decision makers, system users, developers, or engineers with whom the analysts interact during the analysis process. We now present evaluation criteria classified according to the stakeholder group for which they are most relevant.

For the *customer* of a trust analysis, the overall goal is to make the right trust policy decisions. This requires a good understanding of the outcome of potential alternative trust policies. Hence:

> ***EC1****: The trust analysis should provide the customers with a good basis for making trust policy decisions. This means that sufficient information about the impact of the potential alternatives must be provided.*

Clearly, the cost of the analysis needs to be justified with respect to the benefit for the customer. Hence:

> ***EC2****: The trust analysis should be cost effective.*

The task of the *analyst* is to conduct the trust analysis and to document the findings within the allotted time and cost frame. This means that the trust analysis method should be sufficiently simple to be carried out within a reasonable time frame. However, as this is implied by the requirement expressed in **EC2**, we do not include this as a separate criterion. On the other hand, the analyst would like to document the findings, and in particular all assumptions and constraints on which their validity depends, to cover him/herself as much as possible. Hence:

> ***EC3****: The modeling approach should be sufficiently expressive to capture the information, assumptions, and constraints of relevance.*

The *participant* is supposed to communicate her or his knowledge in such a way that the analysis will result in correct models of the target, and the models should serve as a means of communication between the analysts and participants. It is therefore important that the models are comprehensible for the participants when properly

assisted by an analyst. Otherwise, it will be hard for them to identify shortcomings and errors. Hence:

***EC4**: The models should be comprehensible for the participants of the analysis.*

5 Trust analysis in the industrial project

In this section we present how the trust analysis method as described in Sect. 3 was applied in the industrial case outlined in Sect. 4.

5.1 Step 1. Modeling the target

The trust analysis focused on the scenarios where the eProcurement system makes decisions based on trust, i.e. where it is decided whether a received tender should be trusted to be authentic or not. On the one hand, it was an objective to minimize the risk that non-authentic tenders were accepted for further evaluation in the eAwarding phase, as contracts should not be awarded based on non-authentic tenders. On the other hand, it was also an objective to avoid authentic tenders being rejected without further evaluation.

There was some discussion on whether non-authentic tenders actually represent a real problem. Although this may not be the case today, it was agreed that this may easily become a problem in the future, as the use of eProcurement increases. It is easy to imagine cases where economic operators submit false tenders in a competitor's name. The motivation for this could be, for example, to ensure that minimum requirements on the number of received tenders to enable eAwarding are fulfilled, or to bind the competitor to unfavorable obligations, or to make the operator's own tender appear more attractive compared to a false costly tender.

The decision on whether to trust the authenticity of a tender is made in the eTendering phase, while the selection of the best tender based on price, quality, and so on from the tenders judged to be authentic is made in the eAwarding phase. Thus, for the purpose of the trust analysis we are only interested in the behavior related to submission of tenders in the eTendering phase. The task in Step 1 is therefore to model this behavior. Figure 8 shows the resulting overview diagram.

First the eProcurement system needs to find the minimum quality level the signatures have to comply with to be accepted to the eAwarding phase. This particular process is described in more detail in the diagram **chooseQualityPolicy** in Fig. 9.[4] Choosing the quality policy and communicating the choice to the VA is typically done even before the eNotification phase, since the economic operators must be informed about the requirements for the submission. Note that the eNotification phase is not captured in the models, as it is of little relevance for the trust analysis.

After the quality policy has been set an economic operator may submit a tender `t` to the eProcurement system. This is represented by the message `submitTender(t)` in the diagram. The eProcurement system will validate the signatures of this tender by

[4]The `ref` construct is a reference to another diagram. Its meaning is the same as we would get by inserting the contents of the referred diagram at the place of the reference.

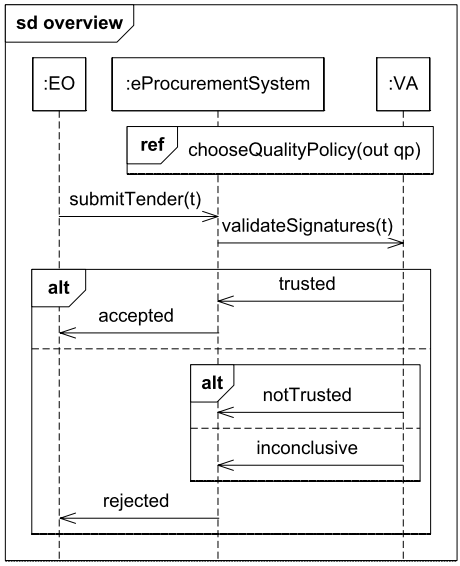

Fig. 8 A simplified description of how submitted tenders are handled by the eProcurement system

using the VA service `validateSignatures(t)`. The VA will then use the required minimum quality level to decide whether the signature should be trusted or not.

The first operand of the outermost `alt` operator describes the case where the tender is reported `trusted` by the VA, and therefore is accepted for further evaluation by the eProcurement system. The second operand describes the case where the tender is reported as `notTrusted` or as `inconclusive`, and therefore rejected by the eProcurement system.

