The SECURE Collaboration Model

S. Terzis, W. Wagealla, C. English, A. McGettrick, and P. Nixon

The Global and Pervasive Computing Group
Dept. of Computer and Information Sciences
University of Strathclyde

Abstract. The SECURE[1] collaboration model builds upon the previously developed theoretical trust and risk models and addresses the issues of managing the trust lifecycle. The theoretical aspects of the model concentrate on the following issues:

1. The interpretation of trust values in context for decision making purposes, namely trust exploitation. This process relies on the definition of context and the use of contextual parameterisation of collaboration requests.
2. The revision of trust values in the light of evaluated evidence about principals' behaviour, starting from complete lack of evidence in the case of unknown principals, namely trust formation and evolution. These processes rely on the separate processing of direct and indirect evidence and the introduction of the notion of attraction, which determines the effects that evidence has on the trustworthiness of a principal.

Trust exploitation, formation and evolution all are centered around a close relationship between trust and risk. In order to operationalise the theoretical collaboration model:

1. A trusting collaboration architecture is described. The architecture supports the decision making, the trust evaluation and the risk evaluation processes. It also places particular emphasis on the collection of evidence, requiring an *evidence gatherer* and an *evidence store*.
2. A layered model for the organisation of trust related information is introduced, namely the trust information structure. The structure enables processing of indirect evidence in a way that avoids the problems documented in literature.

Finally, the collaboration model is applied to two case studies, a smart space scenario and a e-purse scenario in order to demonstrate its various concepts. A comparison of the model to the state of the art and a discussion of areas for future work concludes this appendix.

1 Introduction

Global computing is characterised by large numbers of roaming entities and the absence of a globally available fixed infrastructure [47]. In such an environment

[1] SECURE (Secure Collaboration among Ubiquitous Roaming Entities, IST-2001-32486) is an EU FET Research Project funded under the Global Computing Initiative.

entities meet and need to collaborate with little known or even unknown entities. Entering any kind of collaboration requires entities to make security decisions about the type and level of access to their resources they will provide to their collaborators. In traditional environments with clearly defined administrative boundaries and limited entity movement security decisions are usually delegated to a centralised administrative authority [48, 34, 38]. In the global computing environment no single entity can play this role and as a result traditional techniques that statically determine the access rights of the entities are not an option. Entities are required to make their own security decisions. Moreover, the absence of a globally available security infrastructure means that these decisions need to be made autonomously. At the same time the sheer number of the roaming entities means that it is not feasible to gather and maintain information about all of them. Consequently, in the global computing environment decisions have to be made in the absence of complete knowledge of the operating environment.

Autonomous decision making with partial information is something that humans have to deal with on a day-to-day basis. To help them with the complexity of such a task humans have developed the notion of trust [18]. Although trust is an elusive concept and a number of definitions have been proposed for it, it is our belief that it can be modelled to adequate detail to facilitate security decision making in global computing.

The potential advantages of the notion of trust in dealing with security decisions have been recognised by a number of researchers as is demonstrated by research in Trust Management systems [1, 4, 6, 13, 27, 30, 42, 51, 52]. Although, this work is a move forward in security practice, most of it is based on the exchange of certificates between entities [4, 6, 27, 30, 42], and does not address the fundamental issue of what trust is made of and consequently the related issue of how trust can be formed. Furthermore, this approach provides very limited support for the evolution of trust between entities in the form of certificate revocation. As a result, this work lacks the support for autonomous decision-making and for dynamism in trust evolution necessary for global computing.

Novel approaches have been proposed to address the weaknesses of certificate based trust management [1, 13, 51, 52]. These approaches model explicitly the trustworthiness of entities and support its formation and evolution based on information gathered through personal interactions. As a result they are fundamentally different to other work in trust management. They no longer consider how to provide absolute protection against potential dangers, as is the case for all security decisions. Instead, they accept that dangers are an intrinsic part of any global computing system. Therefore, they attempt to use trust as a mechanism for managing these dangers and of learning from past interactions in order to improve protection. This fundamental change is reflected by the shift in the discussion from security decisions to trusting decisions.

Even these approaches, though, have certain weaknesses. First, they assume a global identification system for entities. This is a very strong assumption to make in the context of global computing. Second, few incorporate explicit notions of

risk (e.g. [16, 20]), and even in these cases the relationship between trust and risk is not made clear or is left for future consideration. Trust and risk are intrinsically related in the sense that there is no need for a trusting decision unless there is risk involved. Third, very few of these approaches model explicitly uncertainty, a consequence of decision making in the absence of total information. The SECURE approach addresses all these weaknesses by focusing on recognition rather than identification of entities, modelling explicitly risk (see section 2.2 below) and uncertainty in the trust domain (see section 2.1 below).

In this appendix we go one step further in the SECURE approach and we describe its collaboration model. The collaboration model addresses issues of trust lifecycle management, in particular the processes of trust formation, evolution and exploitation. Its theoretical aspects are founded on the theoretical trust model and the risk model. We introduce the notion of *attraction* to model the effects of new pieces of evidence on the trustworthiness of principals. Its operational aspects include a high-level collaboration architecture for trust lifecycle management and the trust information structure for the organisation of trust related information. A preliminary formalisation of the operational aspects encoded in the trust policy language provided by the formal trust model is also included.

In the rest of the appendix, we start with a brief overview of the SECURE theoretical trust model in section 2.1 and the SECURE risk model in section 2.2. This is followed by the description of our theoretical collaboration model, which first explores the relationship between trust and risk in section 3.1 and then discusses in detail our approach to trust exploitation, trust evolution and trust formation in sections 3.2 and 3.3 respectively. We then describe the operational aspects of our collaboration model, focusing in turn on a collaboration architecture in section 4.1, a structure for the management of trust related information in section 4.2 and a discussion on the issues of evidence gathering in section 4.3. Section 5 provides the formalisation of the operational aspect of our collaboration model. In order to facilitate the understanding of our proposed model in sections 6.1 and 6.2 we provide two case studies with the aim to demonstrate its application. Finally, we provide a comparison of our proposed model to the the state of the art in section 7 and conclude the appendix in section 8.

2 Trust and Risk in SECURE

2.1 The Theoretical Trust Model

The theoretical trust model [9–11] considers principals to be entities that either have to make trusting decisions or are the subjects of these decisions. $\mathcal{P}$ is defined as the set of all principals. Trust reasoning for principals has two aspects. On one hand decision making principals should be able to associate trust values to other principals. $\mathcal{T}$ is defined as the set of all these trust values. On the other hand, principals should also be able to update their trust values in the light of

evidence. So, given the set of principals $\mathcal{P}$ and the set of trust values $\mathcal{T}$ the global trust is defined as a function $m : \mathcal{P} \to \mathcal{P} \to \mathcal{T}$, where $m(a)(b) \in \mathcal{T}$ expresses a's trust in b.

Following an object-based model, the ability of each principal to reason about trust is modelled as a trust box, which has some internal trust state S and supports two operations:

- `update`: $\mathcal{S} \times \mathcal{E} \longrightarrow \mathcal{S}$, given a particular trust state $\mathcal{S}$ and some evidence $\mathcal{E}$, an updated trust state is produced.
- `trust`: $\mathcal{S} \times \mathcal{P} \longrightarrow \mathcal{T}$, given a particular state $\mathcal{S}$ and a principal $\mathcal{P}$, the trust value for the principal is returned.

The operation of each principal's trust box is described by a local policy function π, which relates principals to trust values and supports references [2]. References provide the ability for a principal to specify trust values as relations over the trust values of other principals. This local policy π is defined as:

$$\pi : (\mathcal{P} \to \mathcal{P} \to \mathcal{T}) \to \mathcal{P} \to \mathcal{T}. \tag{1}$$

The collection of all the local policies defines a global trust policy:

$$\Pi : (\mathcal{P} \to \mathcal{P} \to \mathcal{T}) \to (\mathcal{P} \to \mathcal{P} \to \mathcal{T}). \tag{2}$$

This global trust policy is interpreted in terms of complete partial orders. If the set of trust values $\mathcal{T}$ given an ordering relation $\sqsubseteq$ is a complete partial order (c.p.o.) with a least element $\bot$ (unknown), then the global trust m can be calculated as the least-fixed point of the global trust function Π. If Π is a $\sqsubseteq$-continuous function then the existence of the least fixed point is guaranteed . The ordering relation $\sqsubseteq$ could represent the preciseness of trust information according to its definition in [9]. Alternatively, it can be viewed as describing the amount of trust information as is the case in the example of collecting observations, or even to reflect the level of certainty in the trust values as is the case in the simple trust setting example (see also [9]). From the point of view of the calculation of the global trust function Π all these views are equivalent. The collection of additional information, further iterations in the least fixed point calculation, is in fact reducing the uncertainty about the trust values of referenced principals and leads to more precise trust information. In contrast, from a point of view of trust evolution the different views are not equivalent. The main difference between them is that trust information is monotonically increasing as additional evidence becomes available, while certainty is not. This is the case because additional evidence contradicting our current opinion may in fact reduce our certainty in the trust values. In this document we take the view that this relation refers to

[2] Note that in [9] the term delegation is used instead of reference to denote the same concept

level of certainty in the trust values and we refer to it as *certainty ordering*. We will come back to this issue in section 3.3.

In addition to the $\sqsubseteq$, "more certainty" ordering relation on $\mathcal{T}$, the model also defines $\preceq$, "more trust", which is equally essential. The $\preceq$ relation specifies for two trust values t_1 and t_2, which one expresses more trust. According to the theoretical trust model the set of trust values $\mathcal{T}$ given the ordering $\preceq$ is complete lattice.

In summary, the theoretical trust model requires that trust domain be defined as $(\mathcal{T}, \preceq, \sqsubseteq)$, where $\sqsubseteq$ is an *certainty ordering* and $\preceq$ is a *trust ordering* , such that $(\mathcal{T}, \sqsubseteq)$ is a c.p.o. with a least element and $(\mathcal{T}, \preceq)$ is a complete lattice.

2.2 The Risk Model

The risk model [3, 17, 45] considers each collaboration between principals as consisting of a number of trust mediated actions. Each such action is an interaction between two principals P_r and P_d, the *requester* and the *decision-maker* respectively [3], and has a set of possible results or outcomes. Each outcome has an associated risk. Risk is defined as the likelihood of an outcome occurring and the cost or benefit this outcome incurs if it occurs. The risk of an outcome depends on the trustworthiness of the requester P_r and certain parameters of the action in question. Before each trust mediated action the decision-maker P_d must make a trusting decision. This decision is based on the overall risk of the action in question, which is some kind of combination of the risks of all its outcomes.

In the risk assessment of a particular action, the risk model takes the view that the trustworthiness of the requester P_r affects the likelihood of the various outcomes and not their associated costs or benefits. These in turn are determined by the parameters of the action. For example, in the case of financial transactions the trustworthiness of the principals determines the chances of them paying their debts, while the debt amount determines the specific costs or benefits. At the same time, the risk model recognises that the outcomes can be distributed over a space that has a range of potential costs with corresponding probabilities. For this purpose it introduces the concept of a cost-PDF to represent the risk of each outcome, which is a probability density function with cost on the x-axis. Note that in cost-PDFs, benefits are represented as negative costs. Since the trustworthiness of the principal affects the likelihood of each outcome, the risk of an outcome is represented by a set of cost-PDFs parameterised by the principal's trustworthiness (a family of cost-PDFs). The family of cost-PDFs may be further parameterised by the parameters of the action.

At this point we should also point out a number of things:

1. The security decisions for each action are not necessarily binary, i.e. either accept or reject the action.

[3] Note that in previous documents the two principals have been referred to as the *initiator* and the *executor* respectively

2. For each action a number of outcomes may occur at the same time and as a result the decision making process needs to consider the risk of all these outcomes.
3. It might take a significant amount of time before the real outcome of an action is known.

3 The Theoretical Collaboration Model

The aim of the collaboration model is to capture the dynamic aspects of the trust model. These aspects address issues like how trust is formed, how it evolves over time, how it is exploited in the decision making process and are collectively referred to as trust lifecycle management. Our model builds upon the theoretical trust model and the risk model described above. In particular, it exploits the relationships between trust and risk to both facilitate and evaluate the decision making process of the principals.

We define collaboration as a joint interaction between a set of two or more principals $\mathcal{P}$ involving a set of one or more trust mediated actions $\mathcal{A}$. Before entering a collaboration each principal must make a trusting decision regarding the level of access to its resources it will permit to other principals. From this definition it should be clear that the collaboration and consequently the trusting decisions may be very complex. In this document we simplify our approach by only considering collaborations between two principals involving a single trust mediated action. We call collaborations of this type *simple*.

3.1 The Relationship between Trust and Risk

As it is observed in [3], *"there is no need to trust someone unless there is a risk involved"*. So, it is our premise that not only trust and risk mutually require each other but in fact are so closely related that they reflect each other.

There are two alternative views of the relationship between trust and risk. On one hand, we can view risk "driving" trust. According to this view, risk reflects how vulnerable we are in a particular situation, or in other words how likely is our current situation to lead to an accident or mishap, combined with the severity or cost of this accident or mishap. In this case, our aim is to protect ourselves by only exposing serious vulnerabilities to highly trusted collaborators. In this context the trusting decision we have to make can be expressed as: in a particular situation s, or in the context of a simple collaboration a particular action a which entails a level of risk r, how trustworthy should a principal be in order to be allowed to enter situation s or carry out action a? In this view the level of risk determines the necessary level of required trustworthiness, i.e. risk drives the decision making.

On the other hand, we can view trust "driving" risk. According to this view, trust reflects the likelihood of a principal behaving well in a particular situation.

In this case, our aim is to protect ourselves by only collaborating with principals that are likely to behave and as a result an interaction with them is not very risky. In this context the trusting decision we have to make can be expressed as: in a particular situation s, or a particular action a, involving a particular principal p, how much risk are we willing to accept by allowing principal p to enter situation s or carry out action a? In this view the level of trustworthiness determines the level of acceptable risk, i.e. trust drives the decision making.

In particular situations one or the other of these alternative views seem more appropriate. More specifically, it seems to be the case that the former view is more natural in a safety critical systems setting, while the latter in a financial systems setting. The SECURE risk model has adopted the latter view. This is also demonstrated by Figure 1, which is taken from [3].

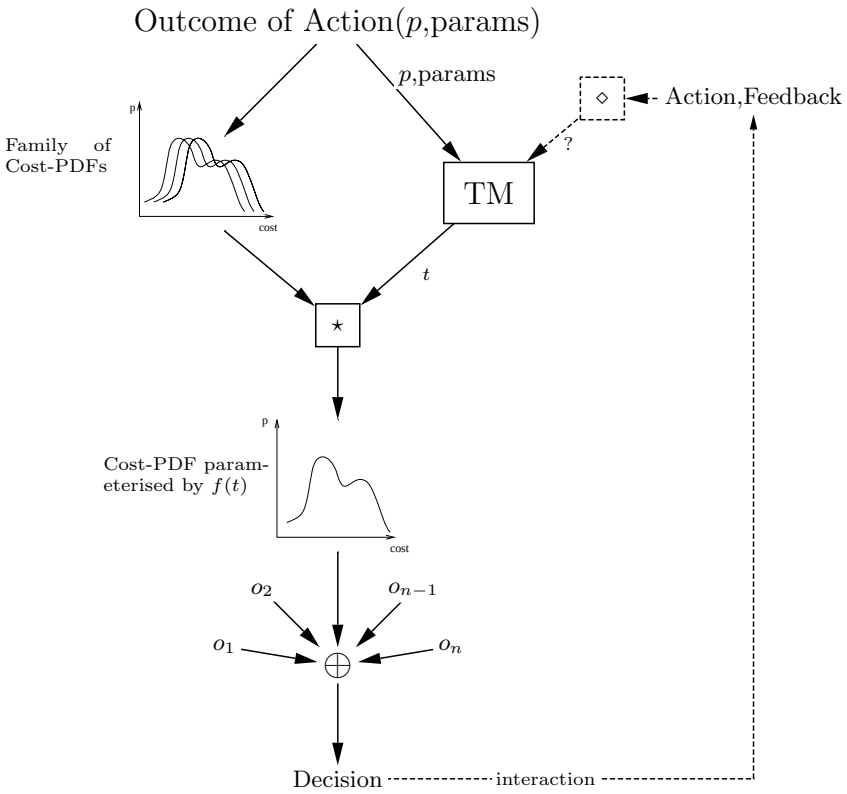


Fig. 1. The interface between trust and risk.

Figure 1 shows a first attempt at determining the nature of the relation between trust and risk from the point of view of the risk model. According to this figure for each outcome o_i of a particular action $\mathcal{A}$, a $\star$-function takes as input the family of cost-PDFs associated with o_i and the trust value t associated with a principal P and selects a single cost-PDF. The $\oplus$-function takes as input these selected cost-PDFs, one for each outcome o_i, and produces a decision for the action. If the decision was to go ahead with the action then after the completion of the interaction some feedback is produced. A $\Diamond$-function takes this feedback as input and provides some information to the trust model. Although in this figure the relationship between trust and risk is not very clearly specified, there are two important observations that we should note. First, the $\star$-function requires the existence of a mapping between trust values and cost-PDFs. Second, the feedback loop requires that the results of any action carried out by a principal be used as feedback regarding its trustworthiness.

Looking at the decision making process as is depicted in figure 1 combined with the adopted view of risk, it is clear that the decision process relies on the ability of the decision-maker to associate each principal to a risk profile. The selected cost-PDFs of the outcomes for each action describe this risk profile. Moreover, this profile can also be seen as a profile of how good or bad the behaviour of a principal is expected to be in the context of the requested action. In this sense, the trust values can be viewed as classifiers of principals, where each principal is classified according to its expected behaviour in one of a number of groups. There is one group for each trust value and all principals within a group have the same trust value. This view of trust values as classifiers of principals has significant benefits for the scalability of the decision making process. It allows the decision-maker to keep only one risk profile for each group of principals instead of one for each principal. This is particularly important in a global computing setting, where the number of principals is expected to be particularly high.

This approach dictates a very close relation between trust values and risk profiles. In fact, every trust value must be associated to a single risk profile. Additionally, two different trust values should be associated to different risk profiles. If this is not the case, then from a decision making point of view the two trust values and subsequently the two sets of principals associated with them will be indistinguishable. As a result, keeping both trust values is questionable. This approach requires that the mapping between trust values and risk profiles is not only a function but an injective or one-to-one function. Note that according to the risk model (see section 2.2), the risk profile of a principal for a particular action is described as a set of cost-PDFs one for each outcome of the action. These cost-PDFs are selected from the family of cost-PDFs associated to each outcome. This injective function that maps trust values to risk profiles corresponds to the $\star$-function of figure 1, which for each outcome of an action selects one cost-PDF. Considering *CPDFS* to be the whole family of cost-PDFs, we

define this function as [4]:

$$select : \mathcal{T} \rightarrow CPDFS \tag{3}$$

At this point, we should point out that the number of trust values and consequently risk profiles is dependent on the required granularity of the decision making process. The larger the number of trust values the more able the decision-maker is to discern variations in the expected behaviour of principals. This allows finer differentiation on the way principals are treated. However, there is a trade-off between the granularity and the complexity of the decision making process. A decision making process that exploits the finer differentiation in the treatment of principals is likely to be more complicated.

