

# A Formal Framework for Modeling Trust and Reputation in Collective Adaptive Systems

Alessandro Aldini

Dipartimento di Scienze Pure e Applicate, Università di Urbino, Urbino, Italy

alessandro.aldini@uniurb.it

Trust and reputation models for distributed, collaborative systems have been studied and applied in several domains, in order to stimulate cooperation while preventing selfish and malicious behaviors. Nonetheless, such models have received less attention in the process of specifying and analyzing formally the functionalities of the systems mentioned above. The objective of this paper is to define a process algebraic framework for the modeling of systems that use (i) trust and reputation to govern the interactions among nodes, and (ii) communication models characterized by a high level of adaptiveness and flexibility. Hence, we propose a formalism for verifying, through model checking techniques, the robustness of these systems with respect to the typical attacks conducted against webs of trust.

## 1 Introduction

Trust and reputation management systems [13] can improve the reliability of the interactions and the attitude to cooperation for several types of collaborative systems, in various different domains, such as participatory sensing systems, wireless sensor networks, peer-to-peer services, mobile ad-hoc networks, user-centric networks, supply networks, and, last but not least, collective adaptive systems. Typically, the models proposed for these systems rely on distributed notions of trust and reputation. More precisely, trust management is distributed over all the nodes, which may collaborate with each others in order to exchange and aggregate personal opinions, calculate trust scores of target nodes, and disseminate such values [24, 28, 8, 21]. For instance, trustworthy sensor networks base their ability to collectively process sensed data on decentralized reputation systems [9, 29, 11, 26, 22]. Nodes monitoring the behavior of neighbor nodes in the network maintain reputation for such nodes. Hence, collaboration among nodes with high reputation can be strengthened while malicious nodes are excluded from the community, thus favoring activities like, e.g., intrusion detection, participatory sensing, and many more.

A web of trust can be established according to a geographical notion of *group* of nodes, as in crowd-sourcing and sensor networks [9], or by following community based models, as in social networks and P2P environments [30]. Trust derives from local, direct observations, e.g., through watchdog mechanisms, quantitatively represented by scores assigned to rate the result of interactions, and from second hand information, represented by recommendations provided to a node by the other nodes of its web of trust. All these values are combined by the specific trust system to derive, e.g., a computational notion of trust, which is then used as a belief level to predict either statistically or deterministically the future behavior of the various network members.

**Example 1.1** In several trust models [5, 23, 30], the trust value of peer *A* towards peer *C* through peer *B* is expressed by a formula of the form:

$$1 - (1 - t_{BC})^{t_{AB}}$$

where  $t_{IJ}$  is the trust from  $I$  to  $J$ . Hence,  $t_{BC}$  plays the role of a recommendation given to  $A$ , which is weighted by the direct trust from  $A$  to  $B$ . Inspired by this model, in [30] a notion of *club* is used to aggregate multiple self-organizing peers with common needs/features in order to improve the efficiency of service discovery/delivery in peer-to-peer collaborative networks. Each club includes a special node, called CDSR, with management tasks. Then, trust is generalized to express relations among clubs. For instance, the trust from club  $X$  to club  $Y$ , reporting the result of direct experiences among peers belonging to the two clubs, depends on the amount of positive experiences  $p$  and negative experiences  $n$  observed by peers in  $X$  when interacting with peers in  $Y$ :

$$t_{XY}(p, n) = \begin{cases} 1 - \lambda^{p-n} & \text{if } p > n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where the configuration parameter  $\lambda$  is the probability of reliability with a single interaction. Instead, the reputation of peer  $K \in Y$  as perceived by the other peers of  $Y$  is non-zero only if all the interactions of such peers with  $K$  are positive and depends on the amount  $p$  of these direct experiences:

$$t_{YK} = 1 - \lambda^p. \quad (2)$$

By combining these trust values, we obtain the trust of any peer in club  $X$  towards peer  $K$  belonging to another club  $Y$ :

$$t_{XK} = 1 - (1 - t_{YK})^{t_{XY}}.$$

**Example 1.2** In reputation-based sensor networks [9, 18], the local, direct trust from node  $I$  to node  $J$  is maintained by using a watchdog mechanism in  $I$  reporting the result of each direct experience with  $J$ . Such a feedback, which may consist of scores or, more simply, the amount of good behaviors and of mis-behaviors observed, is then used to parameterize a trust formula relying on a standard Bayesian approach. The calculated trust value thus represents the expectation estimating the belief level that one node has on another node for a specific action. Second hand information can be asked from neighbor nodes, in the form of recommended trust values reported by such nodes and scaled by a factor proportional to the trust towards such recommending nodes.

**Example 1.3** EigenTrust [14] is a trust system originally proposed for P2P file sharing systems. Peers rate with value 1 (resp.,  $-1$ ) each satisfactory (resp., unsatisfactory) interaction. The local trust  $s_{ij}$  from  $i$  to  $j$  is computed by summing up the scores of the individual transactions conducted by peer  $i$  with peer  $j$ . Then,  $s_{ij}$  is normalized with respect to  $\sum_j s_{ij}$  in such a way to obtain a trust value  $c_{ij}$  between 0 and 1, with  $\sum_j c_{ij} = 1$ . These trust values are then aggregated to form a distributed notion of reputation. The principle behind the computation of the global trust  $t_{ij}$  from  $i$  to  $j$  is to combine the opinions of  $i$ 's neighbors, as follows:

$$\sum_k c_{ik} c_{kj}$$

In matrix notation, given  $C$  the matrix  $[c_{ij}]$  of all the trust values and  $c_i$  the vector containing the values  $c_{ij}$ , then the vector  $t_i$  of the values  $t_{ij}$  is computed as  $C^T \cdot c_i$ . Such a mechanism can be iterated by aggregating the opinions of communities in cascade, i.e., by computing  $(C^T)^n \cdot c_i$ . For  $n$  large enough, the result converges to the same trust vector for every peer  $i$  in the network, which thus represents the vector of global trust values.

