University of Amsterdam
Programming Research Group

Local and Global Trust Based on the Concept of Promises

J.A. Bergstra
M. Burgess

Report PRG0606 September 2006

J.A. Bergstra

Programming Research Group
Faculty of Science
University of Amsterdam

Kruislaan 403
1098 SJ Amsterdam
The Netherlands

tel. +31 20 525.7591
e-mail: janb@science.uva.nl

M. Burgess

Faculty of Engineering
University College Oslo

St Olavs Plass
0130 Oslo
Norway

e-mail: mark.burgess@iu.hio.no

Programming Research Group Electronic Report Series

Noname manuscript No.
(will be inserted by the editor)

Local and Global Trust Based on the Concept of Promises

Jan Bergstra[1], **Mark Burgess**[2]

[1] University of Amsterdam, Netherlands
[2] Oslo University College, Norway

20th September 2006

Abstract We use the notion of a promise to define *local trust* between agents possessing autonomous decision-making. An agent is trustworthy if it is expected that it will keep a promise. This definition satisfies most commonplace meanings of trust. Reputation is then an estimation of this expectation value that is passed on from agent to agent.

Our definition distinguishes *types* of trust, for different behaviours, and decouples the concept of agent reliability from the behaviour on which the judgement is based. We show, however, that trust is fundamentally heuristic, as it provides insufficient information for agents to make a rational judgement. A global trustworthiness, or *community trust* can be defined by a proportional, self-consistent voting process, as a weighted eigenvector-centrality function of the promise theoretical graph.

1 Introduction

I don't trust him. We're friends.
–Bertolt Brecht

The decision to trust someone is a policy decision. Although the decision can be made *ad hoc*, our common understanding of trust is that it is based on a gathering of experience, i.e. a process of learning about the behaviour and reputation of someone in a variety of scenarios. Our particular policy might weight certain sources and behaviours more heavily than others and no one can tell us what is the right thing to do. Hence trust is intimately connected with personal autonomy.

In this paper, we define trust in the spirit of this personal autonomy, by basing it directly on the concept of how reliably a promise is kept. A promise is also an autonomously made declaration of behaviour, that is highly individual, moreover it carries with it the notion of a theme (what the promise is about). By combining promises with reliability, we thus have a natural definition of trust that satisfies well-understood rules for revising both the logical aspects of policy and the statistical observations made about agents' behaviours. We show that this viewpoint satisfies the desirable properties for use in computer security schemes. Note that our aim in this work is not to design a technology for building trust, but rather to analyse the precepts for what trust actually means to users.

The plan for this paper is as follows (see fig. 1), We discuss the notion of trust from a pragmatic and philosophical point of view in order to settle on what properties trust should have. We show that common expressions of trust are often ambiguous, but that we can resolve this ambiguity by defining agent trust as the expectation of keeping a given

promise. Using the graphical notions of promises, we can then establish a notion of global trust in certain cases.

1.1 Trust

The concept of trust is both well known and widely used in all kinds of human interactions. Trust is something that humans hold both for one another or sometimes for inanimate objects ("I trust my computer to give the right answer"). In computer systems, the concept of trust is especially used in connection with security. In risk analysis one considers a secure system to be one in which every possible risk has either been eliminated or accepted as a matter of policy. Trust is therefore linked to the concept of policy in a fundamental way.

Trust is also discussed in the case of network security protocols, for instance, in the case where keys are exchanged. The classic dilemma of key distribution is that there is often a high level of uncertainty in knowing the true originator of a secure identifier (cryptographic key). One therefore hopes for the best and, beyond a certain threshold of evidence "trusts" the assumption of ownership. Several protocols claim to manage such trust issues, but what does this really mean?

In spite of the reverence in which the concept is held, there is no widely accepted technical definition of trust. This has long be a hindrance to the discussion and understanding of the concept. The Wikepedia defines: "Trust is the belief in the good character of one party, they are believed to seek to fulfil policies, ethical codes, law and their previous promises." In this paper, we would like to address the deficiencies of discussions of trust by introducing a meta-model for understanding trust. Our model can be used to explain and describe common trust models like "trusted third parties" and the "web of trust".

1.2 Promises – autonomous claims

Trust is an evaluation that can only be made by an individual. No one can force someone to trust someone else in a given situation. This basic fact tells us something important about how trust should be defined.

Recently, one of us has introduced a description of autonomous behaviour in which individual agents are entirely responsible for their own decisions[1–4]. Promise theory is a graphical model of policy. The basic responsibility of an agent to be true to its own assertions is an important step towards a way of describing trust.

Promise theory is useful in this regard because all agents are automatically responsible for their own behaviour and only their own behaviour. Responsibility is not automatically transitive between autonomous agents: it has to be arranged through explicit agreement between agents in a controlled way; hence one avoids problems such as hidden responsibility that make the question of whether to trust an individual agent complex.

In this paper, we argue that the concept of trust can be defined straightforwardly as a *valuation* of a promise – specifically the *expectation* of autonomous behaviour. When we say that we trust something, we are directing this towards the instigator of some promise, whether implicit or explicit. Moreover *reputation* is simply what happens to trust as it is communicated about a network, i.e. it is a 'rumour' that spreads epidemically throughout a network along different paths, and hence develops into a path-dependent estimate of trustworthiness.

The matter of evidence-gathering, in order to justify the expectation value of keeping a promise is subtle, and so we shall discuss this in some detail. We argue that there is insufficient information in the notions of trust or reputation to make a reliable estimate of trustworthiness. Thus trust is an inherently ambiguous concept; each valuation of trustworthiness is, in essence, an essentially *ad hoc* policy.

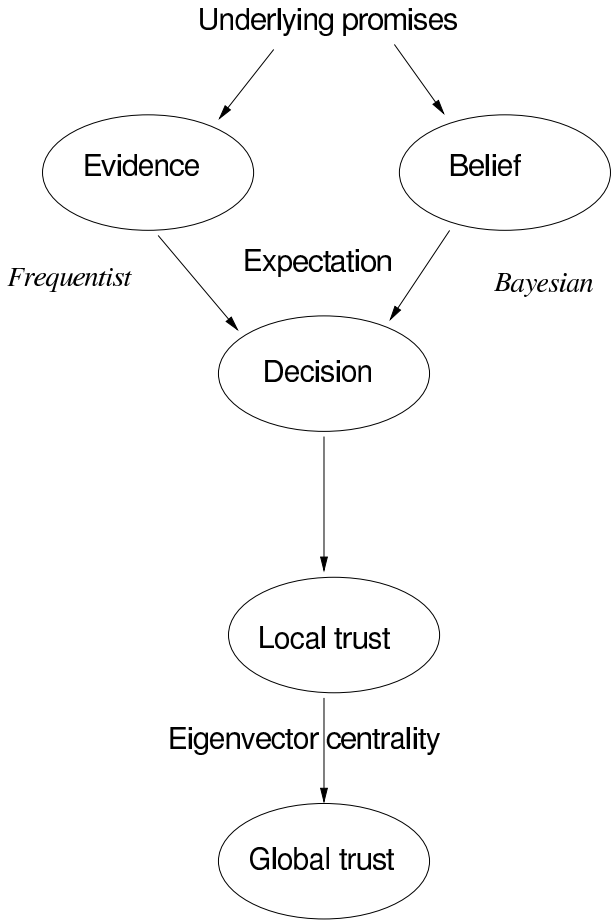

Fig. 1 The chain of trust from verifiable promises to local trust by an agent, to global or community trust which we interpret as reputation.

2 Prior work

There is an extensive literature on trust in computer science[5–10]. Much of it is concerned with generating protocols for the purpose of determining the validity of public keys and other identity tokens, or criticizing these mechanistic views in a wider security perspective. Here we are mainly concerned with general ideas about trust and reputation.

We find the recent work of Klüwer and Waaler to be of interest from the viewpoint of logic[11,12]. These authors present a natural reasoning system about trust which includes the notion of *ordering* by levels of trustworthiness.

The work that seems closest to ours may be found in ref. [13] and ref. [14]. Here the authors distinguish between trust and reputation and provide an epidemic-like procedure for valuating the trust based on some inference rules and numerical measures that are essentially reliabilities. The calculation is hence mainly appropriate for a frequentist interpretation of probability. The authors in ref. [13] are unable to distinguish trust about different issues, or relate these in their model. In ref. [14], an attempt is made at motivating trust types but the underlying properties of these types is not completely clear.

In our proposal:

1. We allow for multiple sources (types) for which trust and reputation are valuated.
2. Our combinatorics are based on logic and on Bayesian probability estimates, which are more appropriate estimators for the small amounts of experience involved.

Other work which we find valuable includes social viewpoints of trust (see ref. [15] for a review). This work brings in the matter of human value judgements, which we feel is an important issue in any definition of trust, since it is humans who make the final decisions in practice. From a sociological viewpoint, there are many forms of currency on which to build trust. Some of these are based on the outcomes of stand-offs such as economic games, bargaining situations and so on[16]. Promises have already been shown to incorporate these considerations neatly within their framework[3].

3 Common usage of trust and reputation

As with most words, the English word 'trust' has a number of related meanings which are worth documenting for reference and comparison.

- Trust implies a confidence or faith character: e.g. one "trusts in friends and family".
- It might be based on an assessment of reliability: e.g. "A trustworthy employee"
- A related, but not identical meaning has to do with presumed safety. It also means to permit something without fear. "I trust the user to access the system without stealing." Such trust can be betrayed.
  This is different because the feeling of safety is not a rationally determined quantity, whereas reliability is observable and measurable. Thus there is both a rational and an irrational aspect to trust.
- A final meaning of trust is the expression of hope, i.e. and expectation or wish: "I trust you will behave better from now on";
  Trust is therefore about the suspension of disbelief. It involves a feeling of benevolence, or competence on the part of the trustee.
  Trust of this kind expresses an acceptance of risk, e.g. a jewelry store trusts that passers-by will not smash a plate glass window very often to steal displayed goods, but rather trusts that the windows will improve sales. There could therefore be an economic decision involved in risk-taking.