In a global computing environment characterised by the lack of complete information about principals, their classification into similarly behaving groups cannot be final. As additional information about the behaviour of individual principals becomes available the classification needs to be evolve. The results of this process may be twofold. It may lead to the re-classification of the principal into a different group whose associated risk profile is a more accurate predictor of the principal's behaviour. Additionally, it may even lead to a reconfiguration of the classification scheme by updating the risk profiles associated to each group. In this context, the two aspects of the evaluation process can be captured by the following questions:

- Has each principal been classified to the correct group, i.e. is the trust value for each principal correct?
- Is the risk profile associated to each group correct, i.e. are the cost-PDFs for each trust value correct?

Note that the answer to both of the above questions is with respect to the currently available evidence. Answering these questions in a general sense requires complete knowledge of a principal's behaviour. This is not feasible in most cases in a global computing setting. The former aspect of the evaluation process can be associated to the feedback loop of figure 1. We refer to this aspect of the process as *trust evaluation*. Moreover, it becomes clear that a second feedback loop to cost-PDFs is necessary to represent the latter aspect of the evaluation process. We refer to this aspect of the process as *risk evaluation*. Since we only focus on the dynamic aspects of trust and not of risk for the remainder of the discussion we will centre our attention on the trust evaluation aspect of the process.

Within this context, we can view the feedback collected from the multiple cases of an action as a profile of observed behaviour for the requester with respect to the action. This profile can in turn be described as set of cost-PDFs one for each outcome of the action, where the likelihood represents the proportion of the total occurrences where this outcome occurred. Then, we could rephrase the above trust evaluation question as follows:

[4] An alternative view of this relationship is to view each cost-PDF as a function parameterised by trust

– Which of the cost-PDFs predicts requester behaviour reflecting most closely the observed behaviour?

Being able to use the answer to this question to determine the appropriate trust value for the user requires an even stronger relationship between trust values and risk profiles. Not only should we be able given any trust value to select a risk profile, but we should also be able given any risk profile to select a trust value. This will enable us to calculate the trustworthiness conveyed by a profile of observed behaviour by finding the risk profile that most closely reflects it. This requirement implies that mapping from trust values to risk profiles should also be surjective or onto function too. As a result, the *select* function defined by equation 3 must be a bijection.

Introducing structure to the risk domain. So far in the discussion of the relationship between trust and risk, we ignored the structure of the trust domain, the set of trust values. According to our trust model the trust domain is defined as $(\mathcal{T}, \preceq, \sqsubseteq)$, a set of values together with a certainty and a trust ordering (see section 2.1). This, in combination with the fact that both the decision making and trust evaluation processes require a very close relationship between trust values and risk profiles, implies that the set of the risk profiles should reflect the structure to the trust domain.

In the case of the trust ordering ($\preceq$), if we consider in the context of a particular action a two trust values t_1, t_2 with respective risk profiles r_1, r_2 such that $t_1 \preceq t_2$ then r_2 must represent less risk than r_1. This means, according to our risk model, that outcomes with lower costs and/or higher benefits are more likely in profile r_2 than in profile r_1 (see section 2.2). Note though that this particular view is the result of risk model's premise that the trustworthiness of a principal only affect the likelihood of the outcomes. Instead, we can take the more general view that the trustworthiness of a principal can affect both the likelihood and/or the associated costs of the outcomes. In this case, we would expect that equally likely outcomes will be associated with lower costs in profile r_2 than in profile r_1. At this point we should note that in the current risk model, the case where there are multiple costs possible for the same general outcome (e.g. failure) can only be modelled with the introduction of a number of specialised outcomes (e.g. disastrous failure and recoverable failure). However, as the variety of costs for the general outcome increases this modelling approach becomes more awkward.

In the case of the certainty ordering ($\sqsubseteq$) things are more complicated. The source of the complication is the lack of a notion of uncertainty in our current risk model. There are three ways of addressing these complications:

1. Ignore the certainty dimension of the trust values in both the decision making and trust evaluation processes. In this approach if the trustworthiness of two principals differs only in terms of certainty then both principals will be treated the same. At the same time, the trust evaluation process will only

affect the trust dimension of the trust values leaving the certainty aspects either completely unaffected or managed through external procedures.

2. Consider the certainty dimension of the trust values only in the decision making process and not in the trust evaluation process. Following this approach the risk profiles reflect only the trust dimension of the trust values. As a result, the decision making process cannot rely exclusively on the risk profiles. Instead it also requires the trust values themselves in order to consider their certainty dimension. At the same time, the trust evaluation process still only affects the trust dimension of the trust values. Similarly to the first approach, this leaves the certainty aspects either completely unaffected or managed through external procedures.
3. Introduce a notion of certainty to the risk model, which will allow consideration of both trust dimensions in both processes. In this approach in contrast to the second one, the risk profiles reflect both trust dimensions. As a result, the decision making process can rely exclusively on the risk profiles. At the same time, the trust evaluation process considers and affects both trust dimensions. For example, as a result of the trust evaluation process the new trust value may be different only in terms of certainty and not in terms of trustworthiness.

The first approach is the least desirable of the three since it does not fully utilise the structure of the trust domain in either process. The second approach is a half way between the other two. On one hand, it does not ignore the certainty content of the trust values during decision making as the first one does. On the other hand, it still considers the certainty aspect as external to the trust evaluation process. As a result it still does not fully utilise the structure of the trust domain in the trust evaluation process. The third approach fully utilises the structure provided of the trust domain in both process. Moreover, it requires that the risk profiles reflect relationships between the respective trust values both in terms of trustworthiness and certainty. This requires a risk model that captures uncertainty.

From the three approaches, we consider the third one as the most desirable, mainly because of the requirements it places on the risk model. We believe that a risk model incorporating uncertainty is more in tune with the global computing setting that is characterised by high degrees of uncertainty about the collaborators. Consequently, we focus the rest of the discussion on the third approach.

Introducing uncertainty to the risk model. Our aim in this section is not to describe a full model for uncertain risks. It is more to suggest ways in which the current risk model can be extended to include uncertainty and to explain how some of these suggestions were applied in the case studies described in section 6.

We can introduce uncertainty to the risk model by considering risk ranges instead of specific risk values. A risk range can be seen as either a set containing a

number of distinct risk values, or provided that an ordering over set of risk values is defined [5], as an interval containing all the values between an upper and a lower bound. In either case, the higher the number of included risk values the more uncertain we are about the risk. As the number of included risk values is reduced our certainty about risk increases reaching complete certainty at the point when we have a specific risk value. In other words, we can compare risk ranges in terms of uncertainty using the set or interval inclusion operator on their included risk values. So, a risk range RR_1 is more uncertain than risk range RR_2 ($RR_2 \preccurlyeq RR_1$), if the set or risk values in RR_2 is a subset of the set of risk values in RR_1.

Regarding the exact meaning of a risk range we could consider it to be that all the included risk values are equally likely while all other risk values are considered totally unlikely. Note that in the constructive method of defining trust values as intervals on a lattice of basic trust values, our trust model models certainty in a similar way, as inverse set inclusion (see section 2.3 in [9]). Moreover, it uses the same meaning for the intervals, i.e. considering all included trust values as equally likely. We can easily extend our current risk model to incorporate the above approach. We can associate a range of cost-PDFs instead of a single cost-PDF to each outcome of each action for each trust value. In this way we can now easily reflect the full structure of the trust domain on the risk domain. We can see the certainty ordering of the trust domain as defining an inverse uncertainty ordering on the risk domain.

Following this approach results in some changes in the decision making and the trust evaluation processes. In the decision making process instead of considering a single risk profile for a principal we will have to consider a range of likely profiles. Any decision taken must acknowledge this fact. Furthermore, the trust evaluation process will have to decide on the appropriateness of the current trust value not only in terms of trustworthiness but also in terms of certainty. In terms of certainty the issue is whether any of the risk profiles of the range can be safely excluded (certainty increase) or if additional profiles need to be included (certainty reduction).

In section 2.2 we pointed out that risk is the combination of the likelihood of an outcome occurring and the cost it incurs. Taking this into consideration we can define two special cases of the above approach:

1. The case where the uncertainty is limited to the costs of the outcomes while their likelihoods are certain. In this case we could use a cost-PDF as defined in [3] to represent the range of likely costs of each outcome of an action for each trust value. This cost-PDF combined with the certain likelihood of the particular outcome give us the risk profile. In section 6.1 we apply this spe-

[5] In contrast to the trust values it seems more natural to define a complete order of risk values. For example, if we define risk values as the product of the likelihood and the cost of an outcome (both represented by real numbers), then the risk value ordering is just the $<$ relation on real numbers.

cial case approach to a smart space application scenario. More specifically, we consider a single action, get user location information. This action has a single outcome, loss of user privacy. The range of costs and benefits of this outcome are determined by the cost or benefit of a meeting with this particular user. We then follow the constructive method mentioned above to both determine the trust values and their associated risk profiles (cost-PDFs). The associated risk profiles of the trust value intervals are constructed by considering the corresponding risk profiles of the included basic trust values as equally likely, averaging of the respective cost-PDFs.

2. The case where the uncertainty is limited to the likelihoods of the outcomes while their costs are certain. In this case, we could represent the risk profile of each outcome of each action for each trust value as an interval of likelihoods, i.e. an interval within the (0, 1) interval, combined with the certain cost. In section 6.2 we apply this special case approach to an e-purse application scenario. More specifically, we consider a single action, e-cash payment, that has two outcomes, valid and invalid e-cash, each with a specific cost determined by the amount of the transaction. Users are considered reliable for transactions up to a certain amount. This means that there is no chance of invalid e-cash for any payments up to that amount. Users are considered unreliable for transactions above a certain amount. This means that there is no chance of valid e-cash for any payments above that amount. The trust values are intervals over the range of e-cash transactions determined by the thresholds of reliable and unreliable behaviour. The level of uncertainty is represented by the size of the interval. In the area of uncertain behaviour the likelihood of invalid and valid e-cash range from 0 to 1. In all cases the risk profile is parameterised by the amount of the transaction.

Concluding remarks. In conclusion, in order to support the decision making and the trust evaluation processes the relationship between trust values and risk profiles needs to be defined as a bijective function from trust values to risk profiles. We call this function *select*() (see equation 3). In fact *select*() is core to the decision making process, while its inverse $select^{-1}()$ is core to the trust evaluation process. Such a close relationship between trust values and risk profiles requires that the risk profiles reflect the structure of the trust domain, both in terms of the trustworthiness and certainty. Specialised examples of how this requirement can be satisfied are provided in the two case studies of section 6.

3.2 Trust Exploitation

Trust exploitation is defined as the interpretation of trust values in context. It acknowledges the situational character of trust. For example, Bob may be trusted to drive a car but may not be trusted to cook a decent meal. Moreover, Bob may be trusted to drive a car when he is sober, but not after a couple of drinks. We could probably think of a variety of situations like the given examples in which

our trust in Bob differs. The important observation is that if the trustworthiness of a principal varies depending on the situation, then the decision making process should take the current situation into account. So, in this section we discuss how various situations/contexts can be captured in a way that allows the proper exploitation of trust in the decision making process.

Before we delve into the discussion we should first define context. There are a number of definitions of context in the literature [7, 2, 37, 40]. In fact context-aware computing is a research area receiving a lot of attention in recent years [12, 15, 22, 35, 36, 39]. Dey's definition of context is one of the most widely used in the context aware computing literature. He defines context as:

Any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves[15].

From this definition it should be clear that there is a variety of different types of context we can consider. However, as Dey pointed out not all types of context are equally important [15]. The most important types are considered to be: location, identity, time and activity.

If we consider Dey's definition from a SECURE point of view, we can see that from the four important types identity is central. The trust model identifies principals and assesses their trustworthiness separately. Taking this into account, in SECURE we define context as:

Definition 1 *A situational modifier which affects the way principals behave.*

In general, the decision making process depends on the expected behaviour of a principal as it is reflected in its risk profile (see section 3.1). According to definition 1 context affects the behaviour of principals. In order for this effect to be taken into account in the decision making process, context should in fact affect directly or indirectly the risk profiles on which this process depends. Consequently, the effect of context can be threefold:

1. Context affects the risk analysis changing the risk profiles associated to each trust value. This results to different expectations for principals' behaviour.
2. Context affects the principals' trustworthiness. Since according to the risk model the trust values of principals parameterise the risk profiles, this results in context indirectly affecting the expectations for principals' behaviour.
3. Context affects the mapping of trust values to risk profiles, changing the risk profile selected by a trust value. This again results in context indirectly affecting the expectations for principals' behaviour.

At this point, we should note a couple of things. First, from the discussion in section 3.1 of the relationship between trust and risk, it should be clear that the above three cases refer to alternative ways in which context can affect the

operation of the *select*() function (see equation 3). In the first case the context affects the range of the function. In the second case it affects the domain of the function. In the third case it affects the calculation of the function. Second, since the risk model associates separate risk profiles to each trust mediated action, a separate family of cost-PDFs, actions already define a context for the principals' behaviour. In fact actions can seen as defining the activity type of context mentioned above. As a result, although the trust and risk models do not incorporate explicit notions of context, they already implicitly take into consideration two of the four important types of it. Moreover, the presentation of the *select*() function so far was in the single context of a single action. As a result it provided quite a simplistic view of its role. If we try to extend this view to include multiple contexts of a single action then we are faced with the challenge of how to capture the contextual variations of trust in a way that facilitates the above mentioned three cases. The challenge becomes more serious if we want also to consider a number of actions, as would be the case in most applications.

The contextual variations within a single action can be easily captured through the parameterisation of actions in our current models. In this case, for each action the application developer will have to determine a set of parameters such that their values can capture the whole range of contextual variations. In the simplest case, there could be a single parameter whose values represent all the different contexts. Although simple, this approach becomes difficult to manage in situations where the number of different contexts is large. An alternative approach would be to separate different types of context, e.g. time and location, and associate a parameter to each particular type. This approach is more manageable, since it separates the different contextual concerns.

The parameterisation of actions can support all three of the above cases. In the first case where context affects the risk profiles, we could use the action parameters to parameterise the family of cost-PDFs or the range of risk profiles associated to each action. In this case, we can refer to the parameters as risk parameters. In the decision making process the risk parameters would determine the exact nature of the cost-PDF family or range of risk profiles before the application of the *select*() function. The rest of the process will remain the same as was described in section 3.1. In the third case where the context affects the mapping of trust values to risk profiles, we could use the action parameters to parameterise the *select*() function. In this case, assuming that $CxtParams$ denotes the contextual parameters, the function will have to be redefined as follows:

$$select : \mathcal{T} \times CxtParams \rightarrow CPDFS \tag{4}$$

The second case, where context affect the trust values used by the *select*() function, requires the trust domain to somehow capture context. There are two alternative approaches:

1. Introduce context explicitly to the trust model. This approach requires each trust value to be contextualised, i.e. associated to a particular context $T[cxt]$. Consequently, the *update* function of the trust box (see section 2.1) will have to be redefined to take the context into account. In the decision making process the trust values provided to the *select*() function would be contextualised according to the action parameters, while the rest of the process will remain the same.
2. Exploit the internal structure of the trust values to capture context. In this approach the trust model is oblivious of context. Instead, the trust values are multi-dimensional reflecting the contextual differences of principals' trustworthiness. In the decision making process the trust values would have to be contextualised before they are provided to the *select*() function. Since, the various dimensions of the trust values reflect contextual variations, the contextualisation process can be seen as a projection function that selects only the relevant dimensions. The action parameters determine the exact nature of the contextualisation. In the decision making process, the trust values provided to the *select*() function would be contextualised, while the rest of the process will remain unchanged. Note that in the general case the contextualisation could be quite complex. For example, in the case where each dimension refers to different independent context, the contextualisation function will be a straight forward projection picking the appropriate dimension. While in the case where different contexts are dependent on each other, the contextualisation function will not only pick the the appropriate dimensions but also apply different weights to them. These weights will reflect how closely related the selected contexts are to the current one. Finally, each dimension may not reflect a particular context. Instead, it may reflect a particular aspect of principal behaviour, e.g. benevolence. These aspects will be relevant to different context with a different contribution. In this case the contextualisation will just apply different weights to the different dimensions reflecting their respective relevance for the current context.

These two approaches for the contextualisation of trust values can be extended to capture also contextual variations between actions. In fact, this only requires that the context considered for the contextualisation of the trust values includes the action too, and does not require any changes to the decision making process.

From the two approaches for the contextualisation of trust values we prefer the second one, since it does not require any changes to our trust model. However, the relative advantages and disadvantages of the two approaches are currently not very clear to us. This is an issue we are planning to investigate further in the future.

We should point out, that in both of the approaches for the contextualisation of the trust values, the strong relationship between trust and risk discussed in section 3.1 refers to contextualised trust and risk. However, if we take the view that certain contexts may be related, then the strong relationship between contextualised trust and risk should also reflect contextual relationships. Moreover,

the relationship also dictates that the trust evaluation process must be contextualised. This means, that if contextualised trust values are the input in the decision making process, i.e. only certain dimensions with a particular weight are considered, then the trust evaluation process should only affect these dimensions and to a degree that reflects their respective weights. This requirement introduces some serious complications to the trust evaluation process. So, in order to simplify matters the discussion in the following sections is restricted within the context of a single action and does not consider contextual variations with that action. The removing of these restrictions is an issue left for further investigation in the future.

The discussion so far, has been restricted within a single application context. If we consider multiple applications sharing trust information, then the applications themselves define the context within which their trust values need to be interpreted. We refer to contextual variations in behaviour within a single application as *internal context*, while we refer to contextual variations between applications as *external context*. In the case of external context, the transfer of trust values from one application to the other requires their contextualisation. This is achieved through a mapping between their respective contexts. This context mapping can be taken into account during the contextualisation of the trust values as discussed above. In fact, the projection of relevant dimensions of the trust values combined with their weighting is a form of internal context mapping. Note though that transferring trust values between applications is not just an issue of contextualisation. In the general case it will also involve a translation of trust values from one trust domain to another.