In PeerTrust [27], developed for distributed systems, trust towards a peer  $i$  depends on the amount of known interactions between  $i$  and other peers, the known feedback reported by such peers, the credibility of such peers, and an adaptive community context factor for peer  $i$ . In turn, credibility of a peer  $j$  from

the viewpoint of a peer  $k$  depends on the recommendations about  $j$  provided by peers that previously interacted with both  $k$  and  $j$ .

In all these examples, the trust-based selection is based on the rule  $t_{ij} \geq th_i$ , where the trust threshold value  $th_i$  may depend on several factors influencing  $i$ , such as the dispositional trust of  $i$ , which represents the initial willingness of the peer  $i$  to cooperate with unknown peers.

Systems such as those mentioned above are typically verified through simulation [15, 9, 30, 14] or game theory [19], possibly leading to results validating the trust model against attacks like, e.g.:

- bad mouthing: negative feedback reported by an adversary about the behavior of a trusted agent;
- ballot stuffing: positive feedback reported by an adversary about the behavior of a malicious agent;
- collusion: attack conducted by multiple adversaries which act together with the aim of damaging a honest agent;
- on-off: attack conducted by an adversary alternating between normal behaviors and misbehaviors.
- sybil: attack conducted by an adversary generating multiple identities with the aim of flooding the system with fake information or misbehaviors.
- white-washing: attack conducted by a misbehaving adversary who leaves the system whenever her reputation is compromised and then rejoins it using a different identity.

However, the lack of formal validation can be seen as a weakness, especially in such a complex framework in which attacks and countermeasures depend on the flexibility and on the dynamic behavior of the web of trust [29, 20]. Classical verification techniques, like model checking, have demonstrated their adequacy in the validation process of systems with respect to properties like safety, reliability, security, and performance. On the other hand, they have not received the same attention in the setting of trust and reputation (see, e.g., [1] and the references therein). To cite few representative examples in the setting of model checking based analysis, Reith et al. [25] verify delegation mechanisms in access control, which can be viewed as a form of trust management, while He et al. [12] apply the same approach to the verification of chains of trust. Finally, in [3, 16] the PRISM model checker is used to estimate the tradeoff between trust-based incentives and remuneration-based incentives in cooperative user-centric networks.

In this paper, which is inspired by [1], we present a process algebraic framework for the modeling and, therefore, analysis of trust-based adaptive systems. With respect to [1], the proposed framework offers different ways of modeling trust and trust-based choices, and introduces mobility and collaboration aspects affecting the establishment and management of dynamic and adaptive webs of trust. To this aim, a notion of environment is modeled explicitly that guides the communication and, as a consequence, the trust relationships, among dynamic agents. Historically, starting with the Ambient Calculus [7], and until the most recent proposals [6, 17], several process calculi have been defined that represent mobile computation with a notion of environment. With respect to such proposals, the contribution of this paper is a dynamic communication model relying on trust relationships.

The rest of the paper is organized as follows. In Section 2, we present the formal framework for the description of an agent-based network of trust. We first define a basic calculus of sequential processes and then we show how to model communications based on trust relations. Then, in Section 3 we show the adequacy of such a framework by presenting two real-world examples. In Section 4, we briefly discuss how to model check trust-based properties and, finally, in Section 5 we comment on future directions for the proposed approach.

## 2 Modeling an agent-based web of trust

All the examples shown in the previous section emphasize that the ingredients needed to feed a trust model for distributed, adaptive systems are:

1. the set of direct experiences affecting a local notion of trust. A direct experience is expressed quantitatively by a positive/negative score assigned to evaluate an interaction.
2. the set of groups of agents collaborating, e.g., through the exchange of recommendations, in order to calculate a global notion of trust. It is worth observing that the composition of such groups may be characterized by high levels of flexibility.

It is worth observing that in the following we abstract from the way in which the basic parameters concerned with local and global notions of trust are combined to compute opinions governing the decision making process, which is a task specific of the trust model adopted. Instead, we concentrate on the specification of the behavior of agents and on the establishment of their networks of trust. For this purpose, as we will see, in the semantics of our formal specification language we have rules describing (i) how the basic parameters needed by the trust system are calculated and maintained, and (ii) how the results computed by the trust system, i.e., the  $t_{IJ}$  values, are then used to govern the trust-based interactions. All the machinery taking in input the basic parameters mentioned above and returning as output the trust values is hidden and left to the specification of the trust model.

Moreover, to simplify the presentation, unless differently specified we restrict our consideration to systems in which one type of service is provided within the network. In order to generalize, it is sufficient to replicate as many instances of the trust infrastructure as the number of different services modeled in the system, because trust-based beliefs are specific to the required service.

### 2.1 Basic Calculus

We denote with *Name* the set of visible action names, ranged over by  $a, b, \dots$ , and we assume that  $Name = Name_o \cup Name_i$ , where  $Name_o$  and  $Name_i$  are disjoint and represent the sets of output actions and input actions, respectively. The fresh name  $\tau$  is used to represent invisible, internal actions. We also use  $\alpha, \dots$  to express visible and internal actions.