Reputation is a related notion to trust. We understand this to mean a received judgement, i.e. an evaluation of an agent's reliability based on hearsay. Reputation spreads like an epidemic process, but it is potentially modified on each transmission. Thus, from a given source, several reputations might emerge by following different pathways (histories) through a network.

4 A typed definition of trust

An agent that is known to keep its promises is considered trustworthy by any normal definition of trust i.e. the agent would be reliable and predictable such that one could put aside one's doubts about whether it might fail to live up to its assertions.

It seems natural then to associate trust with one agent's expectation of the performance of another agent in implementing its promises. This could seem like an unnecessarily narrow definition, but it turns out to be more general than one might expect. What about trust in matters that have not yet occurred? Clearly, trust could be formulated about a future *potential promise*. i.e. a promise does not have been made for us to evaluate its likely reliability. The usefulness of promises is that they encapsulate the relevant information to categorise intentions and actions.

Proposal 1 (Trust) *Trust can be defined as an* agent's expectation *that a promise will be kept. It is thus a probability lying between 0 and 1.*

We shall define "an agent's expectation" in detail below, and we shall additionally give meaning to the concepts of when an agent is deemed to be *trustworthy* or *trusting* which

are global concepts, different from merely *trusted*. This proposal has a number of positive qualities. To begin with it separates the *experiential* aspect of trust from the *nature of the actions* on which it is based. Thus in terms of philosophy of science, it makes a clean distinction between empirical knowledge (expectation) and theoretical knowledge (a promise).

Our definition is specific. The concept of trust, as normally applied in computer science is rather universal and non-specific: either one trusts another agent or one does not; however, it is seldom that we trust or distrust anyone or anything so completely. Our definition is a *typed* definition, i.e. we gauge trust separately for each individual kind of promise – and this is where promises provide a convenient notation and conceptual stepping stone. We assume that promises are a more fundamental notion than trust.

According to our definition, trust is a reliability rating made by some agent that is able to observe two agents involved in a promise. We hesitate to call this a reliability *measure*: for reasons that we shall make clear, there is normally insufficient evidence on which to base a proper reliability estimate, in the sense of reliability theory[17].

A reputation is little more than a rumour that spreads epidemically throughout a network. Common ideas about reputation include.

- "A general opinion of someone."
- "A measure of someone's standing in the community."

Reputation is not necessarily related to trustworthiness. One could have a reputation based on how much money an agent spends, or how much fuel it uses. What characterizes a reputation, as opposed to a personal observation or evaluation, is that it is passed on. One does not observe the characteristic first hand.

Proposal 2 (Reputation) *Reputation can be defined as a valuation of some agent's past or expected behaviour that is communicated to another agent.*

We clarify and develop these basic proposals in the remainder of the paper. In particular trust will be revisited in more detail in section 8.

4.1 Promises

To base our notion of trust on promises, we review the basic concepts from promise theory. Promises are closely linked to the idea of policy, or declarations of autonomous decision-making. Indeed, we define policy to be simply a set of promises, since one can always phrase decisions about how to respond to future events and scenarios as promises about what one will do.

Consider a general set of N agents A_i, where $i = 1, \ldots, N$. We denote agents by capital Roman letters, and shall often use nicknames S for promise-sender or giver, R for promise receiver and T for third parties, to assist the discussion.

Definition 1 (Promise) *A promise is an autonomous specification of unobserved or future behaviour. It involves two agents, a promiser and a promisee, and is announced only to the promisee (if at all). Each promise contains a promise body b that describes the content of the promise. A typical promise from an agent S to an agent R, with body b is written:*

$$S \xrightarrow{\pi:b} R \tag{1}$$

The body b of every promise contains a *type* $t(b)$ and often an additional constraint. Promise types distinguish the qualitative differences between promises, and additional constraint attributes distinguish the quantitative differences. For each promise body b, there is another promise body $\neg b$ which represents the negation of b. We shall assume that $t(\neg b) = t(b)$ and that $\neg\neg b = b$. The negation of b refers to the deliberate act of not performing b, or what ever is the complementary action of type $t(b)$.

Promises fall into two basic complementary kinds, which we can think of as promises for giving and taking, or *service* and *usage*. A promised exchange of some service s requires one of each kind:

$$\begin{aligned} A_1 &\xrightarrow{\pi:s} A_2 \quad \text{or} \quad A_1 \xrightarrow{\pi:+s} A_2 \\ A_2 &\xrightarrow{\pi:U(s)} A_1 \quad \text{or} \quad A_2 \xrightarrow{\pi:-s} A_1 \end{aligned} \tag{2}$$

Exclusive promises are those which cannot physically be realized at the same time. This does not mean that incompatible promises cannot be made, it means that they are meaningless and could lead to problems for the agent.

Definition 2 (Incompatible promises $\#$) *When two promises originating from an agent are incompatible, they cannot be realized physically at the same time. We write*

$$A_1 \xrightarrow{\pi:b_1} A_2 \# A_1 \xrightarrow{\pi:b_2} A_3 \tag{3}$$

If $A_2 = A_3$, we may omit the agents and write $b_1 \# b_2$.

It would probably be unwise for an agent to trust another agent that made simultaneous, incompatible promises. Of course this is a policy decision for each individual agent to make.

Breaking a promise is not the same as not keeping a promise. It is an explicit contradiction. Again, confidence in an agent's promise-keeping ability is reduced when it makes contradictory promises.

Definition 3 (Broken promise) *A promise to another agent is broken if one promise is contradicted by another. We define a broken promise as the promising of exclusive promises*

$$A_1 \xrightarrow{\pi:b_1} A_2,\ A_1 \xrightarrow{\pi:b_2} A_2 \text{ with } b_1 \# b_2 \tag{4}$$

Clearly promising b and $\neg b$ would be excluded.

4.2 Composition of parallel promises

We can compose trivial *bundles* of promises between a single pair of agents by union. Using proof notation:

$$\frac{a \xrightarrow{\pi:b_1} b, a \xrightarrow{\pi:b_2} b, \ldots a \xrightarrow{\pi:b_N} b,}{a \xrightarrow{\pi:b_1 \cup b_2 \cup \ldots b_N} b} \tag{5}$$

where the non-overlap of independent type regions is not necessarily assumed, but helps to make sense of this (the definition should still work even if the types overlap). The composition of promises in a serial fashion is non-trivial and only has meaning in a minority of cases, where autonomy is relinquished.

A promise made conditionally on a Boolean condition C, known to the promising agent is written:

Definition 4 (Conditional promise) *A conditional promise is written:*

$$A_1 \xrightarrow{\pi:b|C} A_2 \tag{6}$$

i.e. A_1 promises b to A_2 if the condition C is true.

Note that a condition can also be the subject of a promise. We write

$$A_1 \xrightarrow{\pi:\mathrm{T}(C)} A_2 \tag{7}$$

for the promise from A_1 to A_2 to ensure that the condition C holds. Now the combination of a conditional promise and the promise of the condition holding leads to an unconditional promise as follows:

$$\frac{A_1 \xrightarrow{\pi:b|C} A_2,\ A_1 \xrightarrow{\pi:\mathrm{T}(C)} A_2}{A_1 \xrightarrow{\pi:b} A_2} \tag{8}$$

A promise is not truly a promise unless the truth of the condition is also promised.

4.3 General notation for promises

The following notation for promises has been designed to be clear and pragmatic, avoiding potential recursion difficulties of promises about promises. We begin with the kind of basic promise from one agent to another and then generalize this:

1. The preferred form of a promise (first kind) is written.

$$S \xrightarrow{\pi:b} R \tag{9}$$

   This is a local and autonomously made promise. This is equivalent to the more general notation:

$$S[S] \xrightarrow{\pi:b} R[R]. \tag{10}$$

   i.e. S promises b to R.
2. A promise of the second kind allows obligation:

$$S[T] \xrightarrow{\pi:b} R \tag{11}$$

   i.e. S promises R that it will oblige T to act as if it had promised b to R. If T is autonomous, this is forbidden and has no influence on T.
3. A promise of the third kind allows indirection.

$$S \xrightarrow{\pi:b} R[T] \tag{12}$$

   i.e. S promises to R that S will do b for T.
4. The most general form of a promise:

$$S[T] \xrightarrow{\pi:b} D[U] \tag{13}$$

   i.e. S promises D that b will act as if it had promised b to U. If T is autonomous, this is forbidden and has no influence of T.

We have potentially a need for all of these variants.

Example 1 Promises about policy in which one does not inform the promise recipient (e.g. the Border Gateway Protocol (BGP) policy) may be written:

$$S \xrightarrow{\pi:b} S[R] \tag{14}$$

i.e. S makes a promise only to itself to honour b toward R (e.g. suppose b is a promise to use packet-data received). This is the case, for instance, in the processing of Access Control Lists by most network devices: the sender of data has no *a priori* idea of whether the device will accept it.

Most of the promises we shall consider in the definition of trust will be of the form

$$A_2 \xrightarrow{\pi:b} A_1[A_3] \tag{15}$$

i.e. a neighbouring agent promises us that it will do something for some third party (where the third party might also be us).

4.4 A general expression for trust

Trust is somehow complementary to the idea of a service promise. This is suggested by the intuition that a promise to *use* a service implies a measure of trust on the part of the receiver. We consider trust a directed relationship from a *truster* to a *trustee*. Moreover, it is a judgement or *valuation* of a promise performed entirely by the *truster*.

We need a notation to represent this, similar to that for promises. In the spirit of the promise notation, we write the general case as:

$$S[T] \xrightarrow{\tau:b} R[U] \tag{16}$$

meaning that S trusts R to ensure that T keeps a promise of b to U.