In summary, our trust exploitation model considers context as situational variations of principals' behaviour. It considers two types of context, internal (intra-application) and external (inter-application). The current model focuses on internal context, which it encodes by principal, action and action parameters. It considers the effects of context on the decision making process along three dimensions: risk profiles, mapping between trust and risk, and trust values. The last of the three dimensions is referred to as contextualisation of the trust values and is the basis for the extension of the model to multiple actions and even multiple applications. The model approaches the contextualisation of the trust values by exploiting their internal structure and defining an internal context mapping.

3.3 Trust Formation and Evolution

Trust formation refers to how evidence creates trust, while trust evolution refers to how trust values change in the light of new evidence. In our collaboration model we consider trust formation and evolution as similar processes. They both receive evidence, evaluate it with respect to the current trust value and produce a new trust value taking into account the results of the evaluation. Both processes may also include a phase of evidence collection, during which the

principal actively seeks additional evidence. There is only one difference in the case of formation compared to evolution:

- The current trust value is "unknown", i.e. the $\perp$ element of the certainty ordering [6].

If we look back into the theoretical trust model (see section 2.1) it should be clear that the update function of the trust box does in fact refer to the formation or evolution of the trust values. The state of the trust box is updated in the light of new evidence. Moreover, the trust evaluation process as described in section 3.1 illustrates also in part the trust evolution process.

Different types of evidence. In general, evidence refers to any kind of information about principals' behaviour. We call the principal in question the *subject* of the evidence. Evidence can be characterised as either *direct* or *indirect*. The former refers to evidence about a subject's behaviour that has been directly witnessed by a principal, the *witness*. The latter refers to evidence about a subject's behaviour that has not been directly witnessed. In fact, it refers to third party information about a subject's behaviour. From these definitions it should be clear that the characterisation of evidence as direct or indirect is relative, in other words direct evidence propagated to another principal becomes indirect. The distinction between direct and indirect evidence is quite important. The validity of direct evidence is unquestionable and should be treated as fact, while indirect evidence can be questionable and should be treated as an opinion about facts. Consequently the validity of indirect evidence depends heavily on the source of the information, the principal expressing the opinion. This difference suggests that during trust evolution indirect evidence must be treated with caution, as different principals may evaluate differently the same facts. Moreover, some principals may be even distort the facts.

The trust management literature identifies three types of evidence:

1. **Observations**. They refer to direct evidence. They are personal experiences usually gathered through interaction with a principal [7]. The action feedback of figure 1, refers in fact to the generation of observations that takes place at the completion of an interaction. According to our model, an interaction between two principals is in the form of trust-mediated actions. Each of these actions has a set of possible outcomes, each with its own costs or benefits (see section 2.2). At the same time, our model prescribes

[6] This does not have to be always the case. In the general case trust formation may start from an initial trust value $T \neq \perp$. In any case though, the initial trust value has to be quite uncertain, i.e. it has to be close to the bottom of the certainty ordering complete partial order.

[7] An extended view of observations may also include observables, indications of a subject's behaviour while interacting with another principal.

that differences in the trustworthiness of principals represent differences in their expected behaviour profiles, i.e. their risk profiles. These differences demonstrate themselves as either differences in the likelihoods of the various outcomes and/or differences in the incurred costs or benefits. Hence, we take the view that an observation is the pair of the outcome that occurred at the end of the interaction and its observed cost or benefit. This approach allows us to distinguish between profiles of expected behaviour that produce the same outcomes with same likelihood, but incur different costs or benefits. For example, consider an email spam filtering application that decides if an email should be classified as spam or not depending on the trustworthiness of the sender. In this scenario, we have two different principals P_1 and P_2, that send us spam messages with same likelihood, e.g. 1 in 4 messages is spam. But principal P_1 usually sends us small text messages a few bytes in size, while principal P_2 usually sends us messages with attachments many megabytes in size. In this case, the risk associated to messages from principal P_2 is significantly higher to that associated to messages from principal P_1. So, in this scenario the evaluation of our interactions with each principal needs to consider both the outcome, spam or not, and its cost, message size.

2. **Recommendations**. They refer to indirect evidence passed between a principal W, the witness, and a principal R, the receiver, describing a judgement on principal S, the subject. In the general case, this evidence can take any form, but in this document we only consider the case of a trust value expressing W's trust in S.
3. **Reputation**. It refers also to indirect evidence that takes the form of a measure of the overall trustworthiness of a subject S. This measure can be expressed as:

$$r(S) = \sum_{P_i \in C} m(P_i)(s) \tag{5}$$

Note that we only assume a community of principals C, which is a subset of the whole principal population $\mathcal{P}$. This is in line with the lack of complete information characterising global computing.

Exchanging trust values between principals. Both recommendations and reputation are based on the ability of principals to exchange trust values. For any such exchange to be meaningful the trust values need to share a common representation and all principals need to have a shared understanding of their meaning. If we consider the variety of potential application scenarios each with its specific requirements for trust values (see [8]), it becomes very difficult to come up with a common trust domain for all of them. So, in line with the discussion on the contextualisation of the trust values in section 3.2, we assume that each application sets the boundary within which the common format and the shared understanding of the trust values is guaranteed. For exchanges of trust values over this boundary, i.e. between different applications, we assume that there are no guarantees.

At this point, it is imperative to clarify what do we mean by a shared understanding of the trust values. A strict interpretation of shared understanding would require that provided with the same trust value in the same context, all principals would make the same decision. Such an interpretation would require that besides the same format of trust values, all principals share:

- The same structure of the trust value domain.
- The same trust contextualisation process.
- The same *select*() function (see equation 3).
- The same risk analysis, i.e. risk profiles.
- The same access control policy.

We find such an interpretation unnecessarily over restrictive, particularly in the context of global computing, which is characterised by the autonomy of entities. In fact, it could be argued that the access control policy and the risk analysis are characteristic of the disposition of an entity and as a result should vary. The argument regarding the contextualisation process and the *select*() function is not as clear cut. It should be clear however, that a shared structure for the trust value domain is the minimum requirement for the meaningful exchange of trust values between principals. Hence in this document we take the view that all principals share the same structure of the trust value domain. This means the following:

1. There is a single format of trust values.
2. Each dimension of the trust values represents the same thing for each principal. For example, if the trust values are (t_1, t_2), then t_1 always refers to the trustworthiness of a principal as a collaborator and t_2 always refers to the trustworthiness of a principal as a recommender.
3. The value range of each dimension of the trust values is the same for all principals. In the above example we could have that for all principals both t_1 and $t_2 \in (0, 1)$.
4. The trust and certainty orderings are the same for all principals. So, if $t_1 \preceq t_2$ and $t_1 \sqsubseteq t_2$ for principal P_i then $t_1 \preceq t_2$ and $t_1 \sqsubseteq t_2$ for all other principals.

The above assumptions guarantee to a certain degree meaningful exchange of evidence in the form of trust values between principals within a particular application. Note, that in particular application scenarios the developer may want to provide stronger guarantees. In general, it is the case that the stronger the guarantees the more meaningful the exchange of trust values becomes. In any case, it is imperative that the application developers are very clear about the provided guarantees. Moreover, the exploitation, formation and evolution processes should not make any assumptions above these guarantees.

The value of evidence. The three different types of evidence, although all valuable for trust formation and evolution, do not have the same value. In general, we would expect direct evidence to be a lot more valuable than indirect

evidence due to its unquestionable character. This means that we would consider observations to be the most valuable type of evidence and as a result to carry the most weight in the evolution process. In fact, we could even see situations where observations would be the only type of evidence considered. The problem with observations is that they require the participation of the witness, making them the most difficult type to collect. It usually takes a lot of time before any principal acquires adequate personal experience.

The exchange of experiences between principals can enhance their perception of the world, especially in cases where personal experience is limited. In these cases recommendations and reputation can be particularly valuable. However, their value is predicated on the assumption that the subject is likely to behave similarly towards both the witness and the receiver. If the assumption does not hold, then the exchanged experiences are worthless. In the general case, there are no guarantees that this assumption holds. Identifying which principals witness similar behaviour from certain subjects is in most cases very difficult. In any case, the fact that exchanged experiences are indirect types of evidence means that the trustworthiness of their source affects their validity and as a result their value. This is acknowledged in the literature and has led to the introduction of the concept of discounting for recommendations [26, 28, 50]. Discounting usually takes the form of an operator which considers the receiver's perception of the integrity or trustworthiness of the witness, namely the *trust in the recommender*. We should note here that this type of trust is usually considered as separate from the trust that is normally associated to principals. It reflects how good the principal is as a source of recommendations, rather than how likely it is to behave well. The value of a recommendation as evidence depends on the certainty of our trust in the recommender. The more certain it is the more valuable the recommendations are. If we are completely certain about a recommender's trust then its recommendations become as valuable as our observations.

There are two alternative approaches to modelling trust in the recommender:

1. As a proper trust value. According to this approach exchanging recommendations is just another form of interaction with a principal and as a result a similar decision making and trust evaluation processes should be followed. In this case, the trust in the recommender is maintained with similar exploitation, formation and evolution processes to the main trust. This type of trust in the recommender could form either an orthogonal second trust value domain, or a separate dimension of the main trust value domain. In either case, these trust values should encode both trustworthiness and certainty and follow the same structure as the main trust value domain in terms of the orderings (see section 2.1).
2. As a specific measure of integrity that is not a proper trust value. According to this approach the whole issue of exchanging recommendations is completely orthogonal to the other types of interaction between principals. As a result, we have completely separate processes for its management in which

viewing trust in the recommender as a proper trust value only introduces unnecessary complexity.

We believe that the former approach is more desirable since it allows us to consider trust in the recommender and main trust within a single framework. It also lets us to explicitly model certainty in the recommenders.

This approach, however, raises a number of issues regarding the discounting operator that should adjust recommendations in the light of the trust in their source. First, using the term discounting for such an operator is misleading. This operator aims to adjust the trust values we receive as recommendations, either by lowering or raising them. More specifically, if we notice that particular principals tend to be quite lenient in their evaluations of subject behaviour, then we should lower their recommendations. While, if we notice that they are quite strict, then we should raise them. Note that this is in line with the notion of semantic distance between recommendation and experience as defined in [1]. So, we refer to such operator as an *adjusting operator* and to the process of applying the operator as *recommendation adjustment*.

Second, the fact that the trust values and the trust in the recommender encode both trustworthiness and certainty, means that the adjusting operator should also work on both these dimensions. In this case the operator could also been seen as the means for making recommendations as valuable as observation. However, at present the exact way in which the operator should work is not very clear. Consequently, this is an issue open for further research. We should note, though, that we incorporate the adjusting operator in our model. We introduce recommendation adjustment as the first step in their evaluation process.

Although, the use of an adjusting operator that takes into consideration the trust in the recommender may be used to increase the value of recommendations, in the case of reputation things are even more complicated. Since, reputation aggregates a number of recommendations in order to follow the same approach we need to adjust each constituent recommendation. This requires that we know exactly which opinions were combined to produce the measure of overall trustworthiness, who were the sources of these opinions and how much each of them contributed to the overall measure. If this is the case, then reputation has the same value as individual recommendations. However, in most cases, this level of detailed information is not available, and as such reputation is not very valuable for evolution. This is the main reason why reputation is not further considered in this document and the SECURE project as a whole.

Evidence evaluation. Evidence evaluation is the first step of the trust formation and evolution process. In order to model the effects of new evidence on our current trust value we introduce the notion of *attraction*. When new evidence becomes available we expect the current trust value T_{curr} to change to a new trust value T_{new} that somehow reflects the new evidence. In other words

we expect the new evidence to attract the current trust value towards it. Thus, attraction is the measure of the impact evidence has on the current trust value.

In this context, we can view the evidence evaluation process as a function, which assuming that *Evd* is the set of evidence, and *Attr* is the set of attractions, is defined as follows:

$$evaluate : Evd \times \mathcal{T} \rightarrow Attr \tag{6}$$

Since, according to the theoretical trust model our trust values reflect both trust and certainty, we can express the impact of attraction in both trust and certainty terms. Therefore, we can view attraction as a two-dimensional measure consisting of a trust dimension (τ) and an certainty dimension (σ). On each dimension attraction can be characterised:

- In the certainty dimension as either *reinforcing* or *contradicting*. In the former case, the new evidence cannot increase the certainty of the current trust value, i.e. $T_{curr} \sqsubseteq T_{new}$. In the latter case, the new evidence cannot reduce the certainty of the current, trust value, $T_{new} \sqsubseteq T_{curr}$.
- In the trust dimension as either *positive* or *negative*. In the former case, the new evidence cannot reduce the trustworthiness of the current trust value $T_{curr} \preceq T_{new}$. In the latter case, the new evidence cannot increase the trustworthiness of the current trust value $T_{new} \preceq T_{curr}$.

We refer to the above characterisation of attraction as the *direction of the attraction*. The reason for this is that if we consider a trust domain $(\mathcal{T}, \preceq, \sqsubseteq)$, then these characterisations are excluding a number of elements of $\mathcal{T}$ producing a subset of acceptable trust values in terms of certainty and trust respectively. So, if we define $T_\sigma, T_\tau \subseteq \mathcal{T}$ to be the set of acceptable trust values in terms of certainty and trust respectively, then the characterisations dictate that the new trust value must belong to the intersection of these sets, $T_{new} \in T_\sigma \cap T_\tau$. Or in other words, they determine the direction we should move on the trust ordering lattice or the certainty ordering c.p.o. (see section 2.1) to find our new trust value T_{new}.

Recommendation evaluation. Firstly, we should remind the reader that recommendations are in the form of trust values. Note that we assume that in the case of recommendations, the adjustment process has already taken place. As a result, we can evaluate a recommendation, *Rec*, by taking advantage of the structure of the trust domain $(\mathcal{T}, \preceq, \sqsubseteq)$ and directly comparing it to the current trust value in terms of certainty, $\sqsubseteq$, and trust, $\preceq$. This comparison will determine the direction of its attraction as follows:

- In terms of certainty, we calculate the greatest lower bound (glb) of T_{curr} and *Rec*. Note that because $(T, \sqsubseteq)$ is a complete partial order with a least element, the glb of any two trust values $t_1, t_2 \in T$ is guaranteed to exist. Then, there are three cases:

1. If the $glb(T_{curr}, Rec) = T_{curr}$, then the attraction of the evidence is reinforcing.
2. If the $glb(T_{curr}, Rec) = Rec$, then the attraction of the evidence is still reinforcing, but in this case Rec does not really add anything to our current trust value and can be safely ignored.
3. Otherwise, the attraction of the evidence is contradicting.

In both the first and the third of these cases the new trust value T_{new} must have the properties: $glb(T_{curr}, Rec) \sqsubseteq T_{new}$ and $T_{new} \sqsubseteq T_{curr}$.

– In terms of trust, T_{curr} and Rec are either comparable or incomparable.

1. If they are comparable, then:
   - If $T_{curr} \preceq Rec$, then the attraction of the evidence is positive.
   - If $Rec \preceq T_{curr}$, then the attraction of the evidence is negative.

   Note that in this case the new trust value T_{new} must be inside the interval $[T_{curr}, Rec]$ or $[Rec, T_{curr}]$ respectively.
2. If they are not comparable, then instead of comparing T_{curr} and Rec we compare T_{curr} to either the glb or the least upper bound (lub) of T_{curr} and Rec. Note that because of the fact that the $(T, \preceq)$ is a complete lattice, both the glb and the lub of any two trust values $t_1, t_2 \in T$ are guaranteed to exist. The choice between the glb and lub is a dispositional characteristic of the principals. According to this characteristic principals are classified as either *trusting*, those selecting the lub, or *distrusting*, those selecting the glb. Then there are the following two cases:
   (a) If the glb was selected, then the attraction of the evidence is negative and the new trust value T_{new} must be inside the interval defined by the glb and the current trust value T_{curr}, $T_{new} \in [glb(T_{curr}, Rec), T_{curr}]$.
   (b) If the lub was selected, then the attraction of the evidence is positive and the new trust value T_{new} must be inside the interval defined by the current trust value T_{curr} and the lub, $T_{new} \in [T_{curr}, lub(T_{curr}, Rec)]$.

The above determines the direction of the attraction, but it does not determine the *value of attraction*, $\|attr\|$. Our trust model does not provide us with a measure of distance between trust values, and as a result cannot be used for determining the value of the attraction. However, in section 3.1 we required a very close relationship between trust values and risk profiles. We also defined a *select*() function (see equation 3), which maps trust values to risk profiles. Here, we exploit the *select*() function combined with the definition of a notion of *risk profile distance* to determine the value of attraction. More specifically, assuming two discrete cost-PDFs C_1 and C_2, representing risk profiles R_1 and R_2 respectively we define their distance as:

$$diff(C_1, C_2) = \sum_{\forall x \in \mathbb{C}} \mid \Pr_1(x) - \Pr_2(x) \mid \tag{7}$$

where $\Pr_1(x)$ and $\Pr_2(x)$ are the probabilities of cost x according to C_1 and C_2 respectively, and $\mathbb{C}$ is the range of potential costs and benefits. In this case,

the value of a recommendation's *Rec* attraction, must be proportionate to the distance of the risk profiles, cost-PDFs, associated to T_{curr} and *Rec*:

$$\|attr\| \propto diff(select(T_{curr}), select(Rec)) \tag{8}$$

The exact function for the calculation of the value is up to the application developer to define.

In summary, the direction of the attraction determines the direction of movement on the trust ordering complete lattice and the certainty ordering c.p.o. While the value of the attraction determines the distance of movement. The bigger the attraction value is the bigger the movement has to be. In other words, the bigger the value of the attraction the more likely we are to move away from the current trust value.

Observation evaluation. At first, we should remind the reader that observations comprise of the observed outcome and its incurred cost or benefit. The evaluation of observations in general can take the following two forms:

1. *Direct Evaluation.* In this case the attraction of an observation is characterised as positive if the observed outcome incurred a benefit and negative otherwise. Further, it is characterised as reinforcing if the likelihood of the observation, *Obs*, according to the risk profile of the current trust value, $\Pr_{select(T_{curr})}(Obs) \geq 50\%$. Otherwise, it is contradicting. Its value should be proportionate to the distance of the likelihood of the observation according to the risk profile of the current trust value from 50%,

$$\|attr\| \propto \mid \Pr_{select(T_{curr})}(Obs) - 50\% \mid \tag{9}$$

   This form of observation evaluation is demonstrated in the e-purse scenario (see section 6.2).
2. *Indirect Evaluation.* In this case, we first produce an *evidential trust value*, T_{evd}, from the observation. Then we evaluate the attraction of T_{evd} following the approach described above in the evaluation of recommendations[8]. The production of the evidential trust value exploits the close relation between trust and risk, and in particular the fact that *select*() is a bijective function (see section 3.1). More specifically, we choose as T_{evd}, the trust value, T_i that is associated to the risk profile, R_i, in which the observation, *Obs* has the highest likelihood:

$$T_{evd} = select^{-1}(R_{max} \mid \Pr_{R_{max}}(Obs) = \max_i(\Pr_{R_i}(Obs))) \tag{10}$$

   If we follow this approach, then it should clear that considering a single observation means that T_{evd} does not offer significant insight into the trustworthiness of the principal in question. Therefore, it seems reasonable to

[8] Note that we can take the view that the recommendation adjustment does in fact produce an evidential trust value, which is subsequently used in the evaluation process.

collect a number of observations before the evaluation takes place. The collected observations construct a profile of observed behaviour. In this profile, the number of occurrences of each particular observation over the total number of observations under evaluation determines its likelihood. This profile of observed behaviour can be compared to the various profiles of expected behaviour, i.e. the risk profiles. The trust value associated to the risk profile closest to the observed one, according to the risk profile distance (see equation 7), is the evidential trust value. This form of observation evaluation is demonstrated in the smart space scenario (see section 6.1).