The set of terms of the basic calculus for sequential processes is generated through the following syntax:

$$P ::= \underline{0} \mid \alpha.P \mid P+P \mid B$$

where we have the constant  $\underline{0}$  denoting the inactive process, the classical algebraic operators for prefix and nondeterministic choice, and a constant based mechanism for expressing recursive processes. As usual, we consider only guarded and closed process terms.

Then, the semantics of process terms is expressed in terms of labeled transition systems.

**Definition 2.1** A labeled transition system (LTS) is a tuple  $(Q, q_0, L, R)$ , where  $Q$  is a finite set of states (with  $q_0$  the initial one),  $L$  is a finite set of labels, and  $R \subseteq Q \times L \times Q$  is a finitely-branching transition relation.

In the following,  $(q, l, q') \in R$  is denoted by  $q \xrightarrow{l} q'$ . Then, the behavior of process term  $P$  is defined formally by the smallest LTS  $(Q, q_0, L, R)$  such that  $Q$  is the set of process terms of our basic calculus (with  $P$  representing the initial state  $q_0$ ),  $L = \{\tau\} \cup Name$ , and the transitions in  $R$  are obtained through the application of the operational semantics rules of Table 1. The semantics of process term  $P$  is denoted by  $\llbracket P \rrbracket$ .

Table 1: Semantics rules of the basic calculus.

	<i>prefix</i> $\alpha.P \xrightarrow{\alpha} P$
<i>choice</i>	$\frac{P_1 \xrightarrow{\alpha} P'_1}{P_1 + P_2 \xrightarrow{\alpha} P'_1} \quad \frac{P_2 \xrightarrow{\alpha} P'_2}{P_1 + P_2 \xrightarrow{\alpha} P'_2}$
<i>recursion</i>	$B \stackrel{\text{def}}{=} P \quad \frac{P \xrightarrow{\alpha} P'}{B \xrightarrow{\alpha} P'}$

## 2.2 Interacting agents

When passing to concurrent processes, we deal with process term instances, called agents, which represent elements exhibiting the behavior associated to a given process term. This separation of concerns between the definition of agents and of their behavioral pattern is inspired by process algebraic architectural description languages (see, e.g., [2] and the references therein). The kernel of the semantics of an agent  $I$  belonging to the behavioral pattern defined by process term  $P$  is obtained from  $P$  by replacing each action  $\alpha$  of  $P$  with  $I.\alpha$ . Hence, the semantics  $\llbracket I \rrbracket$  of agent  $I$  derives from  $\llbracket P \rrbracket$  in the same way. Then, we say that  $I$  is of type  $P$ , denoted  $I : P$ , and with the notation  $I.B$  we express that the local behavior of  $I$  is given by the process term identified by the constant  $B$ . In the following,  $\mathcal{S}$  denotes a finite set of agents  $\{I_i : P_i \mid 1 \leq i \leq n\}$  such that each agent name  $I_i$  is unique.

For notational convenience, from now on,  $P, P' \dots$  represent the kernel of the semantics of agents, hence  $P \xrightarrow{I.\alpha} P'$  denotes a transition performed by agent  $I$  from its current local state represented by process term  $P$  to the new local state represented by process term  $P'$ . Given a set  $\mathcal{S}$  of agents forming a system, a vector of processes expressing the local state of each agent in  $\mathcal{S}$  represents the global state of the system, ranged over by  $\mathcal{P}, \mathcal{P}', \dots$ . Moreover,  $\mathcal{P}[P'/P]$  represents the substitution of  $P$  with  $P'$  in  $\mathcal{P}$ . Such a notation is not ambiguous as  $P, P'$  express the kernel of the semantics of a uniquely identified agent in  $\mathcal{S}$ .

As we will see, the interacting semantics of  $\mathcal{S}$  is given by the parallel composition of its constituting agents, the interactions among which are regulated by communication rules that depend on community membership and trust information. In particular, the communication model is based on the following structures:

- A synchronization set  $S \subseteq \text{Name}_o \times \text{Name}_i$ , containing pairs of actions denoted syntactically by  $a \times b$ . Action  $a$  represents the output, governing counterpart of the synchronous communication, while action  $b$  denotes the input, reacting counterpart. Hence, we assume that synchronous communication is asymmetric, in the sense that one of the two agents involved governs it while the other one reacts.
- A set of groups of agents (also said set of communities)  $\mathcal{G} \subseteq 2^{\mathcal{S}}$ , such that each group represents a set of agents that can communicate directly with each other and can share trust opinions. As we will see, synchronous communication is possible only within the same group, while group membership is dynamic.
- A multiset of trust opinions  $\mathcal{E}$  with support set of type  $(\mathcal{S}, T \cup \{?\})_{\mathcal{S}}$ , where  $T$  is a totally ordered trust domain. Element  $(J, v)_I$  expresses that after a communication between  $I$  and  $J$ , agent  $I$  has

rated the interaction by assigning the score  $v$  (the special symbol  $?$  means that an occurred interaction has not been rated yet). We observe that  $\mathcal{E}$  is a multiset, as agent  $I$  may be involved in several different interactions with agent  $J$ , and some of them could be rated with the same score. As we will see, trust opinions feed the trust system in order to compute the trust values  $t_{IJ}$ , which in turn govern potential synchronous communications from  $I$  to  $J$ .