In most cases, this is too much generality. In a world of autonomous agents, no agent would expect agent S to be able to ensure anything about agent T's behaviour. The more common case is therefore with only three parties

$$A_1[A_2] \xrightarrow{\tau:b} A_2[A_3] \tag{17}$$

i.e. agent A_1 trusts agent A_2 to keep its promise towards some third-party agent A_3. Indeed, in most cases A_3 might also be identified with A_1:

$$A_1[A_2] \xrightarrow{\tau:b} A_2[A_1] \tag{18}$$

which, in turn, can be simplified to

$$A_1 \xrightarrow{\tau:b} A_2. \tag{19}$$

In this case, trust is seen to be a dual concept to that of a promise. If we use the notation of ref. [3], then we can write trust as one possible valuation $v : \pi \rightarrow [0,1]$ by A_1 of the promise made by A_2 to it:

$$A_1[A_2] \xrightarrow{\tau:b} A_2[A_1] \;\leftrightarrow\; v_1(A_2 \xrightarrow{\pi:b} A_1) \tag{20}$$

This is then a valuation on a par with economic valuations of how much a promise is worth to an agent[3]. The recipient of a promise can only make such a valuation if it knows that the promise has been made.

Proposal 3 *Trust of an agent S by another agent R can exist if agent R is informed that agent S has made a promise to it in the past, or if the recipient of the promise R is able to infer by indirect means that S has made such a promise.*

Thus any agent can formulate a trust policy towards any other agent. The only remaining question is, on what basis should such a judgement be made?

Our contention is that the most natural valuation to attach to trust is an agent's estimate of the expectation value that the promise will be kept, i.e. an estimate of the reliability of the agent's promise.

$$A_1[A_2] \xrightarrow{\tau:b} A_2[A_1] \;\overset{P}{\equiv}\; E_1(A_2 \xrightarrow{\pi:b} A_1) \tag{21}$$

where $\overset{P}{\equiv}$ means 'is defined by policy as', and the expectation value $E_R(\cdot)$, for agent R has yet to be defined. We note the essential difficulty: that such valuations of reliability are not unique. They are, in fact, entirely subjective and cannot be evaluated without ad hoc choices of a number of free parameters. We return to this point below.

5 Expectation

An expectation function is a statistical concept that relies either on a body of evidence, or alternatively on a belief informed by limited observation. Such evidences or beliefs are summarized by a probability distribution over the different possible outcomes. We shall consider mainly the outcomes "promise kept" and "promise not kept", though varying degrees are possible.

The notion of an expectation value is well known from the theory of probability and can be based on either classical frequentist-probability or Bayesian belief-probability[18]. There is, for this reason, no unique expectation operator.

Why dabble in intangibles such as beliefs? Computer systems are frequently asked to trust one another without ever having met (for example when they automatically download patches and updates from their operating system provider, virus update from third-parties or even accept the word of trusted third parties in identification) – thus they have little or no empirical evidence to go on. Each time they interact however, they are able to revise their initial estimates on the basis of experience. In this regard, a Bayesian view of probability is a natural interpretation, see e.g. [19]. This is a subjective view of probability that works well with our subjective agents.

Definition 5 (Expectation function $E(X)$) *Given random variables X, Y, an expectation operator or function has the properties:*

1. *If $X \geq 0$, $E(X) \geq 0$.*
2. *If $a, b \in \mathbb{R}$, then $E(aX + bY) = aE(X) + bE(Y)$.*
3. *E(1) = 1.*

For a probability distribution over discrete classes $c = 1 \ldots C$, it is the convex sum

$$E(X) = \sum_{c=1}^{C} p_c X_x \quad \Big| \quad \sum_{c=1}^{C} p_c = 1. \tag{22}$$

The expectation value of a Bernoulli variable (with value 0 or 1) is clearly just equal to the probability of obtaining 1 $p_1 = Pr(X = 1)$. In general, a promise might lead to more than one outcome, several of which might be acceptable ways of keeping the promise, however this possibility only complicates the story for now, hence we choose to consider only the simplest case of

Definition 6 (Agent expectation $E_A(a \xrightarrow{\pi:b} c)$) *The agent expectation $E_A(X)$ is defined to be the agent A's estimation of the probability that a promise $a \xrightarrow{\pi:b} c$ will be kept.*

This can be realized in any number of different ways, e.g. as a mapping from an ensemble of evidence of size N (with the binary outcomes 0 and 1) into the open interval:

$$E_A : \{0, 1\}^N \rightarrow [0, 1] \tag{23}$$

or it could could be an ad hoc value selected from a table of pre-decided values.

5.1 Ensembles and samples of evidence

An ensemble is a collection of experiments that test the value of a random variable. In one experiment, we might evaluate the agent expectation to be p_1. In another, we might evaluate it to be P_1. What then is the probability we should understand from the ensemble of both? We expect that the appropriate answer is an average of these two values, but what if we attach more importance to one value than to the other? Probabilities discard an essential piece of information: the size of the body of evidence on which they are based. Let us consider this important point for a moment.

5.1.1 Frequentistic interpretation In the frequentist interpretation of probability, all estimates are based on past hard evidence. Probabilities are considered reliable as estimators of future behaviour if they are based on a sufficiently large body of evidence. Let lower-case $p_1 = n_1/n$, the probability of keeping a promise, be based on a total of n measurements, of which the *frequencies* n_1 were positive and n_0 were negative, with $n_1 + n_0 = n$. Also, let upper-case $P_1 = N_1/N$ be an analogous set of measurements for which $N \neq n$. How should we now combine these two independent trials into a single value for their ensemble?

In the frequentist interpretation of probability, the answer is clear: we simply combine all the original data into one trial and see what this means for the probabilities. Rationally, the combined probability E for the ensemble must end up having the value $E = (n_1 + N_1)/(n + N)$. If we express this result in terms of the probabilities, rather than the frequencies, we have

$$E = \left(\frac{n}{n+N}\right) p_1 + \left(\frac{N}{n+N}\right) P_1 = \frac{n_1 + N_1}{n+N} \tag{24}$$

This leads us to the intuitive conclusion that the probabilities should be combined according to a weighted average, in which the weights are chosen to attach proportionally greater importance to the larger trial:

$$E = \alpha_1 p_1 + \alpha_2 P_1, \quad \alpha_1 + \alpha_2 = 1. \tag{25}$$

In general, then, with T trials of different sizes, the result would be a convex combination of the expectations from each trial:

$$E = \sum_{i=1}^{T} \alpha_i p_i, \quad \sum_{i=1}^{T} \alpha_i = 1. \tag{26}$$

In the case that there are more possible outcomes than simply 0 and 1, the same argument applies for each outcome.

The problem occurs when we do not have complete knowledge of the sample sizes $n, N, \ldots$ etc., for, in this case, we can only guess the relative importances α_i, and choose them as a matter of policy. If, for example, we could choose to make all the α_i equally important, in which case we have no control over the importance of the expectations.

This so-called *frequentist* interpretation of expectation or probability generally requires a significant body of evidence in the form of independent events to generate a plausible estimate. However, in most ad hoc encounters, we do not have such a body of evidence. Trust is usually based on just a handful of encounters, and one's opinion of the current evidence is biased by prior expectations. Hence, we turn to the alternative interpretation or Bayesian probability.

5.1.2 Bayesian interpretation The policy formula in eqn. (26) is essentially a Bayesian belief formula, which can be derived from the classic Bayes interpretation for *a posteriori* belief.

Suppose we devise an experimental test e to determine whether a hypothesis H of expected trustworthiness is true. We repeat this test, or borrow other agent's observations, thus collecting n of these $e_1 \ldots e_n$. The result for $P(H|e_n, e)$, our belief in the trustworthiness-hypothesis given the available evidence, changes by iteration according to:

$$P(H|e_n, e) = \frac{P(H|e_n) \times P(e|e_n, H)}{P(e|e_n, H)P(H|e_n) + P(e|e_n, \neg H)P(\neg H|e_n)} \tag{27}$$

where we feed back one value $P(H|e_{n-1}, e)$ from the previous iteration as $P(H|e_n)$, and we must revise potentially two estimates on each iteration:

- $P(e|e_n, H)$ is our estimate that the test e will show positive as a direct result of the Hypothesis being true, i.e. because the host was trustworthy.
- $P(e|e_n, \neg H)$ is our estimate of how often e is true due to other causes than the hypothesis H of trustworthiness, e.g. due to trickery.

Note that $P(\neg H|e_n) = 1 - P(H|e_n)$. This gives us a definite iterative procedure based on well-accepted Bayesian belief networks for updating our policy on trust[19]. It can easily be seen that eqn. (26) has this form, but lacks a methodology for rational policy-making.

The advantage of a Bayesian interpretation of policy then, is that it fits well with the notion of trust as a policy decision.

5.2 Policy and rationality

What kind of policy should be employed in defining the expectation of future behaviour? Probability theory is built on the assumption that past evidence can motivate a prediction of the future. At the heart of this is an assumption that the world is basically constant. However, future prediction is the essence of gambling: there are scenarios in which evidence of the past is not an adequate guide to future behaviour. An agent might also look elsewhere for guidance.

- *Initialization*: An agent of which we have initially no experience might be assigned an initial trust value of $1, \frac{1}{2}$, or 0 if we are respectively trusting, neutral or un-trusting by nature.
- *Experience*: One's own direct experience of a service or promise has primacy as a basis for trusting an agent in a network. However, an optimistic agent might choose not to allow the past to rule the future, believing that agents can change their behaviour, e.g. "the agent was having a bad day".
- *Advice*: An agent might feel that it is not the best judge and seek the advice of a reputable or trustworthy agent. "Let's see what X thinks". We shall use this idea in section 9 to define a global trustworthiness.
- *Reputation*: Someone else's experience with a promise can serve as an initial value for our own trust.
- *Damnation*: Some agents believe that, if an agent fails even once to fulfil a promise, then it is completely un-trustworthy. This extreme policy seems excessive, since there might be reasons beyond the control of the agent that prevent it from delivering on its promise.

If we lack any evidence at all about the trustworthiness of an agent with respect to a given promise, we might adopt a policy of using the agent's record of keeping other kinds of promises.

Proposal 4 (Transference of evidence) *In the absence of direct evidence of type $t(b)$, in a promise body b, one may use a policy determined mixture of values from other types as an initial estimate.*

The rationality of such a procedure can easily be questioned, but there is no way to rule out the ad hoc decision as a matter of policy.