Updating the current trust value. The final step in the formation and evolution process is to update the current trust value T_{curr} to a new trust value T_{new} according to the attraction of the evidence. This process is carried out by an *evolve*() function. Following a similar approach to the one described in [23], there are two alternative definitions for such a function:

1. As a trust evolution function that considers a sequence of attractions in order to produce a trust value. More precisely, assuming that $AttrSeq$ represents sequences of attractions:

$$evolve : AttrSeq \rightarrow \mathcal{T} \tag{11}$$

   Considering sequences of attractions instead of sets allows us to define trust evolution functions that can distinguish the past and have fixed memory. For example, we can introduce time discounting of evidence, so that more recent evidence counts for more, or we can drop evidence if it is considered to distant in the past.
2. As a trust update function that considers the current trust value and an attraction in order to produce a new trust value. More precisely, assuming that $Attr$ is the set of attraction values:

$$evolve : \mathcal{T} \times Attr \rightarrow \mathcal{T} \tag{12}$$

   Trust update functions have infinite memory, since all past evidence is reflected in the trust values. This is the main reason why both application scenarios in section 6 use a trust update function. In fact, in the e-purse scenario (see section 6.2) the *evaluate*() and the *evolve*() function have been merged into a single function that provided with an observation, directly produces the new trust value from the current one. Note also that for any trust update function we can generate a trust evolution function by iteration starting from each initial trust value.

The exact nature of either type of *evolve*() function is up to the application developers to define. When defining the exact function they should consider the work of Jonker and Treur [23], who analyse trust evolution and update functions and identify a number of properties that such functions may have. More

importantly, each of the identified properties allows the modelling of alternative principal attitudes towards trust. This type of modelling can also be applied to our trust formation and evolution approach.

Following a similar approach to [23], in our collaboration model we identify two aspects that allow us to characterise the many types of principal attitudes towards trust. We refer to these aspects as the *dispositional characteristics* of a principal. These characteristics are:

1. *Trusting Disposition*. In terms of trusting disposition, principals are classified as either *generally trusting* or *generally distrusting*. This is reflected in the initial trust value that they use for the formation process and the selection of the glb or lub in the case of incomparable trust value during evidence evaluation (see the section on recommendation evaluation above). With respect to the former, a generally trusting principal would select an initial trust value T that conveys more trust than "unknown", $\bot \preceq T$, while a generally distrusting principal one that conveys less trust than "unknown", $T \preceq \bot$. With respect to the latter, a generally trusting principal would select the lub, while a generally distrusting principal the glb.
2. *Type of trust dynamics*. The types of trust dynamics reflect how easily a particular principal's trust in others builds and erodes in the light of evidence. In general, principals may build or erode trust either quickly, slowly, or in balance. This means that a principal that quickly (slowly) builds trust would require a small (large) number of positive evidence to consider another principal as highly trusted. Further, a principal that quickly (slowly) erodes trust would require a small (large) number of negative evidence to consider another principal as highly distrusted, meaning that it is quite unforgiving (forgiving) of bad behaviour. Jonker and Treur suggest in [23] such a model of different types of trust dynamics. Moreover, they identify the following types:
   - Blindly positive. A principal that after a number of good experiences with another principal will always consider it trustworthy.
   - Blindly negative. A principal that after a number of bad experiences with another principal will always consider it untrustworthy.
   - Slow positive, fast negative. A principal that requires a large number of good experiences to build trust and a small number of bad experiences to erode trust.
   - Fast positive, slow negative. A principal that requires a small number of good experiences to build trust and a large number of bad experiences to erode trust.
   - Balanced slow. A principal that requires both a large number of good experiences to build trust and a large number of bad experiences to erode trust.
   - Balanced fast. A principal that requires both a small number of good experiences to build trust and a small number of bad experiences to erode trust.

In order to enable the expression of the above dispositional characteristics in our collaboration model, we introduce *dispositional parameters* in the *evolve*() and *evaluate*() functions. The exact nature of these parameters will of course depend on the exact definition of these functions and the characteristics of the particular application scenario. Therefore, it is up to application developers to define them. The use of dispositional parameters is demonstrated in both case studies in section 6. In the smart space scenario, we use a dynamic threshold *dt*, which determines how quickly or slowly the system adapts to evidence (see section 6.1). In the e-purse scenario, we use α_p, α_n and β that determine how slow or fast trust builds, erodes and becomes certain respectively.

Summary. In summary, trust formation and evolution are considered very similar. Both consist of an evidence evaluation process and an evolve trust process. The evaluation of all evidence is in terms of its attraction. Attraction is calculated in reference to the current trust value and exploits the structure of the trust domain and the notion of risk profile distance. Furthermore, the evidence evaluation process takes into account the type of evidence, direct or indirect, and requires that all indirect evidence is adjusted in accordance to the trustworthiness of its source, thus introducing the notion trust in the recommender. The evolve trust process could either take the form of a trust update or a trust evolution function. The former produces a new trust value based on the current one and the attraction of the evidence, while the latter just considers a sequence of attractions. In both cases, the functions can be parameterised in terms of trusting disposition and types of trust dynamics, allowing the expression of a variety of principal attitudes towards trust.

4 The Operational Collaboration Model

In section 3 we discussed the theoretical aspects of our collaboration model, which incorporates trust exploitation, formation and evolution. The model introduced a number of processes that need to be carried out in order support trust based decision making and evaluation. In this section, the discussion moves from the theoretical to the operational considerations of the model. In particular, in section 4.1 we present a trusting collaboration architecture that supports our theoretical model. A central component of the architecture is a store where trust related information is kept. In section 4.2 we present a layered model for the organisation of trust related information. This model allows us to avoid some of the problems of managing indirect evidence reported in literature [1, 25, 28, 52]. Finally, in section 4.3 we are discussing some issues relating to collection and distribution of evidence.

4.1 A Trusting Collaboration Architecture

In order to identify the components of our trusting collaboration architecture, we start by focusing on the main processes identified in section 3.1, and particularly decision making and trust evaluation. Our aim is to elaborate on figure 1 and to show where the various processes introduced in section 3 should be positioned against each other. Due to the complexity of the decision making and the trust evaluation process, we have separated the two processes in figures 2 and 3 respectively.

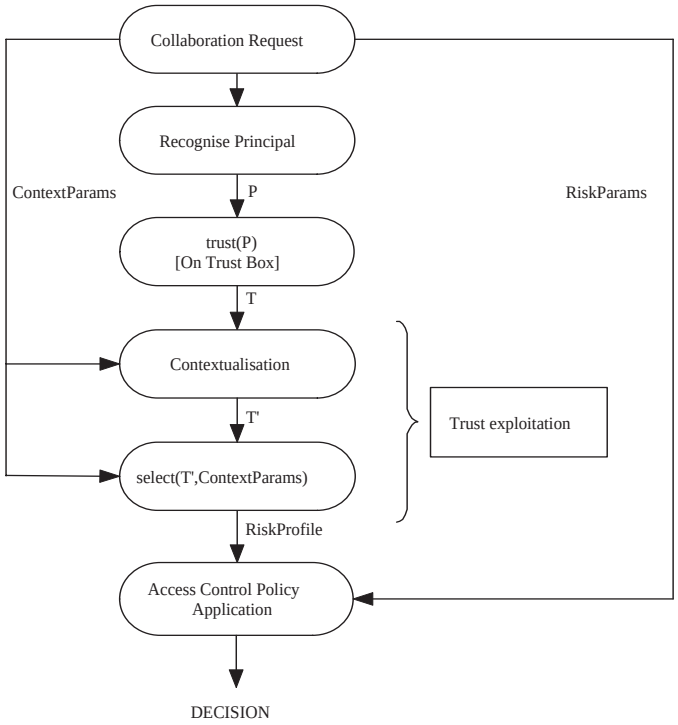


Fig. 2. The decision making process.

According to figure 2 the decision making process is triggered with the receipt of a collaboration request. The request should include all necessary information to both determine who is the requestor and the context of the collaboration. The information included in the request is divided into requestor related and context related information. The former allows us to recognise the requesting principal. While, the latter will be divided into parameters for the trust contextualisation, the *select*() function and the risk analysis.

Having recongised the requesting principal we retrieve its trust value using the *trust*() operation on the trust box (see section 2.1). The retrieved trust value is then contextualised taking into account the relevant context parameters. The contextualised trust value (T' in the figure) together with the relevant context parameters is used to select the appropriate risk profile. The selection is carried out by the *select*() function (see equation 4). Finally, the access control policy of the decision making principal is applied to the selected risk profile adjusted by the risk parameters and a decision is made.

Comparing figure 2 to figure 1 we note the following:

- The process of interpreting the trust value for the requesting principal in the context of the collaboration request and then using it to select the appropriate risk profile relates to the $\star$-function.
- The process of applying the access control policy to the selected risk profiles adjusted by the risk parameters relates to the $\oplus$-function.

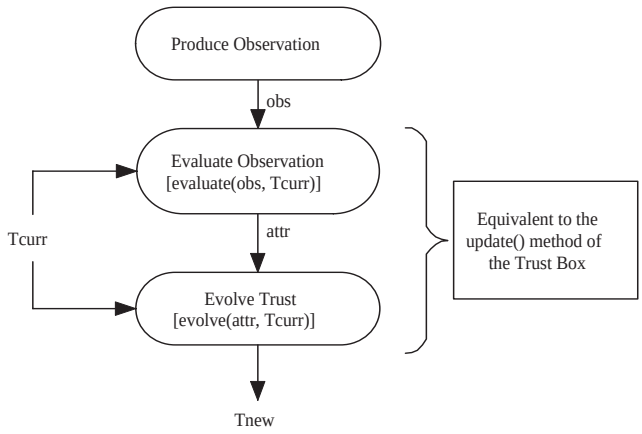


Fig. 3. The trust evaluation process.

According to figure 3 the trust evaluation process is triggered with the production of an observation. Note that the production of the observation requires that a collaboration request was received, the decision was made to take part in the collaboration, which has now been completed and its results are known. In general, a collaboration may last for some period of time and its result may not be known immediately after its completion. Following its production the observation needs to be evaluated with respect to the current trust value T_{curr}. The evaluation is carried out by the *evaluate*() function and is either direct (see equation 6) or indirect (producing an intermediate evidential trust value T_{evd}). In either case it produces the attraction of the observation. Then, the attraction is used to produce a new trust value T_{new}. The production of the new trust value

is carried out by an *evolve*() function which can take the form of either a trust evolution or a trust update function (see equations 11 and 12 respectively).

At this point we should note that the trust evaluate process is represented by the feedback loop and the $\Diamond$-function in figure 1. Moreover, the combination of the *evaluate*() and the *evolve*() function is equivalent to the *update*() function of the trust box (see section 2.1).

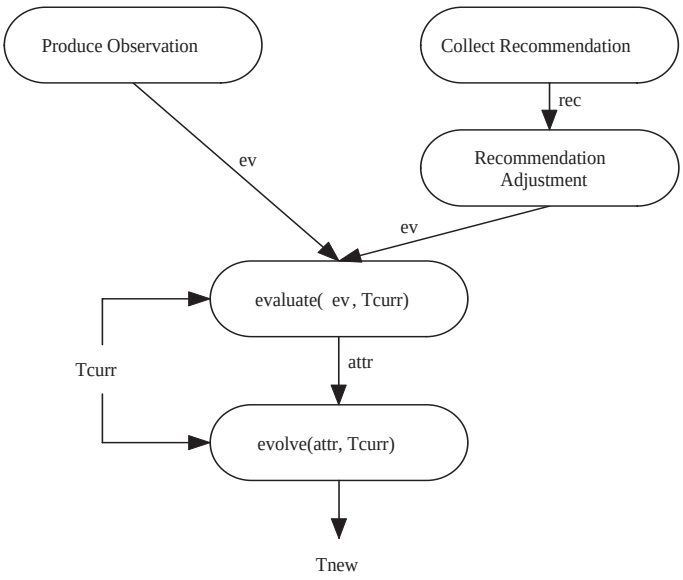


Fig. 4. The trust formation and evolution process.

Figure 3 focuses only to the evolution of trust in the light of observations. In figure 4 we extend this figure to include also recommendations. In this case, either the production of observations or the collection of recommendations triggers the trust evolution process. The collected recommendations have to be adjusted, taking our trust in the recommender into account, before they are evaluated. The rest of the process is the same for both recommendations and observations. At this point, we should note that in the case of trust formation, there are some differences in the process. First, the process is triggered by the trust box realising that there is no current trust value (T_{curr}) for the principal in question. At this stage we set T_{curr} to be the the initial trust value (most likely $\perp$, the bottom element of the certainty ordering c.p.o. according to section 2.1). Second, we either provide as the new trust value the recently set current trust value, i.e. $T_{new} = T_{curr}$, or we initiate a recommendation collection process. In the latter case, the formation process is almost identical to the process depicted in figure 4, focusing though only on the collected recommendations. Note that since we have

already set the current trust value (T_{curr}), then rest of the process is the same for both trust formation and evolution.

Based on the description of the various processes depicted in figures 2, 3 and 4, we identify the following components for our trusting collaboration architecture:

- **Request Analyser**. The request analyser is the front-end of the whole architecture. It receives all collaboration requests from other principals and relays to them all decisions taken. Furthermore, it is also responsible for separating the information included in the collaboration request into requestor related information and context related information. The former is relayed to the entity recognition module, while the latter is relayed to the risk evaluator.
- **Trust Calculator**. The main purpose of the trust calculator is to maintain the local trust state. Note that this state is described by a local trust policy π, which combines current local trust values and references to other principals to provide us with trust values for specific principals. Since, the local trust state is mutable the trust calculator should also provide us with mechanisms to update it. Moreover, it should be aware of the structure of the trust domain and should provide the operations over it necessary for evidence evaluation. For example, it should provide operations that can calculate the glb or lub of two trust values according to the trust ordering complete lattice or the glb of two trust value according to the certainty ordering cpo, etc. It should be clear that the trust calculator is in fact an extended version of the trust box described in section 2.1.
- **Risk Evaluator**. The main purpose of the risk evaluator is to maintain the results of the risk analysis, i.e. the risk profiles. This means that it is responsible for the trust exploitation process, which includes both the contextualisation of the trust values and the application of the $select()$ function. It is also responsible for adjusting the risk profiles in accordance with the risk parameters. Moreover, it should be aware of the structure of the risk domain and should provide the operations over it necessary for trust evaluation. For example, it should be able to calculate the distance between two risk profiles, or to produce an evidential trust value using the $select^{-1}()$ function, etc.
- **Trust Lifecycle Manager (TLM)**. The main responsibility of the TLM is trust formation and evolution. It is thus responsible for evaluating evidence, with the help of the trust calculator and the risk evaluator, to produce attractions. It is also responsible for producing new trust values using the $evolve()$ functions and taking into account the dispositional parameters. In addition to the above, the TLM also provides an interface to the trust calculator allowing it to request the formation of a trust value for new principals.
- **Access Control Manager**. The access control manager is responsible for applying the access control policy to the selected risk profiles to make a decision for each collaboration request.
- **Collaboration Monitor**. The collaboration monitor is responsible for the production of observations. Since, an observation consists of the outcome of the action and its incurred cost or benefit, in order to produce them we need to monitor the effects of all interactions between collaborating principals.

- **Evidence Gatherer**. The evidence gatherer is responsible for collecting recommendations from other principals. Note that the issue of evidence gathering is further discussed in section 4.3.
- **Evidence Store**. The role of the evidence store is to keep all trust related information for as long as it is necessary. This information includes both the local trust values for the various principals as well as collected evidence. Local trust values need to be stored in order to support the trust evolution process. Note that in in a global computing environment it would be infeasible to keep local trust values for all principals we ever interacted with. In this case it is important that the evidence store provides some ageing process for the local trust values. This process can then be used to occasionally clear the store from unnecessary information.
  Collected evidence needs to be stored for both trust and risk evaluation purposes. In the case of trust evaluation how long the evidence needs to be kept depends on the type of $evolve()$ function we use. If we are using a trust evolution function then evidence needs to be kept indefinitely, unless our function specifies an evidence window in which case all evidence outside it can be discarded (see equation 11). If we are using a trust update function then evidence needs to be kept until it is taken into consideration (see equation 12). Note that evidence is not necessarily immediately taken into account. There is a tradeoff between the cost of frequent processing evidence and how up-to-date our local trust values are. Even if evidence might no longer be necessary to keep for trust evaluation purposes, it might be kept for risk evaluation purposes. In general, we would expect the risk evaluation process to run less frequently than the trust evaluation one. Section 4.2 discusses some additional issues regarding the evidence store.
- **Entity Recognition Module**. The absence of a global principal identification scheme in the global computing environment deems traditional authentication techniques problematic. To deal with these problems entity recognition schemes have been proposed [43]. In general, entity recognition is a superset of authentication. In our architecture the entity recognition module acts as a placeholder for the module providing the functionality to support an entity recognition scheme. This module would provide us with a local id for all entities that we have interacted with based on the requestor related information of the collaboration request.