We now explain how the choice of quality policy level performed by the eProcurement system is captured in the models. Intuitively, the process of choosing one of the 21^5 quality policy levels can be described as follows: the eProcurement system uses a threshold value that specifies the least amount of trust that is needed to accept the risk of accepting a non-authentic tender. In order to balance the risk of accepting a non-authentic tender against the desire not to reject authentic tenders, the eProcurement system chooses the lowest quality policy level needed to ensure that its trust exceeds the threshold.

Figure 9 describes the process of choosing a suitable quality policy based on trust. The diagram **chooseQualityPolicy** shows that there are 21 alternative quality policies from which the eProcurement system may choose. After choosing the quality policy in terms of an assignment, the eProcurement system communicates to the VA service the chosen policy, as shown by the `setQualityPolicy(qp)` message at the bottom of the diagram.

The trust level which the `threshold` is compared to, is captured by expressions of the form `trust(j).p`, where `j` is one of the 21 quality policy levels; this value is bound to the input parameter `qp` of the subjective diagram **trust** in Fig. 9. So,

[5] Remember that the quality policy scale is from 0 to 20.

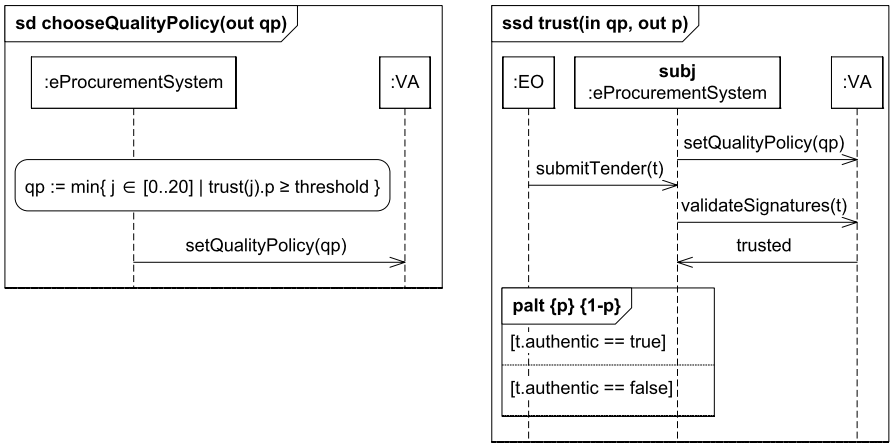

Fig. 9 **chooseQualityPolicy** and **trust** describe how the quality policy is found and set

for example, `trust(7).p` yields the return value of the subjective sequence diagram **trust**, assuming quality policy level 7 is used. As shown by the lifeline head, the eProcurement system is the trustor. Therefore, the expression `trust(j).p` represents the trust of the eProcurement system that a tender that receives a trusted verdict from the VA service is indeed authentic, given that quality policy level `j` is used. Note that the diagrams in Fig. 9 do not refer to one specific VA. Hence, `trust(j).p` may yield different values for different VAs for the same tender.

5.2 Step 2. Analyzing the target

We now explain how the sub-steps of Step 2 were performed.

Step 2.1. Identify critical decision points. The only critical decision point that was identified was the point where the signature quality policy is chosen by the eProcurement system. The reason for this was that the analysis was performed from the point of view of the eProcurement system, with the purpose of setting the right trust threshold. This decision point is represented by the assignment in the diagram **chooseQualityPolicy** in Fig. 9.

Step 2.2. Evaluate well-foundedness of trust. As explained in Sect. 3, this step was not relevant in the project on which this paper reports. Our task was not to evaluate one particular trust solution, but rather come up with a policy for how to choose quality policy level.

Step 2.3. Estimate impact of alternative behavior. As the decision point under analysis was the choice of quality policy level, the task of this sub-step was to estimate the impact of choosing different levels with respect to how many of the authentic or non-authentic tenders would receive the different verdicts from the VA service. This would serve as a basis for the evaluation and comparison in the next sub-step. To do so, we obviously needed to collect data. No historical data were available, and we

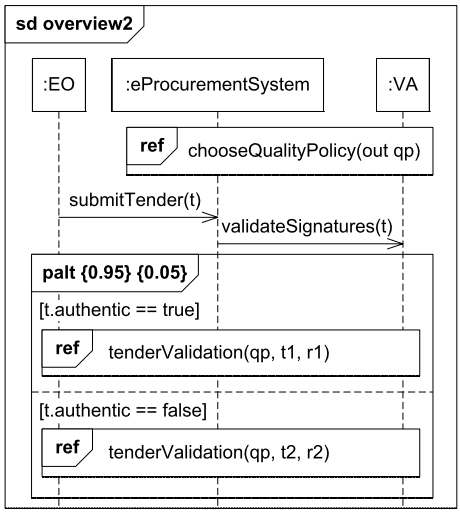

Fig. 10 A simplified description of how submitted tenders are handled by the eProcurement system, where probabilities have been assigned to alternatives