6 Cases: The underlying promises for trust idioms

To ensure that our definition of trust is both intuitive and general, we present a number of 'use-cases' below and use these to reveal, in each case, the expectation of a promise that underlies the trust. In each case, we write the declarations of trust, in notation, in words, and as an expectation value of an underlying promise. In some cases, the expressions of trust are ambiguous and support several interpretations which can only be resolved by going to a deeper explanation in terms of promises.

- *I trust my computer to give the right answer.* This could literally mean that one trusts the computer, as a potentially unreliable piece of hardware:

$$\mathrm{Me} \xrightarrow{\tau:\mathrm{answer}} \mathrm{Computer} \stackrel{P}{\equiv} E_{\mathrm{Me}}(\mathrm{Computer} \xrightarrow{\pi:\mathrm{answer}} \mathrm{Me}) \tag{28}$$

  i.e. I expect that the computer will keep its (implicit) promise to furnish me with the correct answer.
  However, there is another interpretation. We might actually (even subconsciously) mean that we trust the company that produces the software (the vendor) to make the computer deliver the right answer when asked, i.e. I expect the promise by the vendor to me, to make the computer give me the right answer, will be kept.

$$\begin{aligned}&[\mathrm{Me}][\mathrm{Computer}] \xrightarrow{\tau:\mathrm{answer}} [\mathrm{Vendor}][\mathrm{Me}]\\ &\qquad \stackrel{P}{\equiv} E_{\mathrm{Me}}\Big([\mathrm{Vendor}][\mathrm{Computer}] \xrightarrow{\pi:\mathrm{Answer}} [\mathrm{Me}][\mathrm{Me}]\Big)\end{aligned} \tag{29}$$

  In either case, the relationship between the promise, the expectation and the trust is the same.
- *I trust the identity of a person (e.g. by presence, public key or signature).*
  This is one of the classic problems of security systems, and we find that the simple statement hides a muddle of possibilities. It has many possible interpretations; however, in each case we obtain clarity by expressing these in terms of promises.

$$\mathrm{Me} \xrightarrow{\tau:\mathrm{Authentic}} \mathrm{Signature} \stackrel{P}{\equiv} E_{\mathrm{Me}}(\mathrm{Signature} \xrightarrow{\pi:\mathrm{Authentic}} \mathrm{Me}) \tag{30}$$

  In this version, we place trust in the implicit promise that a credential makes of being an authentic mark of identity. This is a simple statement, but we can be sceptical of the ability of a signature to make any kind of promise.

$$\begin{aligned}&\mathrm{Me}[\mathrm{Signature}] \xrightarrow{\tau:\mathrm{Authentic}} \mathrm{Certifier}[\mathrm{Me}]\\ &\qquad \stackrel{P}{\equiv} E_{\mathrm{Me}}(\mathrm{Certifier}[\mathrm{Signature}] \xrightarrow{\pi:\mathrm{Authentic}} \mathrm{Me})\end{aligned} \tag{31}$$

  i.e. I trust a Certifying Agency to ensure that the implicit promise made by the credential to represent someone is kept. Or I expect the certifying agency (possibly the originator of the signature himself) to keep a promise to me to ensure that the signature's promise to me is kept (e.g. the technology is tamper-proof).
  Yet a third interpretation is that the trust of the key is based on the promise to verify its authenticity, on demand. This is the common understanding of the "trusted third party".

$$\mathrm{Me} \xrightarrow{\tau:\mathrm{verify\ key}} \mathrm{Certifier} \stackrel{P}{\equiv} E_{\mathrm{Me}}\Big(\mathrm{Certifier} \xrightarrow{\pi:\mathrm{verify\ key}} \mathrm{Me}\Big) \tag{32}$$

  i.e. I trust that the key has been authorized and is verifiable by the named Certification Agency. This last case avoids the problem of why one should trust the Certifying Agency, since it refers only to the verification service itself.
- A similar problem is encountered with currency denominations, e.g. pound notes, dollars, or Euros. These tokens are clearly not valuable in and of themselves; rather they represent value. Indeed, on British Pound notes, the words "I promise to pay the bearer on demand the sum of ... X pounds" is still found, with the printed signature of the Chief Cashier. Indeed, the treasury will, if pressed, redeem the value of these paper notes in gold. Thus trust in a ten pound note may be expressed in a number of ways.
  We trust the note to be legal tender: i.e.

$$\mathrm{Me} \xrightarrow{\tau:\mathrm{legal}} \mathrm{Note} \stackrel{P}{\equiv} E_{\mathrm{Me}}\Big(\mathrm{Cashier} \xrightarrow{\pi:\mathrm{gold|note}} \mathrm{Me}\Big) \tag{33}$$

we expect that the chief cashier will remunerate us in gold on presenting the note. Alternatively, we assume that others will promise to accept the note as money in the United Kingdom (UK):

$$\text{Me} \xrightarrow{\tau:\text{legal}} \text{Note} \overset{P}{\equiv} E_{\text{Me}}\left(\text{S} \xrightarrow{\pi:\text{U(note)}} \text{Me}\right), \quad S \in UK \tag{34}$$

Interestingly neither dollars nor Euros make any much promise. Rather, the dollar bill merely claims "In God we trust".

– *Trust in family and friends.*
This case is interesting, since it is so unspecific that it could be assigned almost any meaning. Indeed, each agent is free to define its meaning autonomously. For some set of one or more promises $\mathcal{P}^*$,

$$\text{Me} \xrightarrow{\tau:\mathcal{P}^*} \{\text{Family}\} \overset{P}{\equiv} E_{\text{Me}}\left(\bigcup_{i\in *}\{\text{Family}\} \xrightarrow{\pi:\mathcal{P}_i} A_i\right) \tag{35}$$

i.e. for some arbitrary set of promises, we form an expectation about the likelihood that family and friends would keep their respective promises to the respective promisees. These promises might, in fact, be hypothetical and the evaluations mere beliefs. On the other hand, we might possess actual knowledge of these transactions, and base judgement on the word of one of these family/friend members to keep their promises to the third parties:

$$\text{Me} \xrightarrow{\tau:\mathcal{P}^*} \{\text{Family}\} \overset{P}{\equiv} E_{\text{Me}}\left(\bigcup_{i\in *}\{\text{Family}\} \xrightarrow{\pi:\mathcal{P}_i} \text{Me}[A_i]\right) \tag{36}$$

– *A trustworthy employee.*
In this case, one bases trustworthiness is based more on a history of delivering on promises made in the context of work, e.g.:

$$\text{Boss} \xrightarrow{\tau:\text{Deliver}} \text{Employee} \overset{P}{\equiv} E_{\text{Boss}}(\text{Employee} \xrightarrow{\pi:\text{Deliver}} \text{Boss}) \tag{37}$$

– *I trust the user to access the system without stealing.*
Here the promise is not to steal. The promise does not have to have been made explicitly. Indeed, in civil society this is codified into law, and hence all agents implicitly promise this by participating in that society.

– *"I trust you will behave better from now on!"*
This can be understood in two ways. In the first interpretation, this is not so much an evaluation of trust as it is a challenge (or even warning) to the agent to do better. Alternatively, it can be taken literally as an expression of belief that the agent really will do better. In the latter case, it is:

$$\text{Me} \xrightarrow{\tau:\text{Do better}} \text{You} \overset{P}{\equiv} E_{\text{Me}}\left(\text{You} \xrightarrow{\pi:\text{Do better}} \text{Me}\right) \tag{38}$$

7 Expectations of ensembles and compositions of promises

We are not done with policy's intrusion into the definition of expectation. Since promises can be composed according to straightforward rules, we must be able to compute two distinct things:

1. The expectation of a composition of promises that coexist.
2. The composition of expectations from different ensembles.

The difference between these is analogous to the difference between the combinations of experimental data into ensembles for computing probabilities, and the composition of different probable inputs in fault trees (with AND, OR, XOR, etc).

We have already discussed the composition of data sets into ensembles, the effect this has on probabilities, and how this is expressed in terms of the basic expectation values in section 5.1

We shall have need to define the meaning of the following in order to determine the trust deriving from compound promises:

1. The expectation of incompatible promises.
2. The expectation of a composition of parallel promises between a pair of agents.
3. The expectation of a composition of serial promises between a chain of agents.

7.1 Parallel promise (bundle) expectation

When promises are made in parallel, the question arises as to how much to trust them as a bundle. Should one ever base one's trust on a complete package or bundle of promises? This is a subjective judgement based on whether certain promises are related in the view of the promisee. If one makes an expectation valuation for each promise individually, does it make sense to combine them as probabilities, e.g. in the manner of a fault tree[20, 17]. One is used to the probability composition rules for binary logic of independent events.

- (AND): If the promisee is dependent on several mutually reinforcing promises, then AND semantics are a reasonable assumption. In a security situation, this might be reasonable. The multiplicative combination rule means that each additional promise that must be in place reduces the total trust that the promiser will keep all of its promises proportionally.
- (OR) Here one says that if one or more promises are kept, then trustworthiness is reinforced. This is an optimistic policy which seems to suggest that the promisee is understanding about the promiser's potential difficulties in keeping a promise. This cannot be applied to incompatible promises.
- (XOR): An alternative scenario is to have a number of promises that are alternatives for one another. For instance, mutually exclusive conditional promises that behave like a switch: e.g.

$$\begin{aligned} S &\xrightarrow{\pi:x|y} R \\ S &\xrightarrow{\pi:x'|\neg y} R, \end{aligned} \tag{39}$$

  i.e. S promises x to R, iff y, else it promises x'.
- (RANKED) If the promises are ranked in their importance to the recipient, then the measure of trust associated with the package is best judged by weighting the importance appropriately. Referring to the discussion in section 5.1, this admits a general convex combination of contributions for ranking an OR (see below).

Let us consider how these are represented as functions.