Figure 5 shows how the various components described above are put together to define our trusting collaboration architecture. The different types of arrows represent the different processes as they are carried out by the architecture. Note that in the case of decision making, the whole process is triggered by the arrival of a collaboration request to the request analyser. In the case where the requesting principal is not known to us, the trust calculator requests the formation of a new trust value. This is demonstrated in the diagram by the decision making arrow going from the trust calculator to the TLM. If the TLM decides to gather recommendations as part of this formation process it has to contact the evidence gatherer. The process of gathering recommendations and evaluating them as part

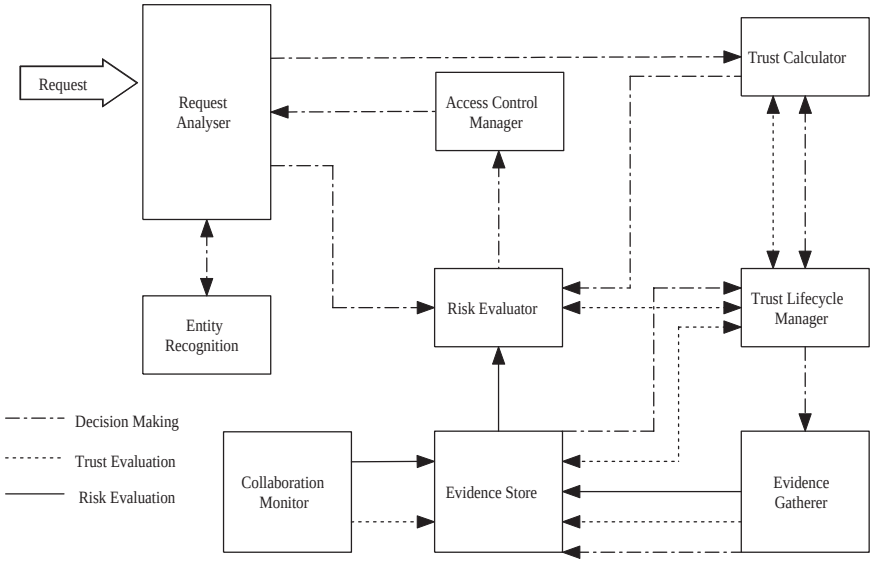


Fig. 5. The overall trusting collaboration architecture.

of this formation process is demonstrated by the decision making arrows going from the TLM to the evidence gatherer, to the evidence store and back to the TLM. Note also in fact, that the formation and evolution processes are in fact triggered by the collaboration monitor and/or the evidence gatherer. Finally, the risk evaluation process also depends on the availability of evidence.

4.2 The Trust Information Structure

In section 4.1 we introduced an evidence store for trust related information, but did not specify its internal structure. In the discussion of the store we identified two types of trust related information, namely trust values and evidence. We can take the view that trust values are in fact summarisations of previously evaluated evidence. According to this view we could see the evidence store as being organised into two layers, one for each type of trust related information. Although this sort of trust information structure adequately supports the decision making and trust evaluation processes, it has certain shortcomings with respect to recommendations. The problem lies in the fact that summarising previously evaluated evidence into a single trust value results in combining both our personal experiences of interacting with a particular principal and also the recommendations that we have received about that principal from others. As reported in the literature, this may lead to double counting of evidence [1, 3, 25, 28, 52]. This results when the personal experience of a particular principal is relayed as recommendations from two different paths. Moreover, the recommendation adjustment process becomes further complicated in the presence of third

party recommendations. In order to circumvent these problems, we summarise our personal experiences separately from the recommendations. This introduces an intermediate layer to the structure of the evidence store. This layer includes a trust value based solely on our personal experiences, T_{obs}, and a trust value based solely on our collected recommendations, T_{rec}. Note that top layer of the structure still contains our overall trust value, T_{ov}. This structure is depicted in figure 6.

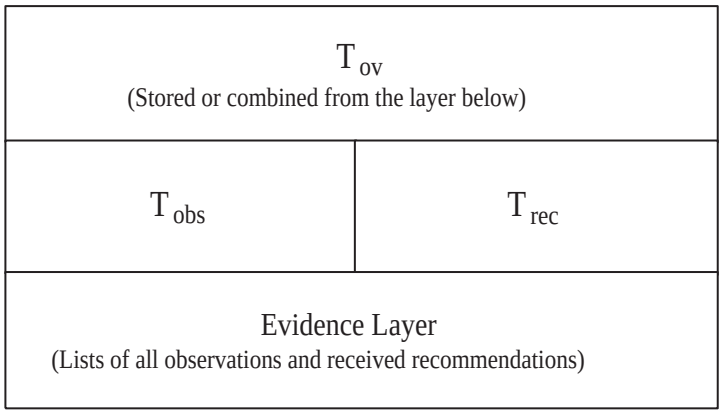


Fig. 6. The Trust Information Structure.

In summary, the trust information structure is organised as follows:

- The base layer contains lists of evidence, i.e. observations and recommendations. This layer is updated by the evidence gatherer and the collaboration monitor whenever a new recommendation is collected or the result of a particular collaboration becomes known respectively. Note that there might be a fixed size for these lists. If the lists have fixed size, then when attempting to add new evidence to a full list, an evidence evaluation and trust value update process will be triggered. This process will result in the evidence being summarized into new trust values for T_{obs}, T_{rec} and T_{ov}. Subsequently, the lists may be cleared of the evaluated evidence. Note that observations and recommendations are evaluated separately with respect to T_{obs} and T_{rec} for the update of these two values, and they are also evaluated with respect to T_{ov} for the update of this value. In this way, we keep our opinions based on our personal experiences separate from our overall opinion. So, whenever we are required to provide a recommendation, we can pass our observational trust value T_{obs} as our opinion. In this way, we avoid all the above mentioned problems.
- The second layer contains just the summaries of our personal experiences and our collected recommendations in the form of T_{rec} and T_{obs}. These sum-

maries are produced in the way we described above. Note that instead of directly evaluating the evidence and updating the overall trust value, we could instead use our observational trust value as our current opinion, then consider the recommendational trust value as an evidential trust value, and use it to produce our overall trust value. This process would be similar to the indirect observation evaluation process described in section 3.3.
- The third layer contains just our overall trust value. This value is the local trust value used by the trust calculator.

4.3 Evidence Gathering

Evidence gathering refers to the processes of collecting and distributing recommendations and thus refers to the functionality provided by the evidence gatherer component of our architecture (see figure 5). There are two ways in which recommendations may be collected, either by explicitly requesting them (pull model) or when others are offering them to us (push model). In either case, since their value depends on our trust in the recommender, we usually do not request or accept them indiscriminately. Finding which recommenders to accept recommendations from is not straightforward. There are a number of approaches we could follow:

- We could bootstrap principals with a list of similar entities that could be likely sources of recommendations. However, it might be the case that none of these entities can be contacted at a particular point in time.
- We could ask unknown principals to provide us with a list of entities that could give us recommendations about them. However, this might be misleading, since no principal would recommend anyone they had a bad interaction with.
- We could ask our neighbours to suggest good recommenders. Note that the neighbourhood might have different meanings. For example, it may be comprised of our acquaintances or directly accessible entities in an ad-hoc network, etc. However, there are no guarantees that our neighbourhood is a good source of recommendations.
- We could also introduce brokers that can suggest recommenders for various types of interactions. In this case, these brokers play the role of trusted third parties.
- In some situations, however, when trustworthy recommenders are not known, a broadcasted request for recommendations might be our only option. This approach offers no guarantees whatsoever and it would probably be our last resort.

Furthermore, in any case, accepting offered recommendations indiscriminately makes us susceptible to recommendation spamming and denial of service attacks.

When we are requested to provide a recommendation, we must bear in mind that by doing so, we reveal personal information. This means that there might

be privacy concerns regarding the provision of recommendations. We can take the view that our experiences are a valuable resource that we need to protect. In this case, we could use the SECURE model to control access to our experiences. This means that requests for recommendations are just one type of action that our application deals with and as a result, recommendations requests trigger the decision making process depicted in figure 2. Alternatively, we could consider our experiences as a valuable resource, but control access to it through a separate process. It could be argued that this alternative is preferable since it may involve a more simple process compared to the usual decision making one.

In general, research has shown that exchange of experiences can be beneficial for a community of entities [21]. However, based on the above discussion it might seem that entities should be inclined to avoid collecting and distributing recommendations. For this reason, it is important that in any system where we want exchange of experiences, that such an exchange is actively encouraged. In other words, the system has to punish principals that do not share their experience and reward those that do. Such behaviour may be enforced if the trust values for the principals include an element referring to their attitude towards experience sharing. In conclusion, the whole issue of evidence gathering is an interesting one, and deserves further investigation.

5 A Formal Model for Trust Lifecycle Management

This section[9] describes a preliminary formalisation of our trust lifecycle management architecture described in section 4.1. The formalisation takes into account the structure of the evidence store described in section 4.2, and is based on the a trust policy language initially introduced in [10]. Note that when we are talking about trust policies in this section, we in fact refer to the local trust policies described in section 2.1.

5.1 The Trust Policy Language

In the trust policy language, policies are formally defined as functions of type $\mathcal{P} \rightarrow \mathcal{T}$ such that for each principal they return a trust value. The language permits the specification of these functions. A policy language has the form $\lambda x : P.\tau$ where τ specifies the returned value of the policy, done by using certain operators. One of these operators is the operator $\ulcorner . \urcorner$, called *reference*. For instance the policy $\lambda x : \mathcal{P}.\ulcorner a \urcorner(x)$ specifies the function that, for any principal $b \in \mathcal{P}$, returns a's trust in b. References make policies dependent on the global trust m. The function m is defined as the least fixed point of all policies (see [10] for more details and formal semantics).

[9] This section includes significant contributions by Marco Carbone of BRICS, University of Aarhus.

The operator of reference can also be combined with other operators provided from the language. For instance we could write a policy which says that our trust in b is a's trust in b but for any other principal c it will be the least upper bound between a's trust in c and a threshold value t:

$$\lambda x : \mathcal{P}.(x = b) \mapsto \ulcorner a \urcorner(x); (t \sqcap \ulcorner a \urcorner(x)) \tag{13}$$

where the operator $\cdot \mapsto \cdot; \cdot$ is semantically equivalent to an if-then-else and $\sqcap$ is the least upper bound (lub) in lattices.

5.2 Encoding in the Policy Language

The encoding in the policy language is done by exploiting two features: the syntax of the policy and the structure of the set $\mathcal{T}$. When we give a policy π written in the language we give an algorithm for computing a function. Hence, when writing the policy, we "store" some information in the syntax of the language which means that it is possible to record the current trust value.

Representing the various layers of the evidence store (see figure 6) could be handled by giving more structure to the set $\mathcal{T}$. Suppose that the initial set of trust values is T_{val}, provided with a certainty ordering $\sqsubseteq$ and a trust ordering $\preceq$ as required (see section 2.1. Then we can define a new set $\mathcal{T}$ as:

$$\mathcal{T} = [EVD] \times T_{obs} \times T_{rec} \times T_{ov} \tag{14}$$

where $[EVD]$ is a list of evidence, e.g. $[evd_1, \ldots, evd_n] \in [EVD]$ for evd_1, $\ldots, evd_n$ pieces of evidence. This means that our policy contains a list of evidence (as shown in the base layer of the evidence store in figure 6), and three trust values (as shown in the second layer in figure 6): the first refers to the current trust value based on observations, the second refers to the current trust value based on recommendations and the third refers to the overall current trust value (the one used for decision making). To describe these trust values as a policy, we will initially start from the situation where there are no observations or recommendations. Therefore, the policy could be defined as:

$$\pi_a = \lambda x : \mathcal{P}.([], \bot, \bot, \bot) \tag{15}$$

where [] is the empty list and $\bot$ is the bottom of the certainty ordering (see section 2.1.

When we have an observation, say o_1, we just need to update the list of evidence and so get a new policy as follows:

$$\pi_a = \lambda x : \mathcal{P}.([o_1], \bot, \bot, \bot) \tag{16}$$

A recommendation from b about c can be represented as $r_1 = \ulcorner b \urcorner(c).2$, where the operator .2 (in general .n for any natural number) stands for the projection

of the 2-nd element (in general n-th) over a tuple (in this case over the set $\mathcal{T}$). When we have a recommendation, say r_1, we also just need to update the list of evidence and so get a new policy as follows:

$$\pi_a = \lambda x : \mathcal{P}.([o_1, r_1], \bot, \bot, \bot) \tag{17}$$

When we decide to consider the evidence and update our current trust value, we get a new policy. Note that this process is triggered with the call of the *update*() method on the trust box (see section 2.1). Supposing that our policy was $\pi_a = \lambda x : \mathcal{P}.([o_1, r_1], t_{obs}, t_{rec}, t_{ov})$, and that both o_1 and r_1 is evidence referring to principal c, then the new policy is as follows:

$$\begin{aligned}\pi_a = \lambda x : \mathcal{P}.([], (x = c) \mapsto evolve(a_{o_1}, \ulcorner a \urcorner(x).2); t_{obs},\\ (x = c) \mapsto evolve(a_{r_1}, \ulcorner a \urcorner(x).3); t_{rec},\\ (x = c) \mapsto evolve(a_{r_1}{}', evolve(a_{o_1}{}', \ulcorner a \urcorner(x).3)); t_{ov})\end{aligned} \tag{18}$$

where

$$a_{o_1} = evaluate(o_1, \ulcorner a \urcorner(c).2), \tag{19}$$

$$a_{r_1} = evaluate(r_1, \ulcorner a \urcorner(c).3), \tag{20}$$

$$a_{o_1}{}' = evaluate(o_1, \ulcorner a \urcorner(c).4) \text{ and} \tag{21}$$

$$a_{r_1}{}' = evaluate(r_1, \ulcorner a \urcorner(c).4) \tag{22}$$

are the respective attractions of the observation and the recommendation [10]. For definitions of *evaluate*() and *evolve*() see equations 6 and 12 respectively. Note that in this case, all evidence taken into account is removed from the list of evidence. Furthermore, we only update the trust values for the subject of the evidence, while the values for the other principals remain the same. Finally, the evaluation and the evolution of the evidence is with respect to the corresponding element of the $\mathcal{T}$ tuple, thus the need to calculate different attractions $a_{o_1}, a_{o_1}{}'$, and $a_{r_1}, a_{r_1}{}'$ respectively.

Here, we should point out that both recommendation and references refer to principals' trust in other principals. However, as the above formalisation demonstrates recommendation are always evaluated before the trust value evolves, i.e. their effects on our trust policy are expressed in terms of their attraction.

5.3 Concluding Remarks

In conclusion, this section presented a preliminary formalisation of trust lifecycle management model based on the trust policy language. This formalisation provides us with a basis for further formal analysis of our model and a first step towards the verification of evidence evaluation and trust evolution algorithms. We consider both of these issues quite important for further investigation in the future.

[10] Recommendations would usually also go through a recommendation adjustment process.

6 Case Studies

In this section we present two case studies on our collaboration model: (a) a smart space scenario and (b) an e-purse scenario. Both case studies demonstrate how to engineer trust and risk domains that have the required properties set out in section 3.1. They also present specific examples of the various functions defined in sections 3.1, 3.2 and 3.3. Furthermore, they show how the dispositional characteristics introduced in section 3.3 can be modelled in the evidence evaluation and evolve trust processes.

We should note that in both case studies the emphasis is in demonstrating the concepts of our collaboration model. As a result, aspects not relating to our model have been significantly simplified or even ignored. The production of complete applications addressing in full the issues of both scenarios was deemed to be outside the scope of this paper.

6.1 The Smart Space Scenario

In smart environments, sensors are placed in rooms and offices to enable the collection of data such as the location of the smart space inhabitants. The vast amounts of personal information collected by such systems has led to growing concerns about the privacy of their users. Users concerned about their private information are likely to refuse participation in such systems.

Privacy control, as the term states, encompasses both the notion of privacy and the notion of control. A good privacy solution should combine both of these two notions. According to Alan Westin "privacy is the claim of individuals, groups, or institutions to determine for themselves when, how and to what extent information is communicated to others" [49]. Privacy on its own is about protecting users' personal information. On the other hand, control is about justification of privacy and plays a role in the management of privacy.

We consider the case of a smart environment equipped with a context information server (CIS) that collects, stores and interprets user contextual information, e.g. location information. Users can request to receive the information that the CIS collects for other users. The way information is exchanged between users classifies them as *information owners*, those whose contextual information is managed by the CIS, or *information receivers*, those who would like to use the managed contextual information. We believe that in any context information system, information owners may be willing to disclose their contextual information if this disclosure is potentially beneficial. Accordingly, for any context information system to be acceptable to the information owners, it must provide mechanisms for controlling access to personal contextual information. These mechanisms should provide fine-grained control of the disclosure of personal contextual information in order to allow maximisation of its expected benefit and minimisation of costs.

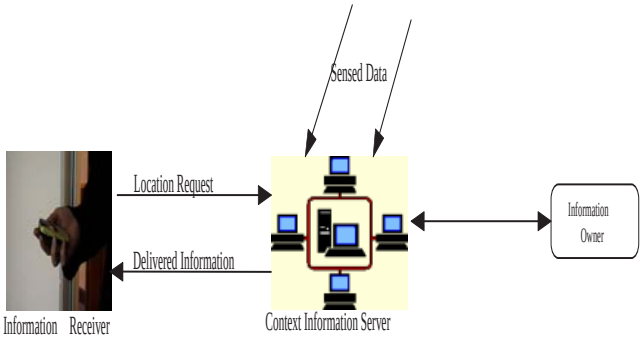


Fig. 7. A user requests location information from the context information server.

In this scenario we consider the case of a CIS that is able to track users, the information owners in this case, as they move within the smart space. Users, the information receivers in this case, can request from the CIS location information of others with the intension to meet them. The CIS provides location information to information receivers in the form of notifications. This allows information receivers to track the movements of information owners for a particular period of time or until a meeting between the information receiver and the information owner takes place.

More specifically, as depicted in Figure 7, first, the information receiver sends a request to the CIS for location information regarding a particular information owner. We assume that all users are registered with the system, and as a result, the identification by the CIS of both the information receiver and the information owner is trivial. On receipt of the request the CIS needs to decide whether to permit the tracking of the information owner. The decision is based on the expected costs and benefits of providing the information as they are perceived by the information owner.

In general, information owners are willing to trade the cost of privacy loss resulting from allowing information receivers to track them with the gains of a beneficial meeting. The cost or benefit of providing their location information depends on two parameters: the cost of tracking, which is analogous to the duration of tracking, and the cost/benefit of the meeting, which depends on the outcome and the duration of the meeting. The meeting outcome could either be beneficial or not. The likelihood of the various costs and benefits depends on the trustworthiness of the information receiver. The more trusted the information receiver is, then an interaction with him or her is more likely to be beneficial. Moreover, information owners specify a privacy policy that determines which interactions should be permitted by the CIS, or in other words, under which circumstances the CIS should disclose their location information. The privacy policies set boundaries on the acceptable expected costs/benefits of interactions.

These boundaries are expressed as limits on both the tracking and meeting duration. Thus, in reply to the information receiver's request and in accordance to the information owner's privacy policy the CIS needs to decide how many of tracking and meeting time units it should provide to the information receiver.