Intuitively, a trust adaptive system is a set of interacting agents obeying the communication model described above. Therefore, formally, a trust adaptive system is a tuple made of a set of agents  $\mathcal{S}$ , a synchronization set  $S$ , a dynamic set of communities  $\mathcal{G}$ , and a dynamic multiset of trust opinions  $\mathcal{E}$  (another parameter, i.e., the trust model, is implicit). The evolution of a trust adaptive system is described by the semantics rules of Table 2, which formalize the parallel composition of the agents forming the system. More precisely, these rules define the moves (deriving from autonomous actions and synchronous communications) from configurations to configurations, where a configuration is defined by the global state of the system, the synchronization set, the current set of interacting communities, and the current multiset of trust opinions. Let us explain intuitively such rules.

Table 2: Semantics rules for parallel composition.

$\frac{P \in \mathcal{P} \quad P \xrightarrow{I.\tau} P'}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.\tau} (\mathcal{P}[P'/P], S, \mathcal{G}, \mathcal{E})}$
$\frac{P \in \mathcal{P} \quad G \in \mathcal{G} \quad P \xrightarrow{I.ent(G)} P'}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.\tau} (\mathcal{P}[P'/P], S, \mathcal{G}[G \cup \{I\}/G], \mathcal{E})}$
$\frac{P \in \mathcal{P} \quad G \in \mathcal{G} \wedge I \in G \quad P \xrightarrow{I.esc(G)} P'}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.\tau} (\mathcal{P}[P'/P], S, \mathcal{G}[G \setminus \{I\}/G], \mathcal{E})}$
$\frac{P_1, P_2 \in \mathcal{P}, P_1 \neq P_2 \quad a \times b \in S \quad G \in \mathcal{G} \wedge I, J \in G \quad P_1 \xrightarrow{I.a} P'_1 \quad P_2 \xrightarrow{J.b} P'_2 \quad a \in H \wedge t_{IJ} \geq th_I}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.a \times J.b} (\mathcal{P}[P'_1/P_1, P'_2/P_2], S, \mathcal{G}, \mathcal{E} \cup \{(J, ?)_I\} \cup \{(I, ?)_J\})}$
$\frac{P_1, P_2 \in \mathcal{P}, P_1 \neq P_2 \quad a \times b \in S \quad G \in \mathcal{G} \wedge I, J \in G \quad P_1 \xrightarrow{I.a} P'_1 \quad P_2 \xrightarrow{J.b} P'_2 \quad a \in L \wedge t_{IJ} < th_I}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.a \times J.b} (\mathcal{P}[P'_1/P_1, P'_2/P_2], S, \mathcal{G}, \mathcal{E} \cup \{(J, ?)_I\} \cup \{(I, ?)_J\})}$
$\frac{P_1, P_2 \in \mathcal{P}, P_1 \neq P_2 \quad a \times b \in S \quad G \in \mathcal{G} \wedge I, J \in G \quad P_1 \xrightarrow{I.a} P'_1 \quad P_2 \xrightarrow{J.b} P'_2 \quad a \notin \{H \cup L\}}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.a \times J.b} (\mathcal{P}[P'_1/P_1, P'_2/P_2], S, \mathcal{G}, \mathcal{E})}$
$\frac{P \in \mathcal{P} \quad G \in \mathcal{G} \wedge I, J \in G \quad (J, ?)_I \in \mathcal{E} \quad P \xrightarrow{I.obs(v)} P'}{(\mathcal{P}, S, \mathcal{G}, \mathcal{E}) \xrightarrow{I.\tau} (\mathcal{P}[P'/P], S, \mathcal{G}, \mathcal{E} \setminus \{(J, ?)_I\} \uplus \{(J, v)_I\})}$

The first rule refers to the internal action  $\tau$ , which is performed autonomously by each agent. Then, we have two additional, internal actions that can be performed autonomously by each agent, which we add to the syntax of the basic calculus:

$$ent(G) \mid esc(G)$$

where  $G \in \mathcal{G}$ . Such actions concern the membership to communities. In particular, action  $ent(G)$  allows an agent to join the group  $G$  of agents (notice that  $G$  is replaced by  $G \cup \{I\}$ , where  $I$  is the agent joining the group). Action  $esc(G)$  allows an agent to leave the group  $G$  of agents (notice that  $G$  is replaced by  $G \setminus \{I\}$ , where  $I$  is the agent leaving the group). We point out that groups are used to dynamically confine the sets of agents that can interact directly through synchronous communication and within which trust based information can be shared. Hence, such sets represent the communities referenced by an agent in a given instant of time in order to obtain trust recommendations.

The following three rules formalize the trust-based synchronous communication between two different agents. Based on the communication model previously described, an interaction from  $I$ , offering output  $a$ , to  $J$ , reacting with input  $b$ , is possible if two conditions hold:

- $a \times b$  belongs to the synchronization set  $S$ ;
- there exists a community of which both  $I$  and  $J$  are members.

Moreover, the communication from  $I$  to  $J$  may depend on the trust of  $I$  towards  $J$ . Inspired by the noninterference approach to information flow analysis [10], all the actions involved in trust-based communications are classified into two disjoint sets,  $H$  and  $L$ , denoting high-level and low-level actions, such that:

- $(H \cup L) \subseteq Name$ ;
- for each  $a \times b \in S$  it holds that  $a \in H$  if and only if  $b \in H$  and  $a \in L$  if and only if  $b \in L$ .