Definition 7 (Expectation of a promise bundle) *Let S (sender) and R (recipient) be agents that make a number of promises in parallel, the composition of a bundle of parallel promises $S \xrightarrow{\pi:b^*} R$ is a function F_R of the expectations of the individual promises:*

$$E_R\left(S \xrightarrow{\pi:b^*} R\right) \stackrel{P}{\equiv} F_R\left(E_R\left(S \xrightarrow{\pi:b_1} R\right), E_R\left(S \xrightarrow{\pi:b_2} R\right), \ldots\right) \tag{40}$$

The function F_R is a mapping from N promise expectations to a new expectation value:

$$F_R : [0,1]^N \rightarrow [0,1] \tag{41}$$

Several such functions are known from reliability theory, e.g. in fault tree analysis (see for instance ref. [17]). Examples include,

$$F_R^{\text{AND}}\left(S \xrightarrow{\pi:b^*} R\right) = \prod_i E_R\left(S \xrightarrow{\pi:b_i} R\right) \tag{42}$$

$$\begin{aligned} F_R^{\text{OR}}\left(S \xrightarrow{\pi:b^*} R\right) &= 1 - \prod_i \left(1 - E_R\left(S \xrightarrow{\pi:b_i} R\right)\right) \\ &\simeq \sum_i E_R\left(S \xrightarrow{\pi:b_i} R\right) \pm O(E^2) \end{aligned} \tag{43}$$

$$\begin{aligned} F_R^{\text{XOR}}\left(S \xrightarrow{\pi:b^*} R\right) &\simeq 1 - \prod_i \left(1 - E_R\left(S \xrightarrow{\pi:b_i} R\right)\right) \\ &\simeq \sum_i E_R\left(S \xrightarrow{\pi:b_i} R\right) \pm O(E^2). \end{aligned} \tag{44}$$

where $O(E^2)$ denotes terms or the order of the probability squared, which are small. A further possibility is to take a weighted mean of the promise estimates. This better supports the view in section 5.1 about different sizes ensembles and their relative weights. There might be additional (irrational) reasons for giving priority to certain promises, e.g. leniency with respect to a difficult promise.

To combine the different possibilities (analogously to fault trees) one could first reduce products of AND promises into sub-bundles, then recombine these using a weighted estimate.

$$\begin{aligned} F_R^{RANKED} &\stackrel{P}{=} \sum_i \alpha_i E_R\left(S \xrightarrow{\pi:b_i} R\right) \\ \sum_i \alpha_i &= 1 \end{aligned} \tag{45}$$

Note that, due to the reasoning of probability theory, the expectation of something AND something else is less than the probability of either. This might be seen as pessimistic as far as trust is concerned. We have to make a policy decision about whether or not to place any weight on the combined expectation of a bundle of promises, or whether to decide to only allow individual expectations.

For example, suppose an agent makes two contradictory promises about services levels, e.g. promise to respond in 4ms and promise to respond in 5ms.

$$\begin{aligned} S &\xrightarrow{\pi:4} R \\ S &\xrightarrow{\pi:5} R \end{aligned} \tag{46}$$

Formally, this is a broken promise, since both promises cannot be true at the same time. The trust in each individual promise can be estimated independently for the two promises. The agent reliability expectations of delivering "4" or "5" units of service are:

$$R \xrightarrow{\tau:4} S = E_R(4) = p(4) = 0.1 \tag{47}$$

$$R \xrightarrow{\tau:5} S = E_R(5) = p(5) = 0.2 \tag{48}$$

Then we can consider what the expectation of the combination of promises is. If the agent S makes both promises simultaneously, the expectation of the combined promises will be:

$$E_R(4\, XOR\, 5) \simeq \frac{(e_4\, E_R(4) + e_5\, E_R(5))}{(e_4 + e_5)} \tag{49}$$

where e_4 is our estimate of likelihood the agent can deliver "4" and e_5 is the estimate of likelihood of delivering "5". These beliefs can be based on many potential sources of information, chosen as a matter of policy; one possibility is to simply identify $e_4 \stackrel{P}{\equiv} E_R(4)$ and $e_5 \stackrel{P}{\equiv} E_R(5)$. Thus a simple policy solution could be to take

$$E_R(4\ OR\ 5) \stackrel{P}{\equiv} \frac{E_R(4)^2 + E_R(5)^5}{E_R(4) + E_R(5)} = 0.17 \tag{50}$$

i.e. in general a sum of squares.

7.2 Incompatible promise expectation

For incompatible promises we must have at least complementary behaviour (NOT):

$$\begin{aligned} E_A(S \stackrel{\pi:\neg b}{\longrightarrow} R) &= 1 - E_A(S \stackrel{\pi:b}{\longrightarrow} R) \\ F_R(E_R(S \stackrel{\pi:\neg b}{\longrightarrow} R)) &= 1 - F_R(E_R(S \stackrel{\pi:b}{\longrightarrow} R)) \end{aligned} \tag{51}$$

Ideally incompatible promises would not be made, without conditionals to select only one of the alternatives.

In the case of AND it is necessary already to resolve the ambiguity in the meaning of the combination of incompatible promises. It is by definition a logical impossibility for incompatible promises to be kept. Thus, while we cannot prevent an agent from promising such nonsense, our expectation of the combination ought to be zero.

Definition 8 (Expectation of incompatible promises with AND) *The expectation of incompatible promises,*

$$F_R\left(A_1 \stackrel{\pi:b_1}{\longrightarrow} A_2\ AND\ A_1 \stackrel{\pi:b_2}{\longrightarrow} A_2\right) \equiv 0 \text{ when } b_1 \# b_2 \tag{52}$$

is defined to be zero for any rational agent.

Hence, in the example above,

$$E_R(4\ AND\ 5) = 0. \tag{53}$$

7.3 Serial promise expectation and transitivity of trust

Several systems base their operation on the idea that trust is to some extent transitive. "The Web of Trust" notion in public key management idea proposes that trust can be conferred transitively. This is not a property of promises, so it is of interest to consider how this works. In other words, if A_1 trusts A_2 to do b, and A_2 trusts A_3 to do b, then A_1 will often trust A_3 to do b. Here b is generally taken to be "reveal one's true identity". This notion does not fit well with a promise theory interpretation of trust because it is type-unspecific.

This is easy to see by noting that

$$A_1 \stackrel{\pi:b}{\longrightarrow} A_2, A_2 \stackrel{\pi:b}{\longrightarrow} A_3 \not\Rightarrow A_1 \stackrel{\pi:b}{\longrightarrow} A_3 \tag{54}$$

i.e. if A_1 makes a promise of b to A_2 and A_2 makes the same promise to A_3, it does not follow that A_1 has made any promise to A_3.

An unspecific trust model might conform to the following property:

$$(i)\ \ (A_1 \text{ Trusts } A_2), (A_2 \text{ Trusts } A_3) \Rightarrow A_1 \text{ Trusts } A_3 \tag{55}$$

In terms of promises, we would interpret this to mean that, if A_1 trusts A_2 (to keep promises to A_1) and A_2 trusts A_3 (to keep promises to A_2) then A_1 should trust A_3 to keep promises

to A_1. This is far from being a rational policy, since there is no evidence passed on about the reliability of agents. A less problematic alternative is:

$$(ii)\ \ (A_1 \xrightarrow{\tau:\text{inform}} A_2), (A_2 \xrightarrow{\tau:b} A_3) \Rightarrow A_1[A_3] \xrightarrow{\tau:b} A_3[A_2] \tag{56}$$

If A_1 trusts A_2 (to inform it about its relations with A_3) and A_2 trusts A_3 (to keep its promise of b to A_2), then A_1 trusts that A_3 is trustworthy in its promise of b to A_2.

The matter of serial promises is one of diverging complication. We make some brief notes about the problems associated with serial promises, and leave the potentially extensive details for elsewhere. The problems with trusting a distributed collection of promises are

1. Promises are not common knowledge, so we do not have all the information.
2. Promises are not transitive.

Knowledge about the promises and the local evaluations by the agents can only be guaranteed by making chains of promises between the agents to share this knowledge.

$$\begin{array}{ccccc} A_1 & \xrightarrow{\pi:\text{tell rep}} & A_2 & \xrightarrow{\pi:\text{tell rep}} & A_3 \\ A_1 & \xleftarrow{\pi:U(\text{tell rep})} & A_2 & \xleftarrow{\pi:U(\text{tell rep})} & A_3 \end{array} \tag{57}$$

In order to pass on the necessary information about trust to a third party, it must be relayed. Expectation of a chain of promises depends on a chain of such trust and Use(trust) promises. However, each agent in the chain agrees only to trust the previous agent. There is no automatic agreement to trust the previous members. If one were to make an explicit promise to trust each agent's information about trust, this would require a promise graph like the one in fig. 2. In order to remove the ambiguity of the trust promises, we must use a

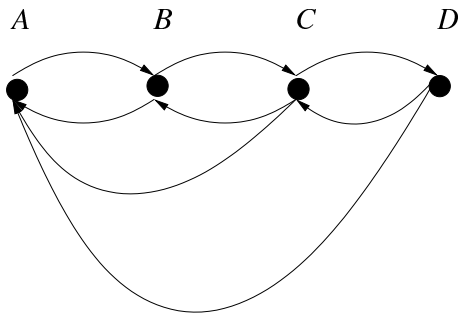

Fig. 2 A chain of trust promises to transfer some valuation of trust in one direction (only), from node a to each agent up to node d. This method is unreliable because nodes b and c are under no obligation to pass on the correct value. Note that these are promise arrows, not trust arrows.

This is clearly a fragile and somewhat complicated structure. An alternative approach is to avoid chains of greater length than one, and also eliminate the extraneous and essentially impotent promises from the chain, as in fig. 3. However, this leads us merely back to the notion of a centralization, either in the form of a trusted party for all agents, or as a complete peer-to-peer graph.

different *promise type* for trust about each agent in the graph. i.e. the trust passed on from agent a must retain this label in being transferred. However, here one has a paradox: if an agent is potentially unreliable, then it can easily lie about this information. Such serial chains are, in general fraught with uncertainty, thus agents might well choose, as a matter of policy, to disregard reputations.