As we mentioned above, when the CIS decides to provide the requested information it starts to send location notifications to the information receiver. These notifications can stop in two different ways. Either the allocated tracking time expires or the sensors detect the information receiver and information owner in proximity of 1 meter to each other. The latter indicates that the purpose of the request has been fulfilled, i.e. a meeting between the information receiver and the information owner is taking place. Moreover, the CIS sends messages to the information owners when the allocated meeting time is due to expire. When a meeting is finished the information owner sends feedback to CIS regarding its outcome. This feedback is used in the evaluation of the trustworthiness of the information receiver.

Privacy policies. In order to increase the flexibility of the privacy policies and reduce their complexity we allow information owners to divide information receivers into groups. For each group the information owner can set a different privacy policy. These policies are configured by setting the following parameters:

- *Maximum tracking duration units, (mtdu).* This parameter determines the maximum time that an information receiver is allowed to track the information owner. The time is measured in units of 5 minutes.
- *Maximum meeting duration units, (mmdu).* This parameter determines the maximum time that a meeting with an information receiver is supposed to last. This parameter helps information owners to better manage their time by giving more time to meetings that are likely to be beneficial and ones that are likely to be a waste of time. This time is also measured in the units of 5 minutes.
- *Value of time, (vt).* This parameter indicates how valuable the information owner considers his or her time. The values for this parameter are: *not so precious*, *precious*, and *extremely precious*.
- *Privacy sensitivity level, (psl).* This parameter determines the degree to which information owners are concerned about their privacy. The values for this parameter are: *little concerned*, *quite concerned* and *very concerned*.

For example let us consider a smart university department, where information receivers can track the location of academic members of staff (information owners) to arrange meetings with them. Examples of different groups of information receivers could be:

- Students, which can be further divided into: general students (GS) and supervised students (SS), either postgraduate students or final year project students.

– Academic staff which can be further divided into: general member of staff (GA), and staff within the same research group (RA).
– Support staff which can be further divided into: system support(Sys), and secretaries (Sec).

Table 1. A Privacy Policy Configuration Example

User Group	mtdu	mmdu	vt	psl
GS	4	2	extremely precious	very concerned
SS	5	5	precious	quite concerned
GA	6	3	not so precious	little concerned
RA	6	4	not so precious	little concerned
Sys	4	2	precious	quite concerned
Sec	8	2	precious	quite concerned

Table 1 provides an example of the parameters that could be associated with the various groups of users in the smart university department.

Observations. Observations are obtained when interactions are finished. These observations reflect the observed cost/benefit of the outcome of the interaction. The actual cost/benefit is combination of both the cost of tracking duration and the cost/benefit of the meeting duration.

The multiplication of tracking duration units by the cost per unit ($tdu{*}unit\_cost$) gives the cost of tracking. The privacy sensitivity level determines the cost of tracking as follows:

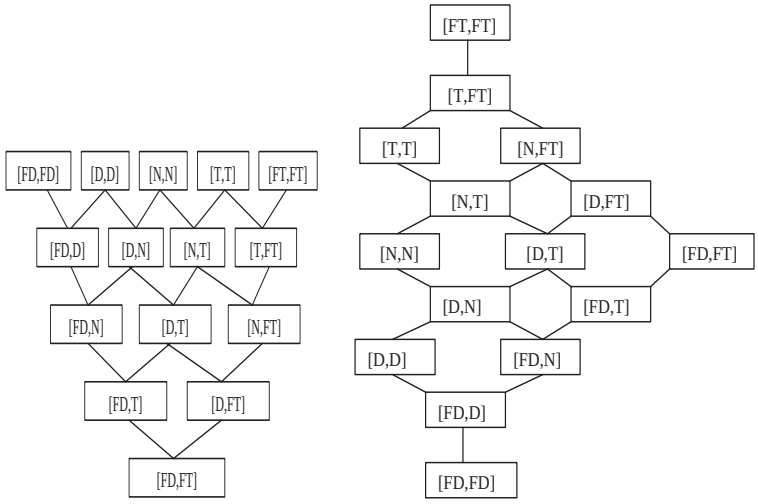
– The $unit\_cost$ could be 2, if the psl is very concerned.
– The $unit\_cost$ could be 1, if the psl is quite concerned.
– The $unit\_cost$ could be 0, in the psl is little concerned.

For example referring back to Table 1, if a a general students tracks a member of staff for duration of 3 units, the cost is 6 ($tdu * unit\_cost = 3 * 2$).

The multiplication of meeting duration units by the meeting cost per unit ($mdu{*}$ $munit\_cost$) gives the cost/benefit of the meeting. The meeting cost per unit depends on the outcome (mo), which could either be: *beneficial*, *average* or *waste of time*. So, the possible meeting costs per unit could be:

– -2, if the meeting was beneficial.
– 2, if the meeting was wasting time.
– If the outcome of the meeting is average, then the cost per unit depends on the value of the time a follows:
  • It could be -1, if vt is not so precious
  • It could be 0, if vt is precious,
  • It could be 1, if vt is extremely precious

Trust values. The decision about granting access to location information depends on both the category of the user and the trust value associated to him or her. For construction of our trust values for this scenario we define a set of basic trust values: FD, D, N, T, and FT representing fully distrusted, distrusted, neutral, trusted, and fully trusted respectively. Following the interval construction method described in [9], we construct our application trust values by defining a certainty complete partial order with a least element and a complete lattice, which define our certainty trust ordering respectively(see Figure 8).



(a) Certainty Ordering (b) Trust Ordering

Fig. 8. The Smart Space Scenario Trust Domain.

Risk analysis. The risk of an outcome is a function of the likelihood and impact in terms of cost and benefit of that outcome. The risks of the interactions in the smart department are centered on both users' privacy and the way their time is exploited in meetings. Users in general are concerned both about others knowing their location information and about how to make the best use of their time.

In order to determine the range of possible costs/benefits for each outcome, we calculate first the maximum costs/benefits. The calculation is based on the functions for calculating the tracking and meeting costs/benefits provided above in section 6.1. The total maximum cost combines the maximum cost of both tracking and a meeting by using the maximum tracking duration units (mtdu) and the max meeting duration units (mmdu). In calculating the total maximum benefit, we will only consider the maximum benefit of a meeting since there is no direct benefit from tracking.

In order to provide an insight into the risk analysis and decision making, we take as an example the supervised student (SS) group (see Table 1 above for the policy parameters for the group). The maximum cost of tracking by a user from this group is $(mtdu * unit\_cost) = (5 * 1) = 5$, where the unit cost unit is 1, since the privacy sensitivity level for a user from this group is 'concerned'. The maximum cost of meeting is $(mmdu * unit\_cost) = (5 * 2) = 10$, where the unit cost is 2, considering the worst outcome of the meeting, which is wasting time. As a result, the total maximum cost is: cost of tracking + cost of meeting $= 5 + 10 = 15$. The maximum benefit of a meeting is: $(mmdu * unit\_cost) = 5 * -2 = -10$, where the unit cost is -2 considering the meeting outcome to be beneficial. So, the range of potential cost/benefits for supervised students are from -10 to 15.

The users' trustworthiness determines the likelihood of the various costs/ benefits. In this example we assume the following cost probability density functions (cost-pdfs) for the basic trust values. To simplify the construction of the cost-pdfs, the range of cost/benefit is divided into 5 intervals and the value on top of each column represents the likelihood of this interval of cost/benefit. The cost-pdfs corresponding to the basic trust values for supervised students are illustrated in figure 9.

For the cost-pdfs of the trust value intervals containing multiple basic values we average out the corresponding cost-pdfs. So, the cost-pdf for the interval [FD,N] is the average of the cost-pdfs for [FD,FD], [D,D] and [N,N] and is depicted in figure 10.

Decision making. From the interaction request, the CIS knows the identity of the requestor and the group he or she belongs to and can select the corresponding privacy policy. The next step in the decision making process is to apply the *select*() function parameterised by the current trust value for the requestor (see equation 3). This function will provide the appropriate cost-pdf. Note that in this scenario, we only consider a single context for the trust values. As a result trust exploitation does not require the trust value contextualisation. In the final step the CIS uses the selected cost-pdf and privacy policy to reach a decision.

Policies are described as functions that given the risks involved in the interaction determine how many units of tracking and meeting duration should be provided to the requestor. The policies in fact describe the risk that the information owner is willing to accept. For example, if the cost-pdf predicts high benefits from an interaction, then the privacy policy will assign more units for both tracking duration and meeting duration.

Evidence evaluation. In this scenario we follow the indirect approach to evidence evaluation. So, we first determine the evidential trust value T_{evd}, which is subsequently evaluated with respect to the current trust value, using the *evalute*() function (see equation 6). Finally, we determine which trust value would have been a more accurate predictor of the observed outcome.

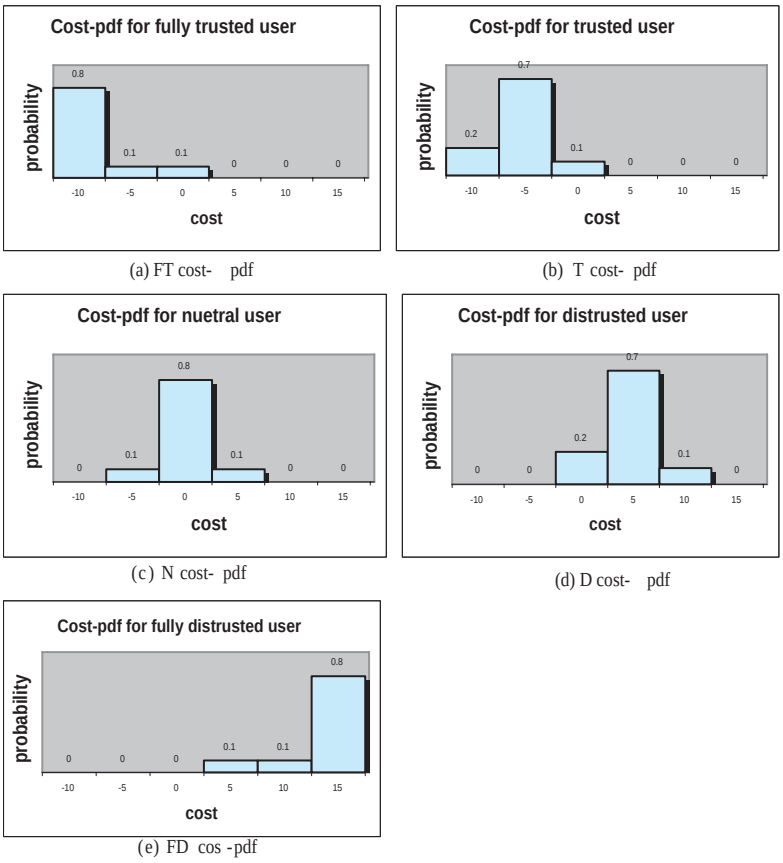


Fig. 9. Risk Analysis for Supervised Students.

For example, suppose that Alice (Bob's PhD student) has interacted with Bob ten times. After collecting the ten observations, we can create an observed cost-pdf. This cost-pdf can then be compared to the set of all cost-pdfs determined by our initial risk analysis. The aim of the comparison is to determine the cost-pdf that reflects most closely the observed outcomes. The trust value related to this cost-pdf is considered to be the evidential trust value.

More specifically, let us suppose that figure 11 depicts the observed cost/benefit of ten interactions. A comparison of this cost-pdf to the set of cost-pdfs provided by the risk analysis shows that the closest one is the cost-pdf for the [T,T] trust value, which is considered to be the value of T_{evd}. Note that in order to determine the closest cost-pdf we use the cost-pdf distance as defined by equation 7. For example, assuming the cost-pdfs for [FT,FT] and [N,FT] are as depicted in figures 9 and 10 respectively, then their distance is: $\mid 0-0 \mid + \mid 0-0 \mid + \mid 0-0.05 \mid + \mid 0.1-0.45 \mid + \mid 0.1-0.4 \mid + \mid 0.8-0.1 \mid = 1.4$.

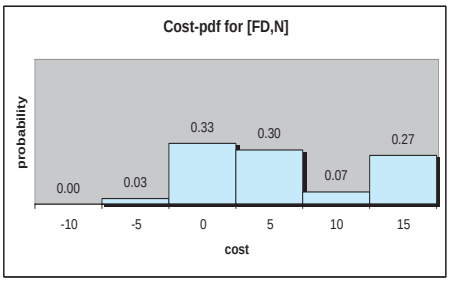


Fig. 10. [FD,N] cost-pdf

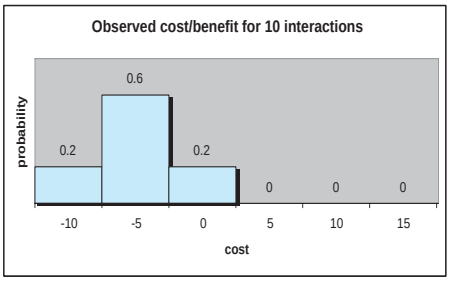


Fig. 11. The cost-pdf of the observed outcomes.

Trust formation and evolution. The evolution of the trust values is based on the calculation of the attraction of T_{evd}. We should remind the reader that the direction of the attraction defines two sets $T_\sigma, T_\tau \subseteq \mathcal{T}$ of acceptable trust values in terms of certainty and trust respectively.

For example, suppose that Alice has just finished ten interactions with Bob and Bob's trust value for Alice is $T_{curr} = [N, N]$, while the ten interactions were evaluated as $T_{evd} = [FT, FT]$. For T_τ, we will check the relative position of T_{curr} and T_{evd} on the trust lattice (see figure 8). While, T_σ is determined by checking the relative position of the two values on the certainty complete partial ordering (see also figure 8). According to section 3.3, the attraction of T_{evd} is increasing, consequently the new trust value T_{new} should be between T_{curr} and T_{evd}, so $T_\tau = \{[N, N], [N, T], [T, T], [N, FT], [T, FT], [FT, FT]\}$. Similarly, the attraction of T_{evd} is contradicting, since the g.l.b. of the two values is [N,FT]. Therefore, $T_\sigma = \{[N, N], [N, T], [N, FT]\}$. So, the new trust value T_{new} should be a member of the intersection of these two set, $T_{new} \in (T_\tau \cap T_\sigma) = [N, N], [N, T], [N, FT]$. The value of the attraction depends on the distance of the cost-pdfs of T_{evd} and T_{curr}.

The *evolve*() function depends on the distance of the cost-pdfs of the potential trust values for T_{new} from the cost-pdf of T_{evd}. The maximum distance for two cost-pdfs is 2 if they are fully contradicting and the minimum distance is 0 if

they are identical. To allow the expression of dispositional characteristics of the principal we introduce in the calculation of T_{new} a dynamic threshold dt, which determines how quickly or slowly the system adapts to evidence. The range of potential cost-pdf distances determines the range of potential dt values from 0 to 2.

Referring back to the example above, the distances between the cost-pdf for $T_{evd} = [FT, FT]$ and the cost-pdfs of [N, N], [N, T] and [N, FT] are 1.6, 1.4 and 1.74 respectively. From these trust values we eliminate those that the distance is greater than or equal to the distance of $T_{curr} = [N, N]$. [N, FT] is such a trust value. After the elimination, we start comparing the distance of the remaining trust values to dt in a descending order. The first trust value with a distance less than dt or the trust value with the lowest distance becomes our new trust value. So, assuming $dt = 1$ the new trust value will be [N, T].

Concluding remarks. In conclusion, in the smart space scenario we consider only a single action with a single outcome that could incur a range of cost and benefits. Moreover, we only consider observations as the source of evidence. This simplifies the application of our collaboration model in a number of ways:

1. It simplifies the risk analysis by allowing the use of cost-pdfs for the representation of the risk profiles. The use of cost-pdfs for the representation of the risk profiles enables us to use the constructive approach to trust model development (see [9]) in our risk analysis.
2. It simplifies the trust exploitation process by removing the need for contextualisation of the trust values.
3. It simplifies the trust evolution process by allowing the construction of observed behaviour profiles as cost-pds in a straightforward manner. These profiles can be directly compared to the cost-pdfs of the risk analysis, i.e. expected behaviour profiles, simplifying the observation evaluation process.

Despite the above simplifications the scenario still demonstrates:

- how the close relationship between trust and risk can be achieved,
- how indirect observation evaluation can take place, and
- how dispositional parameters can be used to influence the trust evolution process.

6.2 The e-purse Scenario

The scenario involves the use of an e-purse when a passenger is interacting with a bus company. The purpose of the e-purse is to hold a relatively small amount of e-cash (in this scenario the e-purse is limited to 100 euro) that the owner can use as if it were real cash for buying bus tickets (see figure 12).

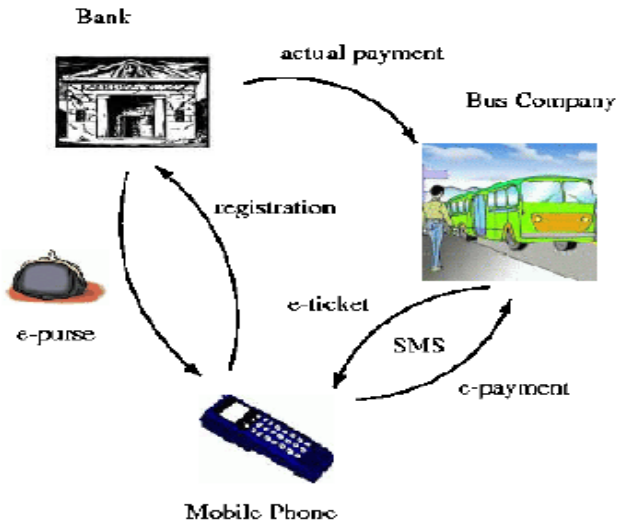


Fig. 12. E-purse scenario interaction.

Users can refill their e-purse by contacting their bank provided that there is enough cash in their account. There are three different principals involved in this scenario: the passenger (owner) of the e-purse, the bus company and the bank. In this scenario we are only interested in modelling the trust relationship between the bus company and the passenger. We consider the example interaction where passengers want to purchase tickets using their e-purse.

E-cash is based on a protocol that although it protects user anonymity during normal transactions, enables identification of guilty parties in fraudulent transactions. Every time the bus company accepts e-cash in a transaction it takes the risk of losing money due to fraud. Therefore, for the bus company to decide how to respond to a purchasing request, it needs to determine the trustworthiness of the passenger. Principals can assign different levels of trust to different entities based on the available information so as to evaluate the level of risk transactions involving the user entail.

Trust values. In the e-purse scenario the range of basic trust values is [0, 100] reflecting the amount of e-cash that the bus company is willing to accept from the requesting user. Following the interval construction technique from [9], we construct our trust values as intervals $[d_1, d_2]$ of the range of basic trust values. An interval $[d_1, d_2]$ indicates that the bus company is quite certain about the validity of amounts up to d_1 of e-cash , fairly uncertain about the validity of amounts between d_1 and d_2 and fairly certain that any amount above d_2 will be invalid. So, for any ticket purchasing request of more than d_2 the user has to pay in cash. It should be clear that the use of such trust values really simplifies the decision making process.