had to base ourselves on expert judgments. The experts were all employees of DNV including the two co-authors from DNV. The data they provided is documented in Figs. 10, 11, and Table 1. Figure 10 is a refinement of Fig. 8. It is unchanged above the `palt` operator. The first operand of the `palt` operator represents the cases where tenders are authentic, while the second alternative represents the cases where they are non-authentic. We consider a tender to be authentic if it actually comes from the company (EO) in whose name it has been submitted. This applies even if the employees who have signed the tender electronically is not strictly speaking authorized by the company to do so, as long as the company acknowledges the tender and intends to honor its commitment. To emphasize the fact that the EO is the only entity who actually knows whether the tender is authentic, we have used guards on the `EO` lifeline in the model to represent whether a tender is authentic or not; the two cases are represented by the guards `t.authentic==true` and `t.authentic==false`, respectively. The probability 0.95 assigned to the first `palt` operand in Fig. 10 captures that 95% of received tenders are authentic. The remaining 5% are non-authentic and the second `palt` operand is therefore assigned the probability 0.05. The two operands of the `palt` operator in Fig. 10 refer to the same parameterized diagram (i.e. **tenderValidation** in Fig. 11), as the behavior of the system for the two cases only differ with respect to the probabilities for the alternatives. The **tenderValidation** diagram has three input parameters; namely the quality policy (`qp`), the probability (`x`) of being judged as trusted by the VA with respect to the selected quality policy, and similarly the probability (`y`) of being judged as not trusted as opposed to inconclusive in the other case. The first operand of the outermost `palt` operator in **tenderValidation** gives the probability for the tender being reported `trusted` by the VA, and therefore is accepted for further evaluation by the eProcurement system. The second operand shows the case where the tender is reported as `notTrusted` or as `inconclusive`, and therefore rejected by the system; this will occur with probability $1 - x$. Within this alternative, the `notTrusted` verdict from the VA will occur with probability `y`, while the `inconclusive` verdict will occur with probability $1 - y$.

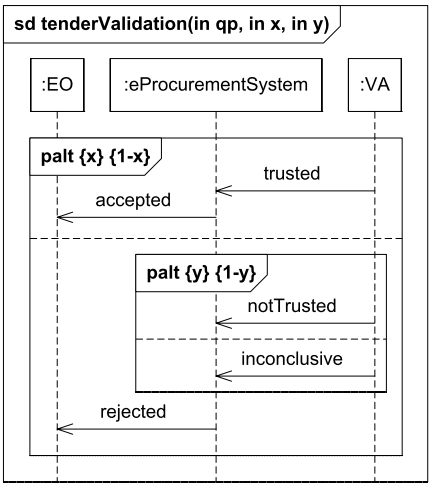

Fig. 11 tenderValidation shows how tenders are handled

The actual values of `x` and `y` depend on whether the tender is authentic or not. Therefore the references to **tenderValidation** in the operands of the `palt` in Fig. 10 bind `x` and `y` to different entities. In the first operand representing the cases where tenders are authentic, `x` is bound to `t1` and `y` is bound to `r1`. In the second operand representing the cases where tenders are non-authentic, `x` is bound to `t2` and `y` is bound to `r2`. The intuitive meaning of `t1`, `t2`, `r1`, and `r2` for a given quality policy `qp` can be summarized as follows: `t1` denotes the probability of assigning a trusted verdict to an authentic tender; `r1` denotes the conditional probability of assigning a not trusted verdict (as opposed to inconclusive) for an authentic tender given that a trusted verdict is not assigned; `t2` denotes the probability of assigning a trusted verdict to a non-authentic tender; `r2` denotes the conditional probability of assigning a not trusted verdict (as opposed to inconclusive) for a non-authentic tender given that a trusted verdict is not assigned.

Table 1 shows how the representatives from DNV estimate that the probabilities will vary according to the quality policy. Note that even though the VA service offers a quality scale from 0 to 20, it was deemed sufficient to analyze only five different quality policy levels for the purpose of this analysis. Based on a consideration of criteria for assigning quality levels, the following steps on the scale were selected for analysis: 0, 5, 7, 10, and 20. The first line in the table provides the values of the parameters `t1`, `r1`, `t2`, and `r2`, in the diagram in Fig. 10, if the quality policy 0 ($\mathtt{qp} = 0$) is chosen. The second line provides the values in the case where the quality policy 5 is chosen and so on.

Given the data captured by Table 1, we calculated the probabilities for the possible combinations of authenticity and verdicts, which amounted to a simple multiplication of the probabilities assigned in the objective diagrams in Figs. 10 and 11. For example, the probability that a given tender is authentic and receives a trusted verdict is obtained by $0.95 \times \mathtt{t1}$, while the probability that it is non-authentic and receives an inconclusive verdict is obtained by $0.05 \times (1 - \mathtt{t2}) \times (1 - \mathtt{r2})$. Table 2 shows the result from these calculations (when inserting the values from Table 1 for the variables `t1`, `t2`, `r1`, and `r2`).