If agent  $I$  offers output  $a \in H$ , then the potential reacting counterpart must satisfy the trust-based selection policy based on the trust threshold  $th_I$ . A typical high-level action is the service request sent by an agent  $I$  to another agent  $J$ , which is chosen as a trusted partner. Notice that since the communication model is asymmetric, then the trust-based condition is applied only by the agent offering the output action, which governs the interaction. On the contrary, if agent  $I$  offers output  $a \in L$ , then an interaction through  $a$  is possible only if the trust-based selection policy based on the trust threshold  $th_I$  is not satisfied by the counterpart. A typical low-level action is the denial of service delivery that is sent by an agent  $I$  to another agent  $J$ , who previously sent a service request to  $I$  that cannot be accepted as  $J$  is not trusted enough by  $I$ . If  $a \notin \{H \cup L\}$ , then every interaction involving  $a$  does not rely on trust-based requirements. The trust-based selection policy enabling a trusted interaction from  $I$  to  $J$  is  $t_{IJ} \geq th_I$ , where  $t_{IJ}$  is the trust of  $I$  towards  $J$  as estimated by the trust model, which relies on the set of basic parameters collected during the system execution. Hence, its calculation strictly depends on the chosen trust model and does not affect the definition of the semantics for interacting processes. As discussed,  $t_{IJ}$  may be based on several different methods [14, 31, 30], an example of which will be given in the following. Whenever a trust-based communication occurs, then a feedback, in the form of a score  $v$ , could be provided by each of the two parties to rate the level of satisfaction in the interaction with the other party. To keep track of such a possibility, terms  $(J, ?)_I$  and  $(I, ?)_J$  are added to the set  $\mathcal{E}$  of local opinions. The former denotes that  $I$  can rate an interaction with  $J$ , and vice versa for the latter. This evaluation may occur later on during system execution. Hence, to report the feedback, we add to the syntax of the basic calculus the special internal action  $obs(v)$ , where  $v \in \mathbb{T}$ , which allows the agent executing it to rate a trust-based interaction previously conducted with a known agent, see the last semantic rule. Notice that the effect of such an action is to replace the symbol  $?$  in  $(J, ?)_I$  with the score  $v$ .

**Well-formedness.** The placeholder  $(J, ?)_I$  is added to the multiset  $\mathcal{E}$  through the union operator  $\cup$ <sup>1</sup>. As a consequence, it can occur in  $\mathcal{E}$  with multiplicity 1 at most. A score assigned to an interaction

<sup>1</sup>Multiset union is defined as the multiset such that each element has the maximal multiplicity it has in either multisets.

between  $I$  and  $J$  refers to the last of the unrated interactions among them. If other, older, unrated interactions among them exist, they lose the possibility to be rated. In this way, we can model the situation in which no feedback is reported, either because it is not needed or when the user is not stimulated to provide trust rates. Whenever the placeholder  $(J, ?)_I$  is removed, an element of the form  $(J, v)_I$  is added to  $\mathcal{E}$  through the multiset sum operator  $\uplus$ <sup>2</sup>, meaning that such an element may occur in  $\mathcal{E}$  with multiplicity greater than 1. Notice that, if two different placeholders  $(J, ?)_I$  and  $(J', ?)_I$  occur in  $\mathcal{E}$ , then the execution of transition  $I.obs(v)$  assigns score  $v$  either to  $J$  or to  $J'$ , nondeterministically. Such a situation is avoided if the feedback is reported before the execution of a new interaction with another agent, as typical in most trust-based systems, in which case we say that the system is well-defined.

As far as the feedback mechanism is concerned, the last rule of the semantics expresses the correct behavior of an agent rating a real interaction, as expected by any trust system. However, such an assumption is a limitation with respect to the modeling of malicious behaviors, which would require an improper use of the action  $obs$ . With the aim of modeling fake trust reports and, therefore, false recommendations, we add a new special internal action and the following rule for pushing fictitious opinions:

$$\frac{P \in \mathcal{P} \quad G \in \mathcal{G} \wedge I, J \in G \quad P \xrightarrow{I.fake\_obs(J,v)} P'}{(\mathcal{P}, \mathcal{S}, \mathcal{G}, \mathcal{E}) \xrightarrow{I.\tau} (\mathcal{P}[P'/P], \mathcal{S}, \mathcal{G}, \mathcal{E} \uplus \{(J, v)_I\})}$$

which allows any agent to rate the other agents of the community without any restriction. Going back to the list of attacks discussed in Section 1, we observe that they can be modeled by using actions  $obs(v)$  and  $fake\_obs(J, v)$ . Moreover, the adaptive community-based communication policy is useful to model sybil and white-washing attacks.

The formal semantics of interacting agents is expressed in terms of an extension of LTSs.

**Definition 2.2** Given a set of trust predicates  $\mathbb{T}\mathbb{P}$  and a set of names  $\mathbb{N}$ , a trust labeled transition system (TLTS) is a tuple  $(Q, q_0, L, R, T)$  where:

- $(Q, q_0, L, R)$  is a LTS.
- $T : Q \rightarrow 2^{\mathbb{T}\mathbb{P}} \times 2^{\mathbb{N}}$  is a labeling function.

In our framework,  $\mathbb{T}\mathbb{P}$  is of the same type as  $\mathcal{E}$ , while  $\mathbb{N}$  is the set of agent names. Then, the semantics of a trust adaptive system described by the tuple  $(\mathcal{S}, \mathcal{S}, \mathcal{G}, \mathcal{E})$ , where  $\mathcal{S}$  contains agents  $I_i$ ,  $1 \leq i \leq n$ ,  $\mathcal{S}$  is the synchronization set,  $\mathcal{G}$  is the initial set of communities, and  $\mathcal{E}$  is the initial multiset of trust opinions, is the smallest TLTS satisfying the following conditions:

- Each global state  $q \in Q$  is a  $n$ -length vector of process terms modeling the local behavior of the agents  $I_i$ ,  $1 \leq i \leq n$ , such that the initial global state  $q_0$  is associated to the vector modeling the initial local state of each agent.
- $L = \{I.\tau \mid I \in \mathcal{S}\} \cup \{I.a \times J.b \mid I, J \in \mathcal{S} \wedge a \times b \in \mathcal{S}\}$ .
- The transitions in  $R$  and the labelings of  $T$  are obtained through the application of the operational semantics rules of Table 2, with the labels of  $q_0$  determined by  $\mathcal{G}$  and  $\mathcal{E}$ .