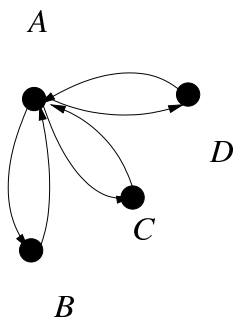

Fig. 3 A more reliable approach of passing on the trust node a holds on to nodes b, c and d.

8 Reputation

We have defined a reputation to be simply a valuation of something (not necessarily a promise) received by an agent about some other agent. A natural basis for reputation (and one that is used on 'reputation systems' in computing) is the valuation of trustworthiness. Here we consider the effect that such transmission of information has on the local trust within a network of agents.

8.1 Borrowed trust

Suppose that and agent T trusts an agent S to keep its promise to R with probability $E_T\left(S \xrightarrow{\pi:b} R\right)$, and suppose that this agent T promises to transmit this as S's reputation to another agent U. U's estimate of the trustworthiness of T's communication is

$$U \xrightarrow{\tau:\text{reputation}} T \stackrel{P}{\equiv} E_U\left(T \xrightarrow{\pi:\text{reputation}} U\right) \tag{58}$$

Can we say what U's expectation for the reliability of the original promise $a \xrightarrow{\pi:b} c$ should be? In spite of the fact that probabilities for independent events combine by multiplication, it would be presumptuous to claim that

$$E_U\left(S \xrightarrow{\pi:b} R\right) = E_U\left(T \xrightarrow{\pi:\text{reputation}} U\right) E_T\left(S \xrightarrow{\pi:b} R\right), \tag{59}$$

since U does not have any direct knowledge of $E_T\left(S \xrightarrow{\pi:b} R\right)$, he must evaluate the trustworthiness and reliability of the source.

Suppose we denote the communicated value of $E_T\left(S \xrightarrow{\pi:b} R\right)$ by $\mathcal{E}_{U\leftarrow T}\left(S \xrightarrow{\pi:b} R\right)$, then one could conceivably (and as a matter of rational policy) choose to define

$$E_U\left(S \xrightarrow{\pi:b} R\right) \stackrel{P}{\equiv} E_U\left(T \xrightarrow{\pi:\text{reputation}} U\right) \mathcal{E}_{U\leftarrow T}\left(S \xrightarrow{\pi:b} R\right). \tag{60}$$

With this notation, we can conceivably follow historical paths through a network of promises.

However, it is important to see that no agent is obliged to make such a policy. Thus trust and reputation do not propagate in a faithfully recursive manner. There is, moreover, in the absence of complete and accurate common knowledge by all agents, an impossibility of eliminating the unknowns in defining the expectation values.

8.2 Promised trust

Trust is an evaluation that is private to an agent. This evaluation can be passed on in the form of a communication (leading to reputation), or it can be passed on as a promise to trust.

- S promises R that S will trust R: $S \xrightarrow{\pi:\tau=0.6} R$.
- S promises R that S will trust T: $S \xrightarrow{\pi:\tau=0.6} R[T]$.

Why would anyone promise a party (R) to trust T without telling R? One reason is that there might be strategic bargaining advantages to doing this[21].

8.3 Updating trust with reputation

An agent can use the reputation of another agent as a sample of evidence by which to judge its trustworthiness. It can then attach a certain weight to this information according to its belief, in order to update its own trust. The weighted addition modifies the old trust value T with the new reputation data R.

$$E \mapsto \frac{w_{\text{new}} R + w_{\text{old}} T}{w_{\text{new}} + w_{\text{old}}} \tag{61}$$

This is indistinguishable from a Bayesian update.

9 Global Measures of Trust

Which are the most trusted agents in a network? Trust has so far been measured at the location of each individual agent. The valuation is private. A trust valuation becomes an agent's reputation when the valuation is passed on to others. The passing-on includes a revisional belief process too; this is also a Bayesian posterior probability update process, just like the case of basing trust on different ensembles in section 5.1.

Let us postulate the existence of a vector of received trusts that is available to any particular agent. The agent is then able to combine this information to work out a global measure, which we can call *community trust*. This is analogous to the graphical security model in [22].

The trust matrix T is defined as follows. The (A, B)-th element of the matrix

$$T_{AB}(b) \equiv E_A(B \xrightarrow{\pi:b} *) \tag{62}$$

is A's trust in B with respect to all promises of type b.

Definition 9 (Community trust (Trustworthiness and trustingness)) *The global or community trust is defined by the principal eigenvectors of T and T^{T}. Since this is a transmitted quantity by definition it is a reputation.*

The global reputations for being trustworthy **W** *are defined by the normalized components of the principal eigenvector of the transpose matrix:*

$$T_{BA} W_B = \lambda W_A. \tag{63}$$

The global reputations for being most trusting **S** *are defined by the normalized components of the principal eigenvector*

$$T_{AB} S_B = \lambda S_A. \tag{64}$$

An agent is said to be trusting if it assigns a high probability of keeping its promises to those agents that it trusts. An agent is said to be trustworthy if other agents assign it a high probability of keeping promises to it.

Observe that, in the absence of labels about specific agent relationships, the concepts of *trustworthiness* and *trustingness* for an agent A are properties of the global trust graph that has A as a source, and not of an individual agent, since they are derived from relationships and by voting.

We can easily show that this has the property of a proportional vote. Let v_i denote a vector for the trust ranking, or connectedness of the trust graph, of each node i. Then, the trustworthiness of node i is proportional to the sum of the votes from all of i's nearest neighbours, weighted according to their trustworthiness (i.e. it is just the sum of their trust valuations):

$$v_i \propto \sum_{j=\text{neighbours of } i} v_j \ . \tag{65}$$

This may be more compactly written as

$$v_i = (\text{const}) \times \sum_j T_{ij} v_j \ , \tag{66}$$

where T is the *trust graph adjacency matrix*, whose entries T_{ij} are 1 if i is a neighbour of j, and 0 otherwise. We can rewrite eqn. (66) as

$$T\,\mathbf{v} = \lambda \mathbf{v} \ . \tag{67}$$

Now one sees that the vector is actually an eigenvector of the trust matrix T. If T is an $N \times N$ matrix, it has N eigenvectors (one for each node in the network), and correspondingly many eigenvalues. The eigenvalue of interest is the principal eigenvector, i.e. that with highest eigenvalue, since this is the only one that results from summing all of the possible pathways with a positive sign. The components of the principal eigenvector rank how self-consistently 'central' a node is in the graph. Note that only ratios v_i/v_j of the components are meaningfully determined. This is because the lengths $|\mathbf{v}| = \sqrt{\sum_i v_i v_i}$ of the eigenvectors are not determined by the eigenvector equation. We normalize them here by setting the highest component to 1. This form of well-connectedness is termed 'eigenvector centrality' [23] in the field of social network analysis, where several other definitions of centrality exist.

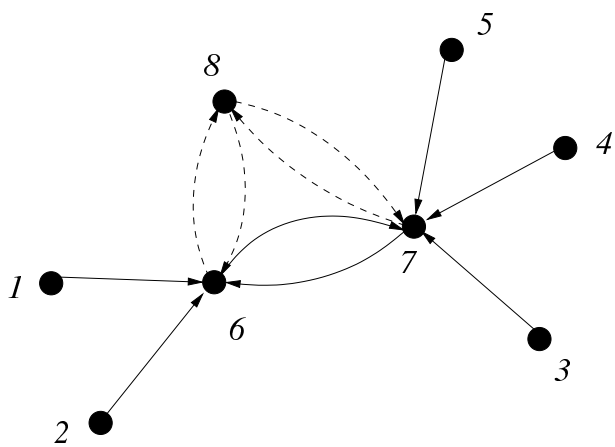

Fig. 4 An example trust graph. For simplicity all trust arrows are assumed of the same type, e.g. trust in the promise to pay bills. Dashed lines are lines which will be removed in the second example.

Note this does not assume any transitivity of trust, it says simply: each agent's trust worthiness is equal the sum of all the other agents' trust measures (as if they are voting), weighted so that the most trustworthy agents' opinions are weighted proportionally highest. It is a proportional representation vote by the agents about one another.

9.1 Example of global trust

Consider a number of promises of a single type, e.g. agents promise to pay their bills in various service interactions. Each payee then rates its expectation of the payer and makes this information globally available as a public measure of its local trust. Referring to fig. 4, we assume the following local trusts:

$$\begin{aligned}
1 &\xrightarrow{\tau:\mathrm{pay}} 6 = 0.2 \\
2 &\xrightarrow{\tau:\mathrm{pay}} 6 = 0.3 \\
3 &\xrightarrow{\tau:\mathrm{pay}} 7 = 0.1 \\
4 &\xrightarrow{\tau:\mathrm{pay}} 7 = 0.1 \\
5 &\xrightarrow{\tau:\mathrm{pay}} 7 = 0.1 \\
6 &\xrightarrow{\tau:\mathrm{pay}} 7 = 0.6 \\
7 &\xrightarrow{\tau:\mathrm{pay}} 6 = 0.5 \\
6 &\xrightarrow{\tau:\mathrm{pay}} 8 = 0.8 \\
8 &\xrightarrow{\tau:\mathrm{pay}} 6 = 0.2 \\
7 &\xrightarrow{\tau:\mathrm{pay}} 8 = 0.8 \\
8 &\xrightarrow{\tau:\mathrm{pay}} 7 = 0.3
\end{aligned} \tag{68}$$

The trust matrix is thus

$$T = \left(\begin{array}{ccccccc|c}
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 & 0.0 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.6 & 0.8 \\
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.8 \\
\hline
0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & 0.3 & 0.0
\end{array}\right) \tag{69}$$

Note that the bars delineate the dashed lines which will be removed in the second example. The normalized right eigenvector $\mathbf{S}_8$ represents how trusting the agents are. The left eigenvector $\mathbf{W}_8$ (or the eigenvector of the transpose matrix) represents the global trustworthiness:

$$\mathbf{S}_8 = \begin{pmatrix} 0.21 \\ 0.31 \\ 0.10 \\ 0.10 \\ 0.10 \\ 1.00 \\ 0.94 \\ 0.50 \end{pmatrix}, \quad \mathbf{W}_8 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.55 \\ 0.65 \\ 1.00 \end{pmatrix} \tag{70}$$

Thus, agent 8 is the most trustworthy. Agents 1 to 5 are not trustworthy at all in this scenario, since we have not rated any promises made by them. Agent 6 is the most trusting of all, since it gives a large amount of trust to agent 8. Thus, these two agents colour the global picture of trust significantly through their behaviours.