Table 1 The probabilities corresponding to the quality policy

Quality policy	t1	r1	t2	r2
0	0.65	0.05	0.05	0.10
5	0.80	0.15	0.01	0.20
7	0.95	0.40	0.005	0.50
10	0.75	0.70	0.002	0.80
20	0.80	0.80	0.001	0.90

Table 2 Probabilities for authentic and non-authentic tenders

Quality policy	Authentic			Non-authentic		
	Trusted	Not trusted	Inconclusive	Trusted	Not trusted	Inconclusive
0	0.61750	0.01663	0.31589	0.00250	0.00475	0.04275
5	0.76000	0.02850	0.16150	0.00050	0.00990	0.03960
7	0.90250	0.01900	0.02850	0.00025	0.02488	0.02488
10	0.71250	0.16625	0.07125	0.00010	0.03992	0.00998
20	0.76000	0.15200	0.03800	0.00005	0.04955	0.00500

Figure 12 shows the left-hand part of Table 2 as a trend-graph, i.e. it shows the probability that a tender is authentic and receives each of the three possible verdicts, depending on the chosen quality level.[6] On the left-hand side of the graph, we see that the probability of getting a trusted verdict is relatively low (but increasing), while the probability of getting an inconclusive verdict is correspondingly high (but decreasing). According to the domain experts providing the data, the reason for this is that when a request for tenders is announced with low certificate and signature requirements, relatively many of the received tenders will use certificates from CAs that are not supported by a VA; there are many CAs offering low quality eIDs, and a VA service is primarily aimed at supporting CAs with higher quality eIDs. After quality policy level 7, we see a decrease in the probability of receiving a trusted verdict. According to the same experts, this is due to the fact that when higher quality policy levels are used, more of the received tenders will use certificates from CAs that are supported by the VA, but of insufficient quality.

Figure 13 shows the right-hand part of Table 2 as a trend-graph, i.e. it shows the probability that a tender is non-authentic and receives each of the three possible verdicts. Here we are operating with very small scales, due to the small amount (5%) of non-authentic tenders, and the probability for getting the trusted verdict is almost non-existing for all quality policy levels. Not surprisingly, the probability for a non-authentic tender getting the not trusted verdict increases with increasing quality policy level. Furthermore, the probability of an inconclusive verdict decreases with increasing quality level, as more of the received tenders will use certificates from CAs that are supported by VA when high quality policies are used.

[6] Recall that 95% of tenders are assumed to be authentic in all cases.

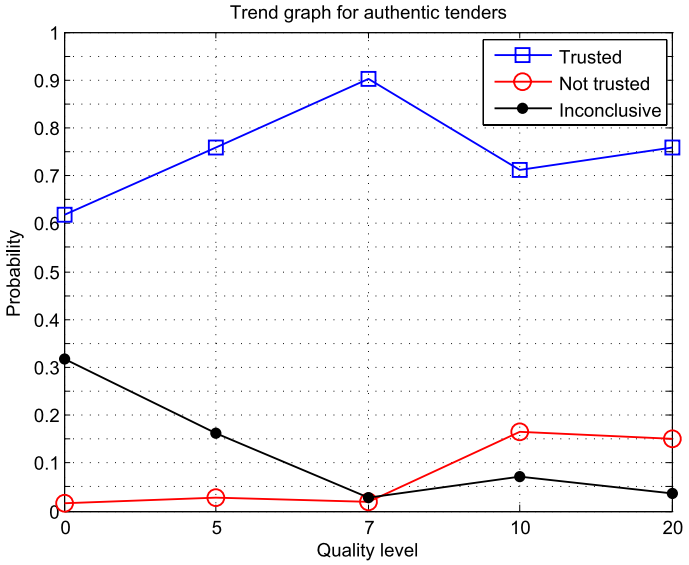

Fig. 12 Trend graph for authentic tenders

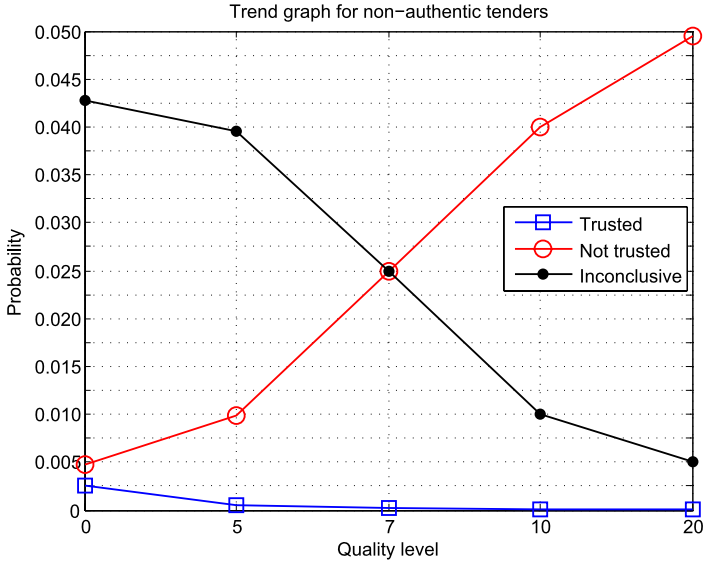

Fig. 13 Trend graph for non-authentic tenders

Step 2.4. Evaluate and compare alternative behavior. In order to decide which of the quality policies that will give the optimal behavior of the system, we looked at the probabilities for desirable and undesirable outcomes (opportunities and risks) for the

different quality levels, with the goal of finding the optimal balance. For each quality level, the desirable outcomes are as follows:

- A tender that has been accepted by the system, due to a trusted verdict from the VA, is authentic.
- A tender that has been rejected by the system, due to a not trusted or an inconclusive verdict from the VA, is non-authentic.