Typically,  $\mathcal{E} = \emptyset$  in  $q_0$ . The assumption concerning the emptiness of  $\mathcal{E}$  in the initial state can be changed according to the trust model. In some case (see, e.g., [14]), in fact, a priori estimations of trust are assigned to agents that are known to be trustworthy in a community, e.g., as they are among the founders of the community. Hence, pre-trusted agents can be modeled by setting adequately  $\mathcal{E}$  in the initial state.

<sup>2</sup>Multiset sum is defined as the multiset such that each element has the sum of the multiplicities it has in both multisets.



### 3 Two examples

In this section, we sketch the formal modeling of two real-world systems using the trust models of [30] and [14], in which local trust deriving from direct experiences is calculated by counting the number of positive and negative experiences. Hence, it is sufficient to assume that the feedback reported through action *obs* is either 1 or  $-1$ , respectively, thus implying  $T = \mathbb{Z}$ .

First, let us consider a system using the trust model proposed in [30]. In the following, we illustrate the main aspects related to the computation of the trust value  $t_{IJ}$  without going into the details of the algebraic specifications expressing the agents behavior. The system includes behavioral patterns for the following categories: nodes consuming services (type *Cons*), nodes delivering services (type *Prod*), and nodes governing clubs (type *CDSR*). Each club is defined as a group including one agent of type *CDSR*, some consumer, and several producers offering the service that characterizes the club. For instance, given two fixed clubs  $G_1$  and  $G_2$ , the process term *Cons* could be defined as follows (the summation symbol  $\Sigma$  is used to generalize the choice operator):

$$Cons \stackrel{\text{def}}{=} \sum_{i \in \{1,2\}} \tau.send\_request_i. \left( \sum_{\{j \in G_i\}} receive\_service_{ij}.(obs(1).Cons + obs(-1).Cons) + receive\_denial_{ij}.obs(-1).Cons \right)$$

where output  $send\_request_i \in H$ , input  $receive\_service_{ij} \in H$ , and input  $receive\_denial_{ij} \in L$ . We assume that the synchronization set enables a communication through  $send\_request_i$  and a corresponding input, say  $receive\_request_i$ , which is offered by every producer  $j$  belonging to  $G_i$ . Notice that the choice of the specific producer  $j$  is nondeterministic among the agents trusted by the consumer, which proposes the request to all the agents of group  $G_i$ . Such an interaction is not rated by the consumer. Afterwards, through adequate synchronizations between the consumer and the responding producer, either the consumer receives the service, and then rates the interaction nondeterministically, or the producer refuses the request, and in such a case the consumer rates negatively the failure. The choice between the two events is deterministic and based on the trust of the chosen producer towards the consumer. We point out that the feedback, reported through action *obs*, is assigned to the unique producer interacting with the consumer in a fully transparent way by virtue of the semantics rules of Table 2.

All the interactions governed by trust are based on the following encoding of the trust model of [30]. Given agent  $k$  in the club  $Y$ , Equation 2 is estimated by setting parameters  $p$  and  $n$  as follows:

$$p = \sum_{j \in Y, j \neq k} mul((k, 1)_j)$$

where  $mul(e)$  denotes the multiplicity of term  $e$  in  $\mathcal{E}$ . The estimation of parameter  $n$  is analogous by replacing 1 with  $-1$  in the definition above. On the other hand, given clubs  $X$  and  $Y$ , Equation 1 is estimated as follows:

$$p = \sum_{i \in X, j \in Y} mul((j, 1)_i)$$

and similarly in the case of parameter  $n$ . Given such a model, any trust-based communication enabled in a global state  $q$  of the TLTS representing the current system behavior, depends on the labeling  $T(q)$ . Notice that, in order to allow agents of different clubs to interact directly, the system includes ad-hoc groups of the form  $\{i, j\}$  enabling the communication between  $i$  and  $j$ . On the other hand, the communication is allowed (or not), depending on the trust  $t_{ij}$  computed as shown above.

As another example, let us consider the encoding of EigenTrust [14] in our framework. First, observe that the local trust from  $I$  to  $J$  is given by  $s_{IJ} = mul((J, 1)_I) - mul((J, -1)_I)$ . Then,  $c_{IJ}$  is obtained through

the normalization function defined in [14]. Hence, the formula used to compute  $t_{IJ}$  is  $trust_{IJ}^{\{I,J\}}$ , where:

$$trust_{IJ}^S = c_{IJ} + \sum_{G.s.t.I \in G} \sum_{K \in G, K \neq S} c_{IK} \cdot trust_{KJ}^{\{K\} \cup S}.$$

## 4 Model checking trust properties

The formal framework proposed in this paper can be used as a basis for the verification of distributed trust systems. For this purpose, in [1], a model checking based approach is defined that relies on a trust temporal logic, called TTL, which is defined for the verification of TLTS-like models and, e.g., can be mapped to the logic UCTL [4]. Here, we specify the atomic statements of such a logic, which depend on the representation of trust information in our calculus, while the logical and temporal operators can be found in [1]. Similarly as for other logics merging action/state based predicates, atomic formulas include actions labeling TLTS transitions and state-based trust predicates:

$$1 \mid w \geq k$$

where the domain of variable  $1$  is the labels set  $L$  of the TLTS,  $k \in \mathbb{T}$ , and  $w$  is a trust variable, which can be equal to:

- $t_{IJ}$ , i.e., the trust of  $I$  towards  $J$  as computed by the trust system;
- $tf_{IJ} = f\{v \mid (J, v)_I \in \mathcal{E}\}$ , where function  $f : 2^{\mathbb{T}} \rightarrow \mathbb{T}$  is taken from a set  $TF$  of associative and commutative functions, like, e.g., sum, min, and count, provided that  $\mathbb{T} = \mathbb{Z}$ .