We note that the agents with zero trust ratings are all recipients of promises; they do not make any promises of their own. These are suppliers of whatever service or good is being sold; they do not promise payments to anyone, hence no one needs to trust them to pay their bills. The reader might find this artificial: these agents might make it their policy to trust the agents even though they have made no promise. In this case, we must ask whether the trust would be of the same type or not: i.e. would the buyers trust the suppliers to pay their bills, or would their trust be based on a different promise, e.g. the promise to provide quality goods.

By contrast, the agents who are not trusted are somewhat trusting by virtue of receiving such promises of payment.

Suppose we eliminate agent number 8 (by removing the dashed lines in the figure), let us see how the ranking changes when we delete this important agent. Now agent 6 still remains the most trusting, but agent 7 becomes the most trusted, once again mainly due to agent 6's contribution.

$$\mathbf{S}_7 = \begin{pmatrix} 0.37 \\ 0.55 \\ 0.17 \\ 0.17 \\ 0.17 \\ 1.00 \\ 0.92 \end{pmatrix}, \quad \mathbf{W}_7 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.91 \\ 1.00 \end{pmatrix} \tag{71}$$

We can note that the symmetries of the graph are represented in the eigenvector in a natural way.

9.2 Boundaries and allegiances

Canright and Monsen have defined regions of a graph, based on the structures that arise naturally from eigenvector centrality[24]. This has been further developed for directed graphs in ref. [25]. Trust is sometimes associated with maintaining certain boundaries or allegiances. The global trust model proposed above falls into a natural landscape based on the graph, that is characterized by local maxima. Agents cluster naturally into distinct hills of mutual trust, separated by valleys of more tenuous trust, in the centrality function.

This characterization is a useful way of identifying a community structure. Humans are not very good at understanding boundaries: they understand identities. e.g. a company name, but where is the real boundary of the company or computer system? Its tendrils of influence might be farther or closer than one imagines. The topology of underlying promises offers a quantifiable answer to this question. Such allegiances can be compared to the notion of a coalition in game theory[26,27].

10 Trust architectures

Trust is closely associated with information dissemination. There are essentially only two distinct models for achieving information distribution: centralization and *ad hoc* epidemic flooding. Alternatively one might call them, central-server versus peer-to-peer.

Two so-called trust models are used in contemporary technologies today, reflecting these approaches: the Trusted Third Party model (e.g. X.509 certificates, TLS, or Kerberos) and the Web of Trust (as made famous by the Pretty Good Privacy (PGP) system due to Phil Zimmerman and its subsequent clones). Let us consider how these models are represented in terms of our promise model.

10.1 Trusted Third Parties

The centralized solution to "trust management" is the certificate authority model, introduced as part of the X.509 standard and modified for a variety of other systems (See fig. 5)[28–30]. In this model, a central authority has the final word on identity confirmation and often acts as a broker between parties, verifying identities for both sides.

The authority promises (often implicitly) to all agents the legitimacy of each agent's identity (hopefully implying that it verifies this somehow). Moreover, for each consultation the authority promises that it will truthfully verify an identity credential (public key) that is presented to it. The clients and users of this service promise that they will use this confirmation. Thus, in the basic interaction, the promises being made here are:

$$\text{Authority} \xrightarrow{\pi:\text{Legitimate}} \text{User} \tag{72}$$

$$\text{Authority} \xrightarrow{\pi:\text{Verification}} \text{User} \tag{73}$$

$$\text{User} \xrightarrow{\pi:U(\text{Verification})} \text{Authority} \tag{74}$$

To make sense of trust, we look for expectations of the promises being kept.

1. The users expect that the authority is legitimate, hence they trust its promise of legitimacy.
2. The users expect that the authority verifies identity correctly, hence they trust its promise of verification and therefore use it.

Users do not necessarily have to be registered themselves with the authority in order to use its services, so it is not strictly necessary for the authority to trust the user. However, in registering as a client a user also promises its correct identity, and the authority promises to use this.

$$\text{User} \xrightarrow{\pi:\text{Identity}} \text{Authority} \tag{75}$$

$$\text{Authority} \xrightarrow{\pi:U(\text{Identity})} \text{User} \tag{76}$$

One can always discuss the evidence by which users would trust the authority (or third party). Since information is simply brokered by the authority, the only right it has to legitimacy is by virtue of a reputation. Thus expectation 1. above is based, in general, on the rumours that an agent has heard.

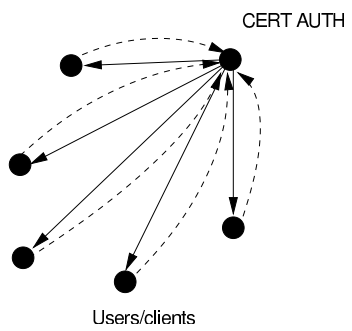

Fig. 5 The Trusted Third Party, e.g. TLS or Kerberos. A special agent is appointed in the network as the custodian of identity. All other agents are expected to trust this. The special agent promises to verify the authenticity of an object that is shared by the agents. In return for this service, the agents pay the special agent.

Most of the trust is from users to the authority, thus there is a clear subordination of agents in this model. This is the nature or centralization.

10.2 Web of Trust

Scepticism in centralized solutions (distrust perhaps) led to the invention of the epidemic trust model, known as the Web of Trust (see fig. 6)[31]. In this model, each individual agent is responsible for its own decisions about trust. Agents confirm their belief in credentials by signing one another's credentials. Hence if I trust A and A has signed B's key then I am more likely to trust B.

As a management approximation, users are asked to make a judgement about a key from one of four categories: i) definitely trustworthy, ii) somewhat trustworthy, iii) untrustworthy, iv) don't know.

An agent then compares these received valuations to a threshold value to decide whether or not a credential is trustworthy to it.

The promises are between the owner of the credential and a random agent:

$$\text{Owner} \xrightarrow{\pi:\text{Identity}} \text{Agent} \tag{77}$$

$$\text{Agent} \xrightarrow{\pi:U(\text{Identity})} \text{Owner} \tag{78}$$

$$\text{Agent} \xrightarrow{\pi:\text{Signature}} \text{Owner} \tag{79}$$

$$\text{Owner} \xrightarrow{\pi:U(\text{Signature})} \text{Agent} \tag{80}$$

The owner must first promise its identity to an agent it meets. The agent must promise to believe and use this identity credential. The agent then promises to support the credential by signing it, which implies a promise (petition) to all subsequent agents. Finally, the owner can promise to use the signature or reject it. Trust enters here in the following ways:

1. The agent expects that the identity of the owner is correct and trusts it. This leads to a Use promise.
2. The Owner expects that the promise of support is legitimate and trusts it. This leads to a Use promise.

What is interesting about this model is that it is much more symmetrical than the centralized scheme. It has certain qualities that remind us of our definition of global trust in section 9. However, it is not equivalent to our model, since the very nature of the web of trust is dictated by the transactions in the model, which are automatically bilateral (ours need not be). Moreover, the information is passed on in a peer to peer way, where as our global idealization makes trust valuations common knowledge (global reputations). In some respects, the web of trust is a pragmatic approximation to the idealized notion of trust in section 9. The main differences are:

- In the Web of trust, a limited number of expectation values is allowed and the user does not control these, i.e. there are few policy choices for agent expectation allowed.
- An agent does not see a complete trust or promise graph. It sees only the local cluster to which it is connected. This is sufficient to compute a global trust for that component of the graph.
- The Web of Trust graph is always bilateral, with arrows moving in both directions, thus no one is untrusted, or un-trusting.
- The information to construct a fully self-consistent measure of trust is not available in the system. Hence there is no clear measure of who is more trustworthy in the web of trust.

Some of these limitations could no doubt be removed. A Bayesian approach could naturally lead to a better approximation. However, a basic flaw in these implementation mechanisms is the need to trust of the mediating software itself. Since, as we have shown, trust is not necessarily transitive, one ends up in most cases trusting the software that is supposed to implement the trust management rather than the parties themselves.

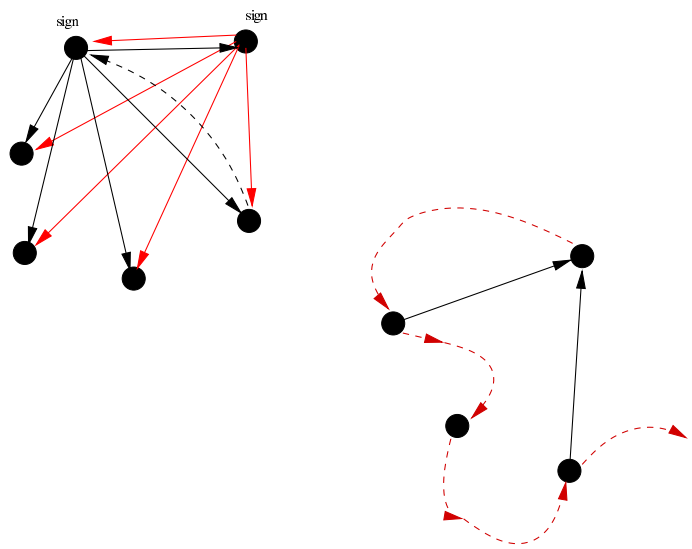

Fig. 6 In a web of trust an agent signals a promise to all other agents that it has trusted the authenticity of the originator's identity. As a key is passed around (second figure) agents can promise its authenticity by signing it or not.