We seek to maximize the probabilities of these outcomes, since these outcomes represent opportunities. The probabilities are given as follows:

- $P(aut|acc)$—The conditional probability of a tender being authentic, given that it is accepted by the system.
- $P(not\ aut|not\ acc)$—The conditional probability of a tender being non-authentic, given that it is rejected by the system.

On the other hand, for each quality level, the undesirable outcomes are as follows:

- A tender that has been accepted by the system, due to a trusted verdict from the VA, is non-authentic.
- A tender that has been rejected by the system, due to a not trusted or an inconclusive verdict from the VA, is authentic.

We seek to minimize the probabilities of these outcomes, since these outcomes represent risks. The probabilities are given as follows:

- $P(not\ aut|acc)$—The conditional probability of tender being non-authentic, given that it is accepted by the system.
- $P(aut|not\ acc)$—The conditional probability of a tender being authentic, given that it is rejected by the system.

From the diagrams in Figs. 10 and 11 we get the following values directly: the probability $P(aut) = a$ for a tender being authentic is 0.95, irrespective of the quality policy, while the probability for a tender being non-authentic is $P(not\ aut) = 1 - a$. We also have the probabilities $P(acc|aut) = \texttt{t1}$ and $P(not\ acc|aut) = 1 - \texttt{t1}$ for a tender being accepted and not accepted by the system, given that it is authentic, while $P(acc|not\ aut) = \texttt{t2}$ and $P(not\ acc|not\ aut) = 1 - \texttt{t2}$ give the probabilities for a tender being accepted and not accepted, given that it is non-authentic. Values for `t1` and `t2`, depending on the chosen quality level, are taken from Table 1. The probabilities for a tender being accepted and not accepted are obtained as follows:

$$\begin{aligned} P(acc) &= P(aut) \times P(acc|aut) + P(not\ aut) \times P(acc|not\ aut) \\ &= a \times \texttt{t1} + (1 - a) \times \texttt{t2} \end{aligned} \tag{1}$$

$$P(not\ acc) = 1 - P(acc) \tag{2}$$

The conditional probabilities, mentioned above, were calculated for the different quality levels by applying Bayes' theorem as follows:

$$P(aut|acc) = \frac{P(acc|aut) \times P(aut)}{P(acc)} = \frac{\texttt{t1} \times a}{a \times \texttt{t1} + (1 - a) \times \texttt{t2}} \tag{3}$$

Table 3 Table showing the probabilities related to opportunity (column 2 and 5) and risk (column 3 and 4) for the different quality policies

Quality policy	$P(aut\|acc)$	$P(aut\|not\ acc)$	$P(not\ aut\|acc)$	$P(not\ aut\|not\ acc)$
0	0.995968	0.875000	0.004032	0.125000
5	0.999343	0.793319	0.000657	0.206681
7	0.999723	0.488432	0.000277	0.511568
10	0.999860	0.826374	0.000140	0.173626
20	0.999934	0.791832	0.000066	0.208168

$$P(aut|not\ acc) = \frac{P(not\ acc|aut) \times P(aut)}{P(not\ acc)} = \frac{(1 - \texttt{t1}) \times a}{1 - (a \times \texttt{t1} + (1 - a) \times \texttt{t2})} \quad (4)$$

$$P(not\ aut|acc) = 1 - P(aut|acc) \quad (5)$$

$$P(not\ aut|not\ acc) = 1 - P(aut|not\ acc) \quad (6)$$

The results of the calculations are shown in Table 3. For $P(aut|acc)$, which we want to maximize, there is little difference between the values of $P(aut|acc)$ for the different quality policies. On the other hand, for $P(not\ aut|not\ acc)$, which we also want to maximize, we see that for level 7 we have a much higher value than for the others.

5.3 Step 3. Capturing a policy to optimize target

The numbers in Table 3 provide useful input, assuming of course that the expert judgments are sound. However, they do not take the nature of the call for tender into consideration, which of course is an essential factor when formulating a policy. After all, the significance of the numbers in Table 3 depends heavily on what is to be procured. If the cost of goods to be procured is low (e.g. pencils for the administration), we would probably worry only about $P(aut|acc)$ and based on that choose quality policy 0. This is partly because the difference in $P(aut|acc)$ for the quality policy levels does not matter much when the cost of goods to be procured is low, and partly because a higher quality level might frighten off potential submitters of tenders.

On the other hand, if the cost of the goods to be procured is very high (e.g. new fighter planes in the extreme case) the procurer would probably want as much legal coverage as possible and use quality policy level 20, since this gives the best value for $P(aut|acc)$. Moreover, if the goods to be procured are so costly that it is important to avoid disqualifying authentic tenders as well as obtaining a high level of trust in certificates, quality policy level 7 seems to be the best option.