Risk analysis and decision making. The essential step in trust exploitation is to determine expected behaviour on the basis of trust intervals. This is achieved by using the trust interval to determine the risk of interacting with a particular principal. The assumption is that the passenger's trustworthiness reflects the expected loss or gain during a transaction involving him or her. The costs involved in an interaction range from -100 to 100, denoting the maximum gain or loss for the bus company.

In the general case, the calculated risk allows entities to decide whether or not to proceed with an interaction. In this scenario, a simplified view is taken, whereby the trust value directly determines the amount of e-cash a bus company is willing to accept. The decision making process for a ticket of value x regarding a passenger with trust value $[d_1, d_2]$ is as follows:

- If $x < d_1$ then the whole amount of the transaction can be paid in e-cash.
- If $x > d_2$ then the option of paying in e-cash is not available and the full amount has to be paid in cash.
- If $d_1 < x < d_2$ then the likelihoods of the possible outcomes are examined. Note that there are only two possible outcomes, the e-cash provided by the passenger will be either valid or invalid. For the calculation of the likelihoods, we divide the range from d_1 to d_2 into a number of units, n. For example n could be equal to the price of the cheapest ticket, say 5 euro. In this case, the number of units is determined by dividing the whole range over five $(d_1 - d_2)/5$. The likelihood of invalid e-cash for each unit is (m/n), where $m = 0, 1, .., n$ (see figure 13). Note that the likelihood of invalid e-cash increases from d_1 (with a probability of 0 for invalid e-cash) to d_2 (with a probability of 1 for invalid e-cash). Considering these likelihoods for the possible outcomes the bus company can place a threshold of acceptable risk. So, it will only accept e-cash for transaction with risk below the threshold.

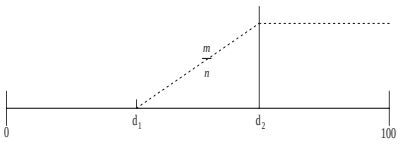


Fig. 13. Risk Analysis.

Trust evolution. In this scenario we again only consider observations and we combine the two processes of trust evolution, namely evidence evaluation and trust evolve. As a result, the attraction of every observed outcome, whether the provided e-cash were valid or not, raises or drops the boundaries of the current trust value T_{curr}.

In accordance with section 3.3, if the e-cash was valid then the attraction of the observation is considered positive, in which case we expect the lower and upper bound of T_{curr} to either remain unchanged or to be raised. While, if the e-cash was invalid the attraction of the observation is considered negative, in which case we expect the lower and upper bound of T_{curr} to either remain unchanged or to be dropped. Moreover, if the outcome was expected, i.e. its likelihood was more than 50%, then the attraction of the observation is considered reinforcing, otherwise is considered contradicting. Let us assume that m_1 and m_2 are the movements of the lower and upper bound of T_{curr} respectively. Then, in the case of reinforcing attraction we expect the size of the interval $(d_2 - d_1)$ to remain unchanged or be reduced, i.e. $m_1 > m_2$. While, in the case of contradicting attraction we expect the size of the interval to remain unchanged or be increased, i.e. $m_1 < m_2$. This is summarised in Table 2.

Table 2. Summary of observation evaluation.

attraction direction	direction of boundary movement	interval size
positive, reinforcing	$\longrightarrow$	$m_1 > m_2$
positive, contradicting	$\longrightarrow$	$m_1 < m_2$
negative, reinforcing	$\longleftarrow$	$m_1 > m_2$
negative, contradicting	$\longleftarrow$	$m_1 < m_2$

For example, let us assume that t denotes the amount of e-cash in the observed transaction and $T_{curr} = [d_1, d_2)]$. Then, the movements m_1 and m_2 of the lower and upper bound of T_{curr} could be determined as follows:

1. If $t < d_1$, then
   - If the e-cash is valid, then the attraction is reinforcing and positive. In this case, the observation does not really contribute any additional information about the principal and is therefore ignored, i.e.:

     $$m_1 = 0 \text{ and } m_2 = 0 \tag{23}$$

     Note that if we do not ignore this kind of observations, but instead use them to raise the trustworthiness of the principal, then we are exposing ourselves to the typical trust exploitation scum, where a large number of very small value transactions could allow a transaction of a substantial value to take place even though there is no evidence to support this decision.
   - If the e-cash is invalid, then the attraction is contradicting and negative. In this case:

     $$m_1 = \alpha_n \times (t - d_1) \text{ and } m_2 = \beta \times m_1 \tag{24}$$

2. If $t > d_1$, then

– If the e-cash is valid and the likelihood of t being valid is less than 50%, $Pro(t, valid) < 50\%$, then the attraction is positive and contradicting. In this case:

$$m_1 = \alpha_p \times (t - d_1) \times (1 - Pro(t, valid)) \text{ and } m_2 = 0 \quad (25)$$

– If the e-cash is valid and the likelihood of t being valid is greater than 50%, $Pro(t, valid) > 50\%$, then the attraction is positive and reinforcing. In this case:

$$m_1 = \beta \times m_2 \text{ and } m_2 = \alpha_p \times ((100 - d_2)/(d_2 - t)) \times (1 - Pro(t, valid)) \quad (26)$$

– If the e-cash is invalid and $Pro(t, invalid) > 50\%$, then the attraction is negative and contradicting. In this case:

$$m_1 = 0 \text{ and } m_2 = \alpha_n \times (t - d_2) \times (1 - Pro(t, invalid)) \quad (27)$$

– If the e-cash is invalid and $Pro(t, invalid) < 50\%$, then the attraction is negative and reinforcing. In this case:

$$m_1 = \alpha_n \times (d_1/(d_1 - t)) \times (1 - Pro(t, invalid)) \text{ and } m_2 = \beta \times m_1 \quad (28)$$

Note that α_p, α_n and β range from [0, 1] and are dispositional parameters that determine how slow or fast are the positive, negative and certainty dynamics respectively. If $\alpha_p > 0.5$ then we are talking about fast positive dynamics, while if $\alpha_p < 0.5$ we are talking about slow positive dynamics. Similarly, depending on the value of α_n we are talking about fast negative or slow negative dynamics. Moreover, if $\alpha_p = \alpha_n$ then we are talking about balanced slow or fast dynamics (see section 3.3). At the same time, if β is small we reduce the size of the trust interval quickly, while if β is large we reduce it slowly.

A specific example. Suppose that a passenger with a trust value [20, 70] paid valid e-cash worth 40 euro to the bus company. Supposing that the range between 20 and 70 is divided into 5 units each with a size of 10, the likelihoods of the two outcomes, valid or invalid e-cash, are: 40% for invalid and 60% for valid. So, the attraction of this observation is positive and reinforcing. Applying the functions described above, we have $m_1 = 8$, $m_2 = 0$. So, the new trust value $T_{new} = [d_1 + m_1, d_2 + m_2] = [20 + 8, 70 + 0] = [28, 70]$.

Concluding remarks. In conclusion, in the e-purse scenario we only consider a single action with two outcomes, valid or invalid e-cash. We also only consider observations. As a result, we remove the need for contextualisation of the trust values. The trust and risk domains are designed in a way that makes decision making quite straightforward. In contrast to the smart space scenario, the evaluation of observations is direct. In fact, the evidence evaluation and evolve trust processes have been merged into into a single process that directly updates the current trust value (see equations 23, 24, 25, 26, 27 and 28).

7 Comparison to the State of the Art

In the previous sections, we have discussed our collaboration model in terms of making decisions and evaluating these decisions. In this section we compare our work to the state of the art in trust management. The comparison is in terms of the following factors:

- The complexity of the trust domain and the decision making process. We are particularly interested in the modelling of uncertainty and risk, as well as the relation between trust and risk. We are also interested in the way in which situational aspects of trust are modelled, i.e. context modelling.
- The provision of trust lifecycle management, which incorporates trust formation, evolution and general dynamics of trust. Of particular interest are the types of evidence used, how this evidence is gathered, evaluated and whether the differences between direct and indirect evidence are taken into account.
- The support for the expression of differences in principal attitudes towards trust, i.e. dispositional character of principals.

We begin this section with a brief look at the origins of trust management in credential based systems in section 7.1, before concentrating on evidence based systems in section 7.2. Evidence based systems are of interest since their dynamic nature is more appropriate for the global computing environment.

7.1 Credential Based Trust Management

Matt Blaze et al. [6] were the first to define trust management, as "a unified approach to specifying and interpreting security policies, credentials and relationships that allow direct authorisation of security-critical actions". In such trust management systems, trust is viewed implicitly through the delegation of privileges to trusted entities via the use of credentials or certificates, which can be chained to represent the propagation of trust between entities. Requestors of a service can prove directly that they hold the correct credentials to authorise the requested service. Examples of this type of trust management system are [6, 5, 14, 27, 42, 30]. From its origins in [6], this model has been extended in various ways. The formation of trust relationships have been enhanced through dynamic credential discovery [14, 31] and trust negotiation for the bilateral exchange of credentials [42]. Further reasoning about evolution of relationships has been enabled through logic based approaches [29] and the ability to prohibit, constrain or even revoke delegations [27].

Despite such advances, these approaches are unsuitable for the global computing environment, primarily due to the implicit notion of trust. As a result, there is no consideration of risk or uncertainty, which are important characteristics of such an environment given the lack of complete knowledge of the system.

Moreover, the basic model only describes a way of exploiting established trust relationships for distributed security policy management, without determining how these relationships are formed. The fact that policy is evaluated with respect to delegated credentials means that if the necessary credentials cannot be discovered, the trust relationship does not exist and access cannot be granted. This is a real problem for the formation and evolution of trust relationships in an environment where network connectivity is not always guaranteed. These types of problems are typical of credential based trust management systems. The only way they can be resolved is with the introduction of an explicit notion of trust, accompanied by the notion of evidence. Evidence enables formation and evolution of trust relationships based on the outcomes of past interactions. Given these limitations of the credential based approach to trust management, henceforth we concentrate on evidence based approaches.

7.2 Evidence Based Trust Management

In recent years there have been attempts at more intuitive computational models of trust, with a basis in the history of past interactions. Examples of such work include [1, 13, 16, 19, 23, 26, 28, 32, 41, 46, 33, 50, 52], which constitute an improvement over credential based trust management. However, based on our discussion in section 3 some limitations still exist:

- Most systems use fixed trust domains (e.g. [1, 26, 28, 32, 52]). This can hamper the flexibility of the trust model for different applications.
- Few incorporate a notion of uncertainty to deal with partial information or the introduction of new unknown entities (e.g. [26, 52]).
- Few include detailed risk analysis or clearly model the relationship with trust [16, 19], usually incorporating only an arbitrary risk threshold (e.g. [13, 32, 46]).

The rest of this section focusses on the most important of the evidence based trust management systems and highlights their most interesting aspects.

Stephen Marsh [32] was the first to formalise the notion of trust for computational use, taking into account trust disposition and even a limited notion of context in situational trust. Furthermore, the notion of risk is incorporated to provide a threshold for trusting decision making, demonstrating the relationship between trust and risk, although perhaps simplistically. Simple linear equations allow the formation of trust values in the range [-1, 1). This simple trust domain, however, cannot represent uncertainty, and coupled with the simple linear equations, caused some counter-intuitive outcomes in certain cases of trust evaluation. In our model, the use of more complex functions operating over an application specific partially ordered trust domain negates these problems. In [32], trust evidence observed from past interactions comes in the form of payoff values, in a game theoretical sense. While this allows for limited evolution of trust, a comprehensive approach with more detailed evidence, allowing propagation and

deeper evaluation would be more useful in terms of evolution. This is one of the strongest features of our model.

Later work in evidence based trust management looked further into the use of direct experience and its effect on trust dynamics. In section 3.3 we referred to the work by Jonker and Treur [23] on the formalisation of trust dynamics. This is interesting formal work regarding the formation and evolution of trust in light of personal experience, providing a framework able to capture a variety of dynamic models. Each event that can influence the degree of trust in a subject is interpreted as either trust negative or trust positive, similarly to our notion of attraction. However, our model also operates on the basis of a certainty dimension and allows information to be propagated in a weighted manner. The work caters for initial trust disposition and trust dynamics, in line with our formation and evolution concepts. The trust domain is represented by a partially ordered set of trust qualifications, T, which may be qualitative, quantitative or some dimensional variant, in line with the flexible nature of our trust domain. However, the certainty ordering requirement of our domain enhances the model for the global computing environment. This framework is not meant as a coherent approach to trust management, but rather an analysis of trust dynamics and as such exploitation for decision making is not addressed, nor are the issues of context and risk.

In section 3.3 we discussed the need for types of evidence other than direct experience for evaluating trust in a principal within a decentralised environment. Abdul-Rahman et al. [1] propose a decentralised approach to trust management incorporating distinct trust levels and dynamics. The work focuses on the formation of trust based on recommendations and experiences, rather than exploitation. The decision making process is not detailed, and as such, a notion of risk is not considered. At any given time, trust in an agent is evaluated from the relevant subset of experiences for a context. Similarly to our approach, an experience is the result of either evaluating an interaction or relying on a recommendation from an agent. In [1], experience takes a value corresponding to the trust degree, such that the experience value that has occurred most frequently in past interactions dictates the level of trust. A notion of uncertainty is incorporated, for situations where there is no single most frequent experience value. In our model, however, the notion of attraction of experience permits more complex dynamics to be modelled, and certainty is reflected to some degree in every trust value in the domain. Furthermore, we can use this certainty, coupled with risk analysis, when deciding about collaboration with unknown entities. This permits experience to be gained, rather than bootstrapping with arbitrarily trusted experiences and recommendations. An important contribution of this work is the adjustment of recommendations based on the 'recommender trust' that weights recommendations based on 'semantic distance' of experience from recommendation. In the development of our adjustment operator for trust in the recommender we will be examining similar approaches. In Abdul-Rahman's work, a recommendation can be direct or a lead to a recommender, which means that chains, and the inherent problem they bring in terms of discounting, can

be omitted by contacting the final recommender directly. This element may be of use to us in developing our approach to recommendations.

Other trust management systems have addressed the issue of finding leads to recommendation sources, through neighbourhoods of known entities, as discussed in section 4.3. An example of this is Kinateder and Rothermel [28], who describe a distributed reputation system for trust building in an online environment. Models represent trust in entities in various categories of expertise and algorithms are described for calculation and update of trust based on experiences. Each user has a local copy of the system's trust and knowledge models. The Trust Model has trust values in the range [0, 1] for a set of categories, and a confidence vector that stores meta-information such as number of direct experiences and number of indirect experiences from semantically related categories, determined by a directed dependency graph with weighted vertices. The Knowledge Model is a way of creating a local profile of 'who knows what', including local knowledge, enabling advertising personal expertise and locating recommendation sources. Recommendations can be exchanged and combined through neighbourhoods and can be evaluated against personal experience by the user. Updating trust in each category uses the old trust value for the recommender, the new experience (influenced by an ageing factor), the recommender's confidence in the recommendation and a notion of semantic distance of the trust category. This combination of trust information from semantically different categories of expertise is akin to the context mapping issues discussed in 3.2. However, problems arise due to the chosen trust domain and lack of dispositional aspects. Setting initial trust in unknowns to 0 is not satisfactory, as this can also imply previous bad experience. As our model incorporates certainty this is not an issue as we can give the unknown value to newcomers.

Another system based on the notion of neighbourhoods is proposed by Yu and Singh [52]. Their trust model for large open systems of agents uses distributed reputation management. Each agent maintains a model for each acquaintance, with information on their ability to act in a trustworthy manner and refer to trustworthy agents. The representation of trust is based on Dempster-Shafer theory [44], showing belief, disbelief and uncertainty, formed and evolved by combination of evidence. Two types of belief are distinguished in the model, local belief based on user evaluation of direct interactions and total belief, which combines local belief with testimonies from witnesses. This is in line with our separation of T_{obs} and T_{ov} in the trust information structure in section 4.2. The local belief can be used to establish trust in acquaintances, and neighbours can be queried for testimonies (or referral to testimonies) to allow formation on an initial opinion in an unknown agent. However, without a notion of disposition, it is unclear what happens in the case of an agent unknown to any recommenders. The set of referral chains returned by a testimony query can be combined by Dempster's rule of combination to give the total belief. This is then be compared to a threshold for decision making, although the nature of this process is unspecified. The agent can then become an acquaintance when we gain direct experience with it through interaction. Only local belief is propagated, as total

belief could create non-well-founded cycles. This is consistent with our reasoning behind the trust information structure in section 4.2. However, in [52], it is stated that as more evidence is gathered, uncertainty in the belief that the agent is trustworthy is reduced. This is inconsistent with our view that evidence which contradicts much of the historical trend should make us less certain, a factor incorporated in the certainty dimension of attraction in our model.

We have acknowledged throughout this document that uncertainty is key to representation of trust in the global computing environment. One of the main contributions in this area is Jøsang's work on Subjective Logic [26, 24], which operates on subjective beliefs (opinions) using standard and non-standard logical operators. An opinion is a probability measure containing uncertainty, represented by a belief model similar to Dempster-Shafer Theory [44]. Aside from the ability to represent uncertainty, the interesting aspects of this work are the two non-traditional operators. These allow discounting of recommended opinions based on an opinion of the advice (referred to in section 3.3) and reaching consensus between two opinions based on the same facts, as if an imaginary entity's opinion represented both. Such an operator could be utilised in calculating a combined opinion for evaluation. Furthermore, an alternative representation of uncertain probabilities with respect to the evidence space is defined using probability density functions, based on the amount of evidence supporting the event and its negation. A mapping is easily defined between the evidence space and the opinion space to allow the use of results from one in the other. Moreover, the combination of evidence from two observers forms the basis for the opinion space consensus operator. This can be seen as a link between risk, i.e. behavioural profiles, and the trust domain. However, our view of risk incorporates a range of possible outcomes, not just a binary view. Furthermore, our model retains uncertainty throughout the decision and evaluation processes, across the relationship between trust and risk.

We have seen that few trust management systems provide for detailed risk analysis. In addition to Jøsang's work above, Grandison and Sloman [19, 20] also contribute in this area with their general-purpose trust management system. SULTAN is a logic based notation with associated tools for specification, analysis and management of trust relationships for Internet applications. Trust relationships, or specifications, with constraints and trust values are defined in a policy language. Recommendations can also be defined as policies, used to define new trust relationships. It is up to the trustor to decide if a recommendation is accepted, although there is no process for adjusting based on trust in source, which is an important part of our model. There are several interesting components of SULTAN. A Monitoring Service updates a central store with state information for the scenario and system, experience information for direct alteration of trust levels, and risk information for analysis through a Risk Evaluation Service. The view is taken that a trust specification may depend on risk, defined as a probability of a transaction failing, such that higher risk means less trust, although the detailed risk analysis is not addressed. A Trust Analysis component evaluates the trust and recommendation specifications to determine conflicts and implicit

relationships, although the exploitation of these relationships is not addressed as it is dependent on specific purpose. Evidence evaluation is not defined further in this work, resulting in limited evolutionary capabilities. A notion of context is defined as the set of actions to which a trust level applies, and a set of constraints that must be evaluated for the trust relationship to apply. This differs somewhat from our model, where contextualisation also occurs during exploitation rather than just in the evaluation of evidence.