Therefore, an atomic statement is a predicate about either the trust between two agents as computed by the trust system, or the set of local, direct experiences between them. In this framework, trust temporal properties can be modeled and verified, like, e.g., “Can  $n$  malicious agents provide false feedback in order to compromise the reputation of a honest agent?”, or “Can an agent trust another agent without sufficient, positive, direct observations?”, thus making it possible the validation of a system against the attacks mentioned in Section 1.

## 5 Conclusion and future work

The formal modeling approach proposed in this paper joins the specification of distributed systems relying on an adaptive and flexible communication model with the specification of the trust model governing the interactions among concurrent processes. These two modeling frameworks are defined separately, as the mutual interaction between them is managed transparently at the level of the semantics of parallel composition.

As work in progress, we mention that the multiset of trust values storing the feedback about direct interactions can be enriched with additional information, such as, e.g., the age of each feedback. This can be done in order to weight the contribution of an experience depending on the time elapsed from the related interaction.

The information expressed by the trust infrastructure is employed to make the model quantitative, in a sense, without adding *numbers* to the behavioral specification of the agents. Such quantitative information can be used to solve nondeterminism in several different ways. For instance, the possibilistic choice among alternative trust-based communications from agent  $i$  to a set of trusted agents  $X$  can be made either probabilistic, by using as weights the trust of  $i$  towards each agent  $j$  in  $X$ , or prioritized, by

using the same trust values, or else a combination of the two policies can be applied. Details about the extension of the TLTS model that is obtained in such a way, which encompasses both nondeterminism and probabilities, can be found in [1].

Finally, as future work, it would be worthwhile to parameterize (without any substantial human intervention) the model checking based verification with respect to the different classes of attacks described in Section 1.

## References

- [1] A. Aldini (2015): *Modeling and Verification of Trust and Reputation Systems*. *Journal of Security and Communication Networks* 8(16), pp. 2933–2946, doi:10.1002/sec.1220.
- [2] A. Aldini, B. Bernardo & F. Corradini (2010): *A Process Algebraic Approach to Software Architecture Design*. Springer, doi:10.1007/978-1-84800-223-4.
- [3] A. Aldini & A. Bogliolo (2014): *Modeling and Verification of Cooperation Incentive Mechanisms in User-Centric Wireless Communications*. In D. Rawat, B. Bista & G. Yan, editors: *Security, Privacy, Trust, and Resource Management in Mobile and Wireless Communications*, IGI Global, pp. 432–461, doi:10.4018/978-1-4666-4691-9.ch018.
- [4] M. ter Beek, A. Fantechi, S. Gnesi & F. Mazzanti (2008): *An Action/State-Based Model-Checking Approach for the Analysis of Communication Protocols for Service-Oriented Applications*. In: *12th Workshop on Formal Methods for Industrial Critical Systems (FMICS'07)*, LNCS 4916, Springer, pp. 133–148, doi:10.1007/978-3-540-79707-4\_11.
- [5] T. Beth, M. Borchering & B. Klein (1994): *Valuation of Trust in Open Networks*. In: *Conference on Computer Security*, Springer, pp. 3–18, doi:10.1.1.50.7349.
- [6] L. Bortolussi, R. De Nicola, V. Galpin, S. Gilmore, J. Hillston, D. Latella, M. Loreti & M. Massink (2015): *CARMA: Collective Adaptive Resource-sharing Markovian Agents*. In Nathalie Bertrand & Mirco Tribastone, editors: *Procs. of 13th Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL, Electronic Proceedings in Theoretical Computer Science* 194, pp. 16–31, doi:10.4204/EPTCS.194.2.
- [7] L. Cardelli & A. D. Gordon (2000): *Mobile ambients*. *Theoretical Computer Science* 240(1), pp. 177–213, doi:10.1016/S0304-3975(99)00231-5.
- [8] J.-H. Cho, A. Swami & I.-R. Chen (2011): *A survey on trust management for mobile ad hoc networks*. *Communications Surveys & Tutorials* 13(4), pp. 562–583, doi:10.1109/SURV.2011.092110.00088.
- [9] S. Ganeriwal, L. K. Balzano & M. B. Srivastava (2008): *Reputation-based Framework for High Integrity Sensor Networks*. *ACM Trans. Sen. Netw.* 4(3), pp. 15:1–15:37, doi:10.1145/1362542.1362546.
- [10] J. A. Goguen & J. Meseguer (1982): *Security Policies and Security Models*. In: *IEEE Symposium on Security and Privacy*, pp. 11–20, doi:10.1109/SP.1982.10014.
- [11] G. Han, J. Jiang, L. Shu, J. Niu & H.-C. Chao (2014): *Management and applications of trust in Wireless Sensor Networks: A survey*. *Journal of Computer and System Sciences* 80(3), pp. 602–617, doi:10.1016/j.jcss.2013.06.014. Special Issue on Wireless Network Intrusion.
- [12] F. He, H. Zhang, H. Wang, M. Xu & F. Yan (2010): *Chain of Trust Testing Based on Model Checking*. In: *2nd Int. Conf. on Networks Security Wireless Communications and Trusted Computing, NSWCTC, IEEE*, pp. 273–276, doi:10.1109/NSWCTC.2010.264.
- [13] A. Jøsang (2007): *Trust and Reputation Systems*. In A. Aldini & R. Gorrieri, editors: *Foundations of Security Analysis and Design IV (FOSAD'07)*, LNCS 4677, Springer, pp. 209–245, doi:10.1007/978-3-540-74810-6\_8.
- [14] S.-D. Kamvar, M.-T. Schlosser & H. Garcia-Molina (2003): *The EigenTrust Algorithm for Reputation Management in P2P Networks*. In: *12th Conf. on World Wide Web (WWW'03)*, ACM, pp. 640–651, doi:10.1.1.11.4846.