Based on these considerations we ended up with the policy rules specified in Figs. 14–16. We make the assumption that the eProcurement system trusts the VA and its assessments. This is important since an eProcurement system cannot make use of the VA service if it is not trustable. The trigger of each rule contains a condition which limits the applicability of the rule to a set of system states. For rule **qp0** in Fig. 14 the condition is that the cost of the goods to be procured is low, while for rule **qp20** in Fig. 15 it is that the cost is very high. For rule **qp7** in Fig. 16 the condition

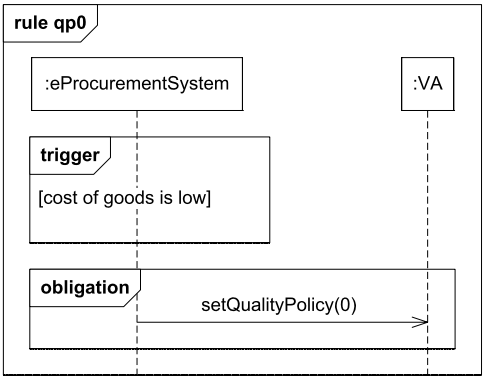

Fig. 14 Policy rule for the selection of quality policy level 0

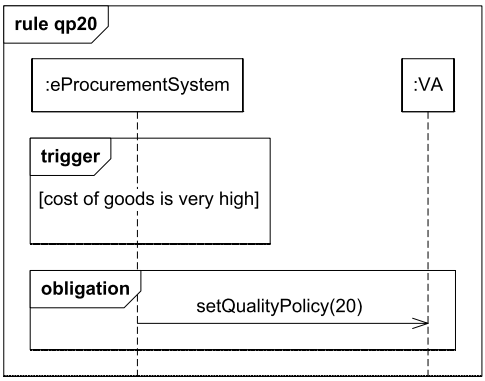

Fig. 15 Policy rule for the selection of quality policy level 20

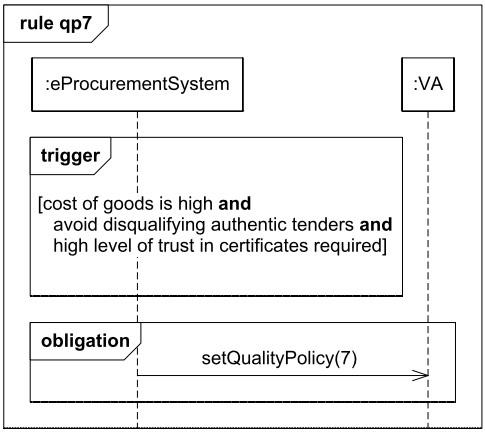

Fig. 16 Policy rule for the selection of quality policy level 7

is that the cost is high, that disqualifying authentic tenders should be avoided, and a high level of trust in certificates is required. Depending on which one of these three conditions that is satisfied, the eProcurement system *must* use either quality policy level 0, 7, or 20.

Table 4 The number of hours used on the trust analysis, not including writing of a final report

	Meetings	Preparations
Analysts	27	100
Participants	20	–

6 Evaluation of the trust analysis method

In this section we evaluate the performance of the trust analysis method in the industrial project with respect to the criteria presented in Sect. 4.3.

***EC1**: The trust analysis should provide the customers with a good basis for making trust policy decisions. This means that sufficient information about the impact of the potential alternatives must be provided.*

The project gave strong indications that the trust analysis method is feasible in practice. We went through the various steps of the method (with exception of Step 2.2) in close interaction with the industrial representatives of the customer and delivered a result in the form of a policy that we believe gives an institution making use of eProcurement system with a VA service useful input on selecting the right quality policy level. Based on models developed in the modeling step of the method we collected expert judgments and documented them in the models.

Of course an industrial case like this can never give solid repeatable evidence of anything. There are too many factors influencing what happens. In the case of our project it may be argued that we should have used historical data rather than expert judgments, but such data were not available. It may also for example be argued that we should have had involvement of representatives of an eProcurement institution, and having the inventors of the trust analysis method in the analyst team is of course also rather extraordinary.

***EC2**: The trust analysis should be cost effective.*

The trust analysis was carried out in a series of five meetings, each of which took about 1.5 hours. Typically, four analysts and two to four participants/representatives of the customer took part in the meetings. In addition, the analysts spent time between the meetings developing models and preparing the next meeting. Table 4 shows an estimate of the total amount of time spent on the trust analysis. Note that time spent on writing a final report is not included in the numbers—this depends heavily on the type of report the customer wants. There are some issues that must be taken into consideration when evaluating these numbers. Firstly, this was the first time the trust analysis method was applied to a real industrial case. Hence, even though the analysis team included authors of the paper [17] proposing the trust analysis method, none of the analysts had any experience with applying the method in a realistic setting. It can reasonably be assumed that the process will be more effective as the analysts gain experience with applying the trust analysis method. Furthermore, the reason for having as many as four analysts was a desire to learn as much as possible from this first application of the method. Normally, we believe two analysts would be enough.

Based on the experience gained, we believe that it should be possible to carry out this kind of analysis with within a time frame of ca. 80 man-hours spent by analysts

(not including writing a final report) and ca. 20 man-hours spent by participants. Whether this counts as being cost effective has to be evaluated in light of the values at stake in the target of analysis.