11 Conclusions

The concept of promises provides a foundation that has been unclear in discussions of trust. It allows us to decouple the probabilistic aspect from the network aspect of policy relationships, without introducing instantaneous events. It provides (we claim) a natural language for specific policies, extended over time. Promises have types and denote information flow which in turn allows us to discuss what is trusted and by whom. We believe the use of promises to be superor to a definition based on actions, since the localization of actions as space-time events makes trust ill-defined if the action has either not yet been executed or after it has been executed.

Promises allow us to relate trust and trust-reputation in a generic way, and suggest an algorithm from which to derive global network properties, based on social network theory. This is a significant improvement over previous models. Reputation is not uniquely coupled to trust, of course – it can be related to many different valuations of promised behaviour, including wealth, kindness etc.

We show how bundles of promises can be combined using the rules for probabilistic events (similar to fault tree analysis) and we model the two main trust architectures easily. The PGP Web of Trust as well as the Trusted Third Party can be explained as a special case the global trust models derived here; however standard tools do not permit users to see the entire web, or measure relative trust-worthiness in a community using these implementations.

In future work there is the possibility to use this notion of trust in explicit systems. The Unix configuration system cfengine[32] uses the notion of promises and agent autonomy to implement a policy based management system. The trustworthiness of hosts with respect to certain different behaviours can be measured directly by neighbouring agents to whom promises are made. More generally, if one has a monitoring system that one believes trustworthy to begin with, it is possible to observe whether an agent stops keeping its own promises about security issues. This might be a signal to reevaluate one's expectation that the system is trustworthy. These tests have been partially imeplemented in cfengine and are presently being tested.

Trust is merely an expression of policy and it is therefore fundamentally *ad hoc*. Promises reveal the underlying motives for trust and whether they are rationally or irrationally formed.

Acknowledgement. We are grateful to Jürgen Schönwälder for his hospitality at the International University of Bremen, where part of this work was done. This work is supported in part by the EC IST-EMANICS Network of Excellence (#26854).

References

1. Mark Burgess. An approach to understanding policy based on autonomy and voluntary co-operation. In *IFIP/IEEE 16th international workshop on distributed systems operations and management (DSOM), in LNCS 3775*, pages 97–108, 2005.
2. M. Burgess and S. Fagernes. Pervasive computing management: A model of network policy with local autonomy. *IEEE Transactions on Software Engineering*, page (submitted).
3. M. Burgess and S. Fagernes. Voluntary economic cooperation in policy based management. *IEEE Transactions on Network and Service Management*, page (submitted).
4. M. Burgess and S. Fagernes. Autonomic pervasive computing: A smart mall scenario using promise theory. *Proceedings of the 1st IEEE International Workshop on Modelling Autonomic Communications Environments (MACE); Multicon verlag 2006. ISBN 3-930736-05-5*, pages 133–160, 2006.
5. L. LaPadula. A rule-set approach to formal modelling of a trusted computer system. *Computing systems (University of California Press: Berkeley, CA)*, **7**:113, 1994.
6. M.D. McIlroy. Virology 101. *Computing systems (University of California Press: Berkeley, CA)*, **2**:173, 1989.
7. I.S. Winkler. The non-technical threat to computing systems. *Computing systems (MIT Press: Cambridge MA)*, **9**:3, 1996.
8. M. Patton and A. Jøsang. Technologies for trust in electronic commerce. *Electronic Commerce Research Journal*, 4:9–21, 2004.
9. Audun Jøsang, Claudia Keser, and Theo Dimitrakos. Can we manage trust? In *Proceedings of the Third International Conference on Trust Management (iTrust), Versailes*, 2005.
10. T. Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. Developing an integrated trust and reputation model for open multi-agent systems. In Rino Falcone, Suzanne Barber, Jordi Sabater, and Munindar Singh, editors, *AAMAS-04 Workshop on Trust in Agent Societies*, 2004.
11. J. Klüwer and A. Waaler. Trustworthiness by default, 2005.
12. J. Klüwer and A. Waaler. Relative trustworthiness. In *Formal Aspects in Security and Trust: Third International Workshop, FAST 2005, Newcastle upon Tyne, UK, July 18-19, 2005, Revised Selected Papers, Springer Lecture Notes in Computer Science 3866*, pages 158–170, 2006.
13. T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS), LNCS*, volume 875, pages 3–18. Springer, 1994.
14. Audun Jøsang and Simon Pope. Semantic constraints for trust transitivity. In *APCCM '05: Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling*, pages 59–68, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.
15. D. Fahrenholtz and A. Bartelt. Towards a sociological view of trust in computer science. In *Proceedings of the Eighth Research Symposium on Emerging Electronic Markets (RSEEM 01)*, page 10, 2001.
16. R. Axelrod. *The Evolution of Co-operation*. Penguin Books, 1990 (1984).
17. A. Høyland and M. Rausand. *System Reliability Theory: Models and Statistical Methods*. J. Wiley & Sons, New York, 1994.
18. G.R. Grimmett and D.R. Stirzaker. *Probability and random processes (3rd edition)*. Oxford scientific publications, Oxford, 2001.
19. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgen Kaufmann, San Francisco, 1988.
20. M. Burgess. *Analytical Network and System Administration — Managing Human-Computer Systems*. J. Wiley & Sons, Chichester, 2004.
21. *The Strategy of Conflict*. Harvard Univesity Press, Cambridge, Mass., 1960.
22. M. Burgess, G. Canright, and K. Engø. A graph theoretical model of computer security: from file access to social engineering. *International Journal of Information Security*, 3:70–85, 2004.
23. P. Bonacich. Power and centrality: a family of measures. *American Journal of Sociology*, 92:1170–1182, 1987.

24. G. Canright and K. Engø-Monsen. A natural definition of clusters and roles in undirected graphs. *Science of Computer Programming*, 53:195, 2004.
25. M. Burgess, G. Canright, and K. Engø. Inportance-ranking functions from the eigenvectors of directed graphs. *Journal of the ACM (Submitted)*, 2004.
26. J.V. Neumann and O. Morgenstern. *Theory of games and economic behaviour*. Princeton University Press, Princeton, 1944.
27. A. Rapoport. *N-Person Game Theory: Concepts and Applications*. Dover, New York, 1970.
28. ITU-T. *Open Systems Interconnection - The Directory: Overview of Concepts, models and service. Recommendation X.500.* International Telecommunications Union, Geneva, 1993.
29. ITU-T Recommendation. X.509 (1997 e): Information technology - open systems interconnection - the directory: Authentication framework. Technical report, 1997.
30. R. Housley, W. Polk, W. Ford, and D. Solo. Internet x.509 public key infrastructure: Certificate and certificate revocation list (crl) profile. http://tools.ietf.org/html/rfc3280, 2002.
31. A. Abdul-Rahman. The pgp trust model. *EDI-Forum: the Journal of Electronic Commerce*, 1997.
32. M. Burgess. Cfengine www site. *http://www.cfengine.org*, 1993.

Electronic Reports Series of the Programming Research Group

Within this series the following reports appeared.

[PRG0605] J.A. Bergstra and J.V. Tucker, *Division Safe Calculation in Totalised Fields,* Programming Research Group - University of Amsterdam, 2006.

[PRG0604] J.A. Bergstra and A. Ponse, *Projection Semantics for Rigid Loops,* Programming Research Group - University of Amsterdam, 2006.

[PRG0603] J.A. Bergstra and I. Bethke, *Predictable and Reliable Program Code: Virtual Machine-based Projection Semantics (submitted for inclusion in the Handbook of Network and Systems Administration),* Programming Research Group - University of Amsterdam, 2006.

[PRG0602] J.A. Bergstra and A. Ponse, *Program Algebra with Repeat Instruction,* Programming Research Group - University of Amsterdam, 2006.

[PRG0601] J.A. Bergstra and A. Ponse, *Interface Groups for Analytic Execution Architectures,* Programming Research Group - University of Amsterdam, 2006.

[PRG0505] B. Diertens, *Software (Re-)Engineering with PSF,* Programming Research Group - University of Amsterdam, 2005.

[PRG0504] P.H. Rodenburg, *Piecewise Initial Algebra Semantics,* Programming Research Group - University of Amsterdam, 2005.

[PRG0503] T.D. Vu, *Metric Denotational Semantics for BPPA,* Programming Research Group - University of Amsterdam, 2005.

[PRG0502] J.A. Bergstra, I. Bethke, and A. Ponse, *Decision Problems for Pushdown Threads,* Programming Research Group - University of Amsterdam, 2005.

[PRG0501] J.A. Bergstra and A. Ponse, *A Bypass of Cohen's Impossibility Result,* Programming Research Group - University of Amsterdam, 2005.

[PRG0405] J.A. Bergstra and I. Bethke, *An Upper Bound for the Equational Specification of Finite State Services,* Programming Research Group - University of Amsterdam, 2004.

[PRG0404] J.A. Bergstra and C.A. Middelburg, *Thread Algebra for Strategic Interleaving,* Programming Research Group - University of Amsterdam, 2004.

[PRG0403] B. Diertens, *A Compiler-projection from PGLEc.MSPio to Parrot,* Programming Research Group - University of Amsterdam, 2004.

[PRG0402] J.A. Bergstra and I. Bethke, *Linear Projective Program Syntax,* Programming Research Group - University of Amsterdam, 2004.

[PRG0401] B. Diertens, *Molecular Scripting Primitives,* Programming Research Group - University of Amsterdam, 2004.

[PRG0302] B. Diertens, *A Toolset for PGA,* Programming Research Group - University of Amsterdam, 2003.

[PRG0301] J.A. Bergstra and P. Walters, *Projection Semantics for Multi-File Programs,* Programming Research Group - University of Amsterdam, 2003.

[PRG0201] I. Bethke and P. Walters, *Molecule-oriented Java Programs for Cyclic Sequences,* Programming Research Group - University of Amsterdam, 2002.

The above reports and more are available through the website: www.science.uva.nl/research/prog/

Electronic Report Series

Programming Research Group
Faculty of Science
University of Amsterdam

Kruislaan 403
1098 SJ Amsterdam
the Netherlands

www.science.uva.nl/research/prog/