The notion of context is very important for trust based decision making, as we discussed in section 3.2. There are other trust management systems which address this issue. One good example is Xiong and Liu's PeerTrust [50, 51], an adaptive trust model for P2P e-commerce communities. The basic trust model is based on transaction service satisfaction feedback, collected and distributed by a feedback system. Interestingly, feedback to the evaluation process is weighted according to the number of transactions that contributed to it as well as the feedback source. This means that the amount of information affects the decision. In our model, we take a more integrated approach, by defining certainty as a property of our trust domain. This certainty can change based on the evaluation of evidence against the trend of previous behaviour, encoded in the current trust value. Certainty can therefore be considered directly in both the decision and the evaluation processes. In Xiong and Liu's work, two further adaptive trust factors are defined for feedback, transaction context for aggregating feedbacks from several transaction types, and community context to account for specific community characteristics. A general trust metric for calculating trust in a peer over a specific period is given, which allows further weighting of transaction vs. community characteristics. Thus the metric may be adapted by the developer if community characteristics form an important part of the decision process. The trust value can be exploited for comparison of peers or to determine whether to transact with a specific peer on the basis of a decision rule, such as comparison with a threshold of trust. However, the decision process does not incorporate a notion of risk analysis or the disposition of the entity. Furthermore, personal observations are not treated separately from the evidence of other peers and cannot be implicitly trusted regardless of one's own reputation. The manner in which credibility of source is established is not strictly specified, although it is suggested that this might be done using a function of the trust value for the peer or feedbacks on feedbacks. This would introduce a notion of trust in recommender, which our model currently incorporates. Furthermore, in this work there is a limited view of trust evidence as user rated satisfaction of service and initial trust formation in unknowns not addressed.

Concluding remarks In this section, we have highlighted the shortcomings of other trust management systems for the global computing environment. In contrast, we provide a comprehensive general framework, incorporating the gathering and evaluation of detailed observational evidence, and the propagation of source-weighted recommendations in a context aware fashion to incorporate a range of opinions. While most systems use fixed trust domains, our flexible

trust representation allows a domain to be defined for a particular collaborative application. Moreover, the requirements for both trust and certainty orderings facilitate both evaluation and decision making in a more realistic way than domains with no notion of certainty. Furthermore, this notion of certainty allows us to deal with unknown entities when they are introduced to the system. This problem is further addressed in our model through the use of dispositional characteristics. In contrast to much of the work highlighted here, we model a close relationship between trust and risk, retaining the certainty dimension throughout exploitation of trust for detailed risk analysis in decision making. This relationship is also invaluable for evaluation of the decisions we make, allowing our model to provide more appropriate observational evidence to the trust evaluation process than many other systems. Furthermore, as we separate the gathering of information from policy fulfilment a lack of information will not lead to policy violations, as would happen in credential based systems. An added advantage of this is that evidence can influence a variety of decisions, contributing to decision making in a variety of contexts. The limited evidence and evaluation thereof in other systems leads to limited evolution capabilities. We can examine patterns of historical behaviour rather than just separate pieces of evidence, and by using the attraction, essentially allow all the history to remain at some level within the trust values, albeit with depleting influence over time.

8 Conclusions and Future Work

In conclusion, this appendix builds upon the previously defined SECURE theoretical trust model [9] and risk model [3] to define a collaboration model addressing the dynamic aspects of the trust lifecycle. The model relies on the close relationship between trust and risk, which is expressed as a bijective function *select*() (see equation 3) from the trust to the risk domain. This function is utilised both in the decision making and the trust evaluation process, which view trust as a way of classifying principals according to their expected behaviour. More specifically, in the decision making process, the function allows us to produce a profile of expected behaviour, i.e. risk profile, based on the trustworthiness of a principal. In the trust evaluation process, it allows us to calculate the trustworthiness conveyed by a profile of observed behaviour and subsequently update our opinion about the trustworthiness of a principal. An interesting observation is that if the two processes rely exclusively on this function, then the risk domain must have a structure similar to the one that the theoretical trust model requires from the trust domain. This leads to the introduction of the notion of uncertainty in the risk domain.

The SECURE collaboration model addresses the following dynamic aspects of the trust lifecycle:

- Trust exploitation, which is the interpretation of trust values in context for the purposes of decision making, and

– Trust formation and evolution, which is the revision of the trust values in the light of available evidence starting from complete lack of evidence in the case of unknown principals.

Trust exploitation acknowledges the situational character of trust, which recognises that principals may behave differently in different situations. Subsequently, context is defined as a situational modifier of principals behaviour (see definition 1). Context can be modelled as a set of parameters that affect either the trustworthiness of principals, or the profiles of their expected behaviour, or even the operation of the *select*() function. In the context of a simple collaboration, these parameters include the action in question and the requester. Context can also be divided into internal, intra-application, and external, inter-application one. The latter will require some process of translation or mapping for transfer between applications.

Trust formation and evolution are considered very similar processes. They both receive some evidence, which they evaluate in relation to the current trust value to produce a new trust value. The main difference between them is the use of an initial trust value, most likely "unknown", in the case of formation. Evidence is classified as either direct or indirect. The former is considered a fact and its value is unquestionable. While the value of the latter depends on the trustworthiness of its source and as a result requires appropriate adjustment before its consideration. Direct evidence takes the form of observations, the outcome of an action and its incurred cost or benefit, and indirect the form of recommendations, the recommender's trust value for the subject. The evaluation of evidence is in terms of their attraction, the effect they have on the current trust value and is carried out by an *evaluate*() function (see equation 6). Attraction characterises evidence as either positive or negative in terms of trustworthiness and as either reinforcing or contradicting in terms of certainty. Although recommendations are only evaluated directly through a comparison of their value to the current trust value, observation can be evaluated either directly or indirectly. The latter case, first obtains an evidential trust value, which can then be evaluated in the same way as recommendations. The update of the current trust value is carried by an *evolve*() function, which can either take the form of trust evolution or a trust update function (see equations 11 and 12 respectively). These functions also allow the room for principal specific configuration according to its dispositional characteristics, which take the form of trusting disposition and type of trust dynamics.

The operational considerations of the collaboration model include:

– The definition of a trusting collaboration architecture that supports decision making, trust evaluation and risk evaluation (see figure 5). Important elements of the architecture include a collaboration monitor and an evidence gatherer that collect observations and recommendations respectively, an evidence store that maintains trust related information, both evidence and

trust values, and a trust lifecycle manager responsible for the evaluation of evidence and the update of the current trust value.
- The definition of a layered architecture for trust related information, called the trust information structure (see figure 6). The main characteristic of this architecture is the separate evaluation of the trustworthiness of principals according to direct and indirect evidence. This separation avoids the problems of double counting evidence in the case of recommendations.
- A discussion of the problems associated with the selection of recommenders, where a number of alternative approach are presented.

The formal model for trust lifecycle management uses a policy language for describing the local trust policies of principals to describe the trust formation and evolution process. The formalisation assumes that evidence is organised according to the trust information structure. An interesting result of the formalisation is that it clarifies the difference between references and recommendations. They both refer to other principals' trust values of a subject with the difference that only the latter are evaluated, i.e. the way they are taken into consideration depends on their evaluation in terms of attraction.

The two case studies, the smart space scenario and the e-purse scenario, are only provided as examples of how the collaboration model can be applied. They both focus on the engineering of a trust and risk domain that has the appropriate characteristics so that the decision making and the trust evaluation processes can completely rely on the *select*() function. Moreover, both scenarios provide examples of specific evidence evaluation and trust update functions.

Finally, a comparison to the state of the art shows that our collaboration model takes into consideration all the issues highlighted in the literature, and improves on the current state.

8.1 Future Work

Although, this appendix describes a quite extensive collaboration model for SECURE, there are still a number of areas that require further work and investigation. Future work can take a number of directions. The first priority would certainly be an evaluation of the various aspects of the model. We are in the process of developing a simulation framework to assist us in this evaluation. The starting point in this process is to simulate the two case studies presented in section 6. We also plan to use this simulation framework to investigate desirable properties that our evidence evaluation and trust update functions must exhibit.

The areas that require further investigation in the order that we plan to examine them are:

1. Recommendation adjustment. In order to have a fully operational collaboration model it is essential that we develop an recommendation adjustment operator. A promising starting for this investigation is the notion of semantic distance introduced by Abdul-Rahman and Hayes [1].

2. A detailed model of context. Currently, the discussion about the introduction of context in the collaboration model is still quite high level. As a next step it is necessary to develop a more detailed model of context for trusting collaborations.
3. A collaboration model for non-simple collaborations. Currently, our model is focusing only on simple collaborations, between only two principals involving just a single action. An important extension of our model would be towards multi-principal and multi-action collaborations.
4. An investigation into collaboration monitoring and evidence gathering. The aim of such investigation would be to develop a general collaboration monitor and a set of algorithms for evidence propagation.
5. The relation between trust and risk. Regarding this relationship from a theoretical point of view a possible avenue of investigation could be the development of a unified model of trust and risk. This model would certainly have to include a model for uncertain risks that will build upon our preliminary approach. On the same issue and from an engineering point of view, it would be interesting to investigate the relative advantages and disadvantages of the alternative approaches suggested. This investigation could subsequently be developed into a methodology for developing trusting collaboration applications.

In conclusion, we believe that although the model described in this appendix is a considerable step towards a complete model for trusting collaboration, there is still a long way to go.

References

1. A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6007. IEEE Computer Society Press, January 2000.
2. Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, LNCS, pages 304–307, Karlsruhe, Germany, September 1999. Springer-Verlag.
3. Jean Bacon, Nathan Dimmock, David Ingram, Ken Moody, Brian Shand, and Andy Twigg. Definition of risk model. *SECURE Deliverable 3.1*, 2002.
4. Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. Keynote: Trust management for public-key infrastructures. In *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, volume 1550 of *LNCS*, pages 59–63. Springer-Verlag, 1998.
5. Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. The role of trust management in distributed system security. In *Secure Internet Programming: Security Issues for Mobile and Distributes Objects*, volume 1603 of *LNCS*, pages 185–210. Springer-Verlag, 1999.
6. Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, Los Alamitos, USA, May 1996. AT&T.

7. P. J. Brown, J. D. Bovey, and X. Chen. Context-aware applications: From the laboratory to the marketplace. *IEEE Personal Communications*, 4(5):58–64, October 1997.
8. C. Bryce, V. Cahill, G. Di Marzo Serugendo, C. English, S. Farrell, E. Gray, C. D. Jensen, P. Nixon, J.-M. Seigneur, S. Terzis, W. Wagealla, and C. Yong. Application scenarios. *SECURE Deliverable 5.1*, 2002.
9. Marco Carbone, Oliver Danvy, Ivan Damgaard, Karl Krukow, Anders Mller, Jesper B. Nielsen, and Mogens Nielsen. A model for trust. *SECURE Deliverable 1.1*, 2002.
10. Marco Carbone, Mogens Nielsen, and Vladimiro Sassone. A formal model for trust in dynamic networks. In *Proceedings of the International Conference on Software Engineering and Formal Methods*, pages 54–63, Brisbane, Australia, September 2003.
11. Marco Carbone, Mogens Nielsen, and Vladimiro Sassone. A formal model for trust in dynamic networks. RS RS-03-4, BRICS, DAIMI, January 2003. 18 pp.
12. Deborah Caswell and Philippe Debaty. Creating web representations for places. In *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, pages 114–126. Springer-Verlag, 2000.
13. Rita Chen and William Yeager. Poblano - a distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems, 2001.
14. Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, and Martin Strauss. REFEREE: Trust management for Web applications. *Computer Networks and ISDN Systems*, 29(8–13):953–964, 1997.
15. Anind K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, Georgia Institute of Technology, November 2002.
16. Theo Dimitrakos. System models, e-risks and e-trust. towards bridging the gap? In *Proceedings of the 1st IFIP Conference on e-Commerce, e-Business, e-Government*. Kluwer Academic Publishers, October 2001.
17. Nathan Dimmock. How much is 'enough'? Risk in trust-based access control. In *Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises: Enterprise Security (Special Session on Trust Management)*, Linz, Austria, June 2003.
18. Diego Gambetta. Can we trust trust? In Diego Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–237, Oxford, 1990. Basil Blackwell.
19. Tyrone Grandison and Morris Sloman. Specifying and analysing trust for internet applications. In *Proceeedings of the 2nd IFIP IEEE Conference*, pages 145–157, October 2002.
20. Tyrone Grandison and Morris Sloman. Trust management tools for internet applications. In Paddy Nixon and Sotirios Terzis, editors, *Proceedings of the First International Conference on Trust Management*, volume 2692 of *LNCS*, pages 91–107, Heraklion, Crete, Greece, May 2003. Springer.
21. Bernardo A. Huberman and Fang Wu. The dynamics of reputation. Technical report, HP Laboratories, Palo Alto, CA, USA, November 2002.
22. R. Hull, P. Neaves, and J. Bedford-Roberts. Towards situated computing. In *Proceedings of the 1st International Symposium on Wearable Computers (ISWC'97)*, pages 146–153, Cambridge, Massachusetts, USA, October 1997. IEEE Computer Society Press.
23. Catholijn M. Jonker and Jan Treur. Formal analysis of models for the dynamics of trust based on experiences. In Francisco J. Garijo and Magnus Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in*

a Multi-Agent World : Multi-Agent System Engineering (MAAMAW-99), volume 1647 of *LNCS*, pages 221–231. Springer-Verlag, June 1999.
24. Adun Jøsang. The consensus operator for combining beliefs. *Artificial Intelligence Journal*, 142(1-2):157–170, October 2002.
25. Adun Jøsang, Elizabeth Gray, and Michael Kinateder. Analysing topologies of transitive trust. In Theo Dimitrakos and Fabio Martielli, editors, *Proceedings of the Workshop on Formal Aspects of Security and Trust (FAST2003) at FM2003*, volume TR-10/2003 of *IIT Technical Reports*, pages 9–22, Pisa, Italy, September 2003.
26. Audun Jøsang. A logic for uncertain probabilities. *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.
27. Lalana Kagal, Jeffrey L Undercoffer, Filip Perich, Anupam Joshi, and Tim Finin. A security architecture based on trust management for pervasive computing systems. In *Grace Hopper Celebration of Women in Computing*, October 2002.
28. Michael Kinateder and Kurt Rothermel. Architecture and algorithms for a distributed reputation system. In Paddy Nixon and Sotirios Terzis, editors, *Proceedings of the First International Conference on Trust Management*, volume 2692 of *LNCS*, pages 1–16, Heraklion, Crete, Greece, May 2003. Springer.
29. Ninghui Li, Benjamin N. Grosof, , and Joan Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Transactions on Information and System Security (TISSEC)*, 6(1).
30. Ninghui Li and John C. Mitchell. RT: A role based trust management framework. In *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition (DISCEX III)*. IEEE Computer Society Press, April 2003.
31. Ninghui Li, William H. Winsborough, and John C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1).
32. Stephen Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.
33. M. Marx and J. Treur. Trust dynamics formalised in temporal logic. In L. Chen and Y. Zhuo, editors, *Proceedings of the 3rd International Conference on Cognitive Science, ICCS*, pages 359–363. USTC Press, Beijing, 2001.
34. John McLean. Security models. In J. Marciniak, editor, *Encyclopedia of Software Engineering*. John Wiley & Sons, 1994.
35. G. J. Nelson. *Context-aware and location systems*. PhD thesis, University of Cambridge, 1998.
36. Jason Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2nd International Symposium on Wearable Computers (ISWC'98)*, pages 92–99, Pittsburgh, Pennsylvania, USA, October 1998. IEEE Computer Society Press.
37. N. S. Ryan, J. Pascoe, and D. R. Morse. Enhanced reality fieldwork: the context-aware archaeological assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer Applications in Archaeology 1997*, British Archaeological Reports. Tempus Reparatum, Oxford, UK, October 1998.
38. Ravi Sandhu. Access control: The neglected frontier. In *Proceedings of the First Australian Conference on Information Security and Privacy*, volume 1172 of *LNCS*, pages 219–227, Wollong, Australia, June 1996. Springer.
39. Bill Schilit. *System architecture for context-aware mobile computing*. PhD thesis, Columbia University, 1995.
40. Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of the 1st IEEE International Workshop on Mobile Com-*

puting Systems and Applications, pages 85–90, Santa Cruz, CA, US, 1994. IEEE Computer Society Press.
41. V. Schmatikov and C. Talcott. Reputation-based trust management (extended abstract). In *Proceedings of the Workshop on Issues in the Theory of Security (WITS)*, 2003.
42. K. E. Seamons, M. Winslett, T. Yu, B. Smith, E. Child, J. Jacobson, H. Mills, and L. Yu. Requirements for policy languages for trust negotiation. In *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002)*, pages 68–79, Monterey, CA, USA, June 2002.
43. J.-M. Seigneur, S. Farrell, C. Jensen, E. Gray, and C. Yong. End-to-end trust starts with recognition. In *Proceedings of the First International Conference on Security in Pervasive Computing*, 2003.
44. G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
45. B. Shand, N. Dimmock, and J. Bacon. Trust for Ubiquitous, Transparent Collaboration. In *Proceedings of the First IEEE Annual Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 153–160, Dallas-Ft. Worth, TX, USA, March 2003.
46. Yao-Hua Tan and Walter Thoen. Formal aspects of a generic model of trust for electronic commerce. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6*, page 6006. IEEE Computer Society Press, January 2000.
47. Global Computing Initiative Website. http://www.cordis.lu/ist/fet/gc.htm, 2002.
48. Kerberos Website. http://web.mit.edu/kerberos/www/.
49. Alan F. Westin. *Privacy and Freedom*. Bodley Head, 1970.
50. Li Xiong and Ling Liu. Building trust in decentralized peer-to-peer electronic communities. In *Proceedings of the 5th International Conference on Electronic Commerce Research(ICECR-5)*, Montreal, Canada, October 2002.
51. Li Xiong and Ling Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In *Proceedings of the 4th ACM conference on Electronic commerce*, pages 228–229, San Diego, CA, USA, 2003. ACM Press.
52. Bin Yu and Munindar P. Singh. An evidential model of distributed reputation management. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 294–301, Bologna, Italy, 2002. ACM Press.