***EC3**: The modeling approach should be sufficiently expressive to capture the information, assumptions, and constraints of relevance.*

The participants provided a lot of information about the target during the analysis process. There were no instances where we were not able to capture the relevant information in the models. The diagrams in Figs. 8–11 contain all the information that was finally used (and deemed relevant) for finding the best trust policy. These diagrams have been abstracted from more detailed models of the target. We also formalized the policy we recommended in the end.

***EC4**: The models should be comprehensible for the participants of the trust analysis.*

During the meetings, models were presented and explained by an analyst in order to validate the correctness of the models. There were many instances where the participants pointed out parts of a model that did not correctly represent the target, provided additional information, or asked relevant questions about some detail in a model. This indicates that the models were in general comprehensible for the participants, and our experience is that the models served well as an aid in establishing a common understanding of the target between the participants and analysts. The fact that all the participants in this analysis had a strong technical background may have contributed to making the models easier for them to understand than would be the case for a more diverse group. Note also that we do not know whether the models would have been well understood by the participants without any guidance or explanation, as all the models were presented by an analyst. In particular with respect to subjective diagrams and their relation to the objective diagrams, we believe it is necessary to have an analyst explain the diagrams in order for them to be understood by the participants if they have no prior experience with the notation, other than some background in UML.

There was one aspect of the models that proved hard to understand for the participants. This occurred when the operands of `palt` operators contained more `palt` operators. In Fig. 10 the `palt` operator contains references to the diagram in Fig. 11, which again contains a `palt` operator with another `palt` operator inside one of its operands. This nesting of operators made it hard for the participants to understand exactly what each of the alternatives represented. In order to explain, one of the analysts drew a tree-structure where the root represented the outermost `palt` operator and each branch represented a `palt` operand. Based on this experience, we believe that the presentation style of UML interaction overview diagrams are better suited than sequence diagrams to present cases with nested alternatives. Interaction overview diagrams have the same kind of semantics as sequence diagrams and are often used in combination with sequence diagrams, but nested alternatives are represented syntactically by a branching point (the operator) with branches (the operands), rather than boxes inside boxes.

7 Conclusion and related work

The paper has presented experiences from using a UML-based method for trust analysis in an industrial project focusing on the modeling and analysis of a public eProcurement system making use of a validation authority service for validating electronic certificates and signatures. The contributions of the paper include:

1. a detailed account of how the method proposed in [17] scales in an industrial context; in particular, we have illustrated the specific constructs for capturing
   - beliefs of agents (humans or organization) in the form of subjective probabilities;
   - factual (or objective) probabilities of systems which may contain agents whose behavior is described in the form of subjective probabilities;
   - trust decisions in the form of policy rules;
2. an evaluation of the feasibility of the method in an industrial context; in particular, it is claimed that
   - the project gave strong indications that the trust analysis method is feasible in practice;
   - this kind of trust analysis can be carried within the frame of 100 man-hours (not including writing of a final report);
   - there were no instances were the analysts were not able to capture the relevant information in the models;
   - the models to a large extent were comprehensible for the industrial participant with some experience in UML but no background in the specific extensions used by the method.

The method for trust analysis makes use of models that capture the subjective trust considerations of actors, as well as their resulting behavior. We are not aware of other approaches that combine these elements in this way. However, the issues of uncertainty, belief, and trust have received much attention in the literature. We now present a small selection of the proposed approaches.

Giorgini et al. [6] presents a formal framework for modeling and analyzing trust and security requirements. Here, the focus is on modeling organizations, which may include computer systems as well as human actors. The approach is based on a separation of functional dependencies, trust, and delegation relationships. Trust and security requirements can be captured without going into details about how these will be realized, and the formal framework supports automatic verification of the requirements.

An interesting approach to modeling and reasoning about subjective belief and uncertainty is subjective logic [7, 8], which is a probabilistic logic that captures uncertainty about probability values explicitly. The logic operates on subjective belief about the world. Different actors have different subjective beliefs, and these beliefs are associated with uncertainty. The approach makes it possible, for example, to calculate to what degree an actor believes that a system will work based on the actor's beliefs about the subsystems, or to calculate the consensus opinion of a group of actors. Subjective logic deals strictly with the actors' beliefs and reasoning, and does not address the question of how their beliefs affect their behavior.

The belief calculus of subjective logic can be applied in risk analysis to capture the uncertainty associated with such analysis, as shown in [9]. This is achieved by using subjective beliefs about threats and vulnerabilities as input parameters to the analysis. Through application of the belief calculus, the computed risk assessments provides information about the uncertainty associated with the result of the analysis.

With respect to situations in which the outcome of a choice of one actor depends on the subsequent choice of another actor, the field of game theory [4] is highly relevant. Game theory provides strategies for making rational choices with respect to desirable and undesirable outcomes from the point of view of the different players/actors. These potential outcomes are described by a payoff structure in terms of the loss and gain to which the various players are exposed; a rational player will seek the outcome with the best payoff for herself. Not surprisingly, game theory can also be applied to analyze trust, as shown by Bacharach and Gambetta [1]. They explain how the trustor's choice to trust or not, and the trustee's subsequent choice to deceit or not, can be modeled in terms of this rational choice theory.