- [15] W.-S. Kim (2009): *Effects of a Trust Mechanism on Complex Adaptive Supply Networks: An Agent-Based Social Simulation Study*. *Journal of Artificial Societies and Social Simulation* 12(3), p. 4. Available at <http://jasss.soc.surrey.ac.uk/12/3/4.html>.
- [16] M. Kwiatkowska, D. Parker & A. Simaitis (2013): *Strategic Analysis of Trust Models for User-Centric Networks*. In: *Int. Workshop on Strategic Reasoning (SR'13)*, 112, EPTCS, pp. 53–60, doi:10.4204/EPTCS.112.10.
- [17] G. Marion L. Vissat, J. Hillston & M. Smith (2016): *MELA: Modelling in Ecology with Location Attributes*. In: *Procs. of 14th Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL*.
- [18] J. Li, R. Li & J. Kato (2008): *Future trust management framework for mobile ad hoc networks*. *IEEE Communications Magazine* 46(4), pp. 108–114, doi:10.1109/MCOM.2008.4481349.
- [19] Z. Li & H. Shen (2012): *Game-Theoretic Analysis of Cooperation Incentives Strategies in Mobile Ad Hoc Networks*. *Transactions on Mobile Computing* 11(8), pp. 1287–1303, doi:10.1109/TMC.2011.151.
- [20] F. G. Marmol & G. M. Perez (2009): *Security Threats Scenarios in Trust and Reputation Models for Distributed Systems*. *Computers and Security* 28(7), pp. 545–556, doi:10.1016/j.cose.2009.05.005.
- [21] M. Momani (2010): *Recent Trends in Network Security and Applications: Third International Conference, CNSA 2010, Chennai, India, July 23-25, 2010. Proceedings*, chapter Trust Models in Wireless Sensor Networks: A Survey, pp. 37–46. Springer, doi:10.1007/978-3-642-14478-3\_4.
- [22] H. Mousa, S. Ben Mokhtar, O. Hasan, O. Younes, M. Hadhoud & L. Brunie (2015): *Trust management and reputation systems in mobile participatory sensing applications: A survey*. *Computer Networks* 90, pp. 49–73, doi:10.1016/j.comnet.2015.07.011.
- [23] E. C. H. Ngai & M. R. Lyu (2004): *Trust- and clustering-based authentication services in mobile ad hoc networks*. In: *24th Int. Conf. on Distributed Computing Systems Workshops*, IEEE, pp. 582–587, doi:10.1109/ICDCSW.2004.1284091.
- [24] H. S. Packer, L. Dragan & L. Moreau (2014): *An auditable reputation service for collective adaptive systems*. In Daniele Miorandi, Vincenzo Maltese, Michael Rovatsos, Anton Nijholt & James Stewart, editors: *Social Collective Intelligence: Combining the Powers of Humans and Machines to Build a Smarter Society*, Springer, pp. 159–184. Available at <http://eprints.soton.ac.uk/365559/>.
- [25] M. Reith, J. Niu & W. H. Winsborough (2007): *Apply Model Checking to Security Analysis in Trust Management*. In: *IEEE 23rd Int. Conf. on Data Engineering Workshop*, IEEE, pp. 734–743, doi:10.1109/ICDEW.2007.4401061.
- [26] A. Tarable, A. Nordio, E. Leonardi & M. G. Ajmone Marsan (2015): *The Importance of Being Earnest in Crowdsourcing Systems*. In: *IEEE Conference on Computer Communications, INFOCOM*, IEEE, pp. 2821–2829, doi:10.1109/INFOCOM.2015.7218675.
- [27] L. Xiong & L. Liu (2004): *PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities*. *IEEE Trans. on Knowl. and Data Eng.* 16(7), pp. 843–857, doi:10.1109/TKDE.2004.1318566.
- [28] R. Yaich, O. Boissier, P. Jaillon & G. Picard (2012): *An Adaptive and Socially-Compliant Trust Management System for Virtual Communities*. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing, SAC'12*, ACM, pp. 2022–2028, doi:10.1145/2245276.2232112.
- [29] Y. Yu, K. Li, W. Zhou & P. Lib (2012): *Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures*. *Journal of Network and Computer Applications* 35(3), pp. 867–880, doi:10.1016/j.jnca.2011.03.005.
- [30] Y. Zhang, L. Lin & J. Huai (2007): *Balancing Trust and Incentive in Peer-to-Peer Collaborative System*. *Journal of Network Security* 5, pp. 73–81, doi:10.1.1.148.3767.
- [31] R. Zhou & K. Hwang (2007): *PowerTrust: a Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing*. *Transactions on Parallel and Distributed Systems* 18(4), pp. 460–473, doi:10.1109/TPDS.2007.1021.