A formal model for trust in dynamic networks based on domain theory is proposed by Carbone et al. in [2]. Here, trust is propagated through delegation in a "web of trust", where the trust of one actor is affected by the trust of other actors. An important contribution of the approach is the distinction between a trust ordering and an information ordering. The former represents degrees of trust, while the latter represents degrees of precision of information from which trust is formed. An interval construction is introduced to capture uncertainty, and a simple trust policy language is proposed based on the formal model.

Acknowledgements We would like to thank Aida Omerovic (SINTEF ICT) and Anette Andresen (DNV) for participating in discussions and commenting on the work in the industrial project.

References

1. Bacharach, M., & Gambetta, D. (2001). Trust in signs. In *The Russel Sage Foundation Series on Trust: Vol. 2. Trust in society* (pp. 148–184). Thousand Oaks: Russel Sage Foundation.
2. Carbone, M., Nielsen, M., & Sassone, V. (2003). A formal model for trust in dynamic networks. In *Proceedings of the international conference on software engineering and formal methods (SEFM'03)* (pp. 54–61). New York: IEEE.
3. European Dynamics S.A. Functional requirements for conducting electronic public procurement under the EU framework (vol. 1). http://ec.europa.eu/idabc/servlets/Doc?id=22191, January 2005. Accessed: January 19, 2009.
4. Fudenberg, D., & Tirole, J. (1991). *Game theory*. Cambridge: MIT Press.
5. Gambetta, D. (1988). Can we trust? In *Trust: making and breaking cooperative relations* (pp. 213–237). Oxford: Blackwell.
6. Giorgini, P., Massacci, F., Mylopoulos, J., & Zannone, N. (2004). Requirements engineering meets trust management: Model, methodology, and reasoning. In *LNCS: Vol. 2995. Trust management* (pp. 176–190). Berlin: Springer.
7. Jøsang, A. (2001). A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *9*(3), 279–311.

8. Jøsang, A. (2007). Probabilistic logic under uncertainty. In *Proceedings of the thirteenth computing: the Australasian theory symposium (CATS2007). Conferences in research and practice in information technology (CRPIT)* (Vol. 65, pp. 101–110). Australian Computer Society.
9. Jøsang, A., Bradley, D., & Knapskog, S. J. (2004). Belief-based risk analysis. In *Proceedings of the Australasian information security workshop (AISW). Conferences in research and practice in information technology (CRPIT)* (Vol. 32, pp. 63–68). Australian Computer Society.
10. Jøsang, A., Keser, C., & Dimitrakos, T. (2005). Can we manage trust? In *Proceedings of the third international conference on trust management (iTrust), versailes* (pp. 93–107). Berlin: Springer.
11. Lysemose, T., Mahler, T., Solhaug, B., Bing, J., Elgesem, D., & Stølen, K. (2007). ENFORCE conceptual framework. Technical Report A1209, SINTEF ICT.
12. McNamara, P. (2006). Deontic logic. In *Handbook of the History of Logic: Vol. 7. Logic and the modalities in the twentieth century* (pp. 197–288). Amsterdam: Elsevier.
13. Object Management Group. UML Superstructure, V2.1.2. http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF, November 2007.
14. Ølnes, J. (2006). PKI interoperability by an independent, trusted validation authority. In *Proceedings of the 5th annual PKI R&D workshop: making PKI easy to use* (pp. 68–78). NIST.
15. Ølnes, J., Andresen, A., Buene, L., Cerrato, O., & Grindheim, H. (2007). Making digital signatures work across national borders. In N. Pohlmann, H. Reimer, & W. Schneider (Eds.), *ISSE/SECURE 2007 securing electronic business processes: highlights of the information security solutions Europe/SECURE 2007 conference* (pp. 287–296). Wirsbaden: Vieweg.
16. The European Parliament and the Council. Directive 2004/18/EC of the European Parliament and of the Council of 31 March 2004 on the coordination of procedures for the award of public works contracts, public supply contracts and public service contracts. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2004:134:0114:0240:EN:PDF, March 2004. Accessed: January 19, 2009.
17. Refsdal, A., Solhaug, B., & Stølen, K. (2008). A UML-based method for the development of policies to support trust management. In *Proceedings of the 2nd joint iTrust and PST conferences on privacy, trust management and security (IFIPTM'2008)* (pp. 33–49). Berlin: Springer.
18. Refsdal, A., & Stølen, K. (2008). Extending UML sequence diagrams to model trust-dependent behavior with the aim to support risk analysis. *Science of Computer Programming*, *74*(1–2), 34–42.
19. Sloman, M. (1994). Policy driven management for distributed systems. *Journal of Network and Systems Management*, *2*(4), 333–360.
20. Sloman, M., & Lupu, E. (2002). Security and management policy specification. *IEEE Network*, *16*(2), 10–19.
21. Solhaug, B., & Stølen, K. (2009). Compositional refinement of policies in UML – Exemplified for access control. Technical Report A11359, SINTEF ICT.