From cognitive trust theories to computational trust*

Jomi F. Hübner[1], Emiliano Lorini[2], Laurent Vercouter[1], and Andreas Herzig[2]

[1] ENS Mines Saint Etienne, France
{hubner,boissier,vercouter}@emse.fr
[2] IRIT, Toulouse, France
{lorini,herzig}@irit.fr

Abstract. Among the several categories of trust models, cognitive models have important features. Initially these models were only informally defined, but formalizations were recently proposed. The concepts of the models are thus sufficiently well defined to be implemented and evaluated. In this paper, the cognitive trust model proposed by Castelfranchi and Falcone is integrated into a BDI (belief, desire, intention) agent architecture and implemented with the *Jason* programming language. The ART testbed scenario is then used to experiment and evaluate both the model and the implementation.

1 Introduction

The concept of trust is important for recent application domains where agent technologies are relevant, such as information retrieval, e-commerce, and peer-to-peer systems. It has been in the focus of many research projects during the last few years, and many theoretical models and systems have been developed. One of the most prominent theoretical model is the *cognitive* model of trust proposed by [2], henceforth abbreviated C&F. Their informal definition of trust is formulated as an individual *belief* about some properties of the trustee.

In this paper we develop further this approach, with the aim of bridging the gap between C&F's cognitive theory and computational models. A first formalisation of C&F trust is proposed in [7], where the definition is refined step by step into more primitive concepts, namely actions, agency, preference and choice (Section 2 briefly presents this formalisation). We here evaluate this definition by means of the ART scenario, which is commonly used as a testbed for trust models (Section 3). We first present an implementation of the C&F definition in a BDI (belief, desire, intention) agent programming language (Section 4). The implementation of that conceptualisation of trust is suitable for such languages since both rely on the same concepts such as beliefs and goals. Besides showing that an agent equipped with the C&F concept of trust performs quite well against other agents of the ART testbed, an important result is that all the trustee's properties included in the C&F definition of trust is shown to be useful in the experiments (Section 5).

* The first two authors are financed by the ANR Project ForTrust (ANR-06-SETI-006)

2 Trust Definition

According to C&F, trust has four ingredients: a truster i, a trustee j, an action α of j, and a goal φ of i.[3] C&F provide a definition of trust which is based on four primitive concepts: capability, intention, power, and goal. In their definition, "i trusts j to do α in order to achieve φ" if and only if:

1. i has the *goal* φ;
2. i believes j is *capable* to do α;
3. i believes j has the *power* to achieve φ by doing α;
4. i believes j *intends* to do α.

For example, when i trusts j to send product P in view of satisfying i's goal of possessing P then (1) i wants to possess P, (2) i believes that j is capable to send P, (3) that j's sending P will result in i possessing P, and (4) that j has the intention to send P. C&F stress the importance of the goal component: it makes no sense to say that I trust j to do α when α is completely irrelevant for my goals.

In [7] this concept was detailed in two types: occurrent trust and dispositional trust. In the former case, the truster has a certain goal and believes that the trustee is going to act here and now in such a way that its goal will be achieved. In the latter case, the truster thinks to be possible that it will have a certain goal in the future and believes, whenever it will have such a goal, the trustee will act in such a way that the goal will be achieved. In this paper only the former type of trust is considered, and it is defined by:

$$\begin{aligned} OccTrust(i,j,\alpha,\varphi) \stackrel{\texttt{def}}{=} \; & Goal(i,\varphi) \wedge \\ & Believes(i, OccCap(j,\alpha)) \wedge \\ & Believes(i, OccPower(j,\alpha,\varphi)) \wedge \\ & Believes(i, OccIntends(j,\alpha)) \end{aligned} \tag{1}$$

2.1 The underlying BDI logic

The definition of occurrent trust presented in the previous section has been initially formalised in [10], where a modal logic for reasoning about trust in multi-agent system has been proposed. This logic enables us to specify the five predicates for belief, goal, capability, intention and power on the right hand side of the definition of occurrent trust (definition (1)), namely the predicates $Goal$, $Believes$, $OccCap$, $OccPower$ and $OccIntends$. The proposed logic (called $\mathcal{L}$) is a multimodal logic which combines the expressiveness of dynamic logic [6] with the expressiveness of a so-called BDI logic of agents' mental attitudes (see [4] for instance).

It is not the aim of this work to discuss the precise semantics of the modal operators of the logic $\mathcal{L}$. We just present them in an informal way in order to help the reader to understand the relationship between the logical specification of our trust model and its implementation in the *Jason* architecture.[4]

[3] We use α to denote actions and φ to denote goals.

[4] See [10] for an analysis of the semantics of these operators, their relationships, and their correspondence with the structural conditions on the models of the logic $\mathcal{L}$.

The syntactic primitives of the logic $\mathcal{L}$ are the following: countable sets of atomic formulas $ATM = \{p, q, \ldots\}$, agents $AGT = \{i, j, \ldots\}$ and actions $ACT = \{a, b, \ldots\}$. The language of $\mathcal{L}$ is the set of formulas defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \texttt{After}_{i:\alpha}\,\varphi \mid \texttt{Does}_{i:\alpha}\,\varphi \mid \texttt{Bel}_i\,\varphi \mid \texttt{Pref}_i\,\varphi$$

where p ranges over ATM, α ranges over ACT and i ranges over AGT. Thus, the logic $\mathcal{L}$ has four types of normal modal operators:[5] $\texttt{Bel}_i$, $\texttt{Pref}_i$, $\texttt{Does}_{i:\alpha}$, and $\texttt{After}_{i:\alpha}$.

These operators have the following intuitive meaning. $\texttt{Bel}_i\,\varphi$: the agent i believes that φ; $\texttt{After}_{i:\alpha}\,\varphi$: after agent i does α, it is the case that φ ($\texttt{After}_{i:\alpha}\,\bot$ is read: agent i cannot do action α); $\texttt{Does}_{i:\alpha}\,\varphi$: agent i is going to do α and φ will be true afterward ($\texttt{Does}_{i:\alpha}\,\top$ is read: agent i is going to do α); $\texttt{Pref}_i\,\varphi$: agent i prefers that φ holds.

Operators for actions of type $\texttt{After}_{i:\alpha}$ and $\texttt{Does}_{i:\alpha}$ are normal modal operators satisfying the axioms and rules of inference of system K [3]. Operators of type $\texttt{Bel}_i\,\varphi$ are just standard doxastic operators satisfying the axioms and rules of inference of system KD45. Therefore, positive and negative introspection over beliefs is supposed, and it is assumed that an agent cannot have inconsistent beliefs. Finally, operators of type $\texttt{Pref}_i$ are used to express an agent's binary preference. These are similar to Cohen & Levesque's operators [4]. It is supposed that every operator $\texttt{Pref}_i$ satisfies the axioms and rules of inference of system KD, that is, it is assumed that an agent cannot have conflicting preferences (i.e. an agent cannot prefer φ and $\neg\varphi$ at the same time).

The most important relationships between the four types of operators are expressed by the following logical axioms.

Active: $\bigvee_{i\in AGT,\alpha\in ACT} \texttt{Does}_{i:\alpha}\,\top$
Inc$_{Act,PAct}$: $\texttt{Does}_{i:\alpha}\,\varphi \rightarrow \neg\texttt{After}_{i:\alpha}\,\neg\varphi$
IntAct1: $(\neg\texttt{After}_{i:\alpha}\,\bot \wedge \texttt{Pref}_i\,\texttt{Does}_{i:\alpha}\,\top) \rightarrow \texttt{Does}_{i:\alpha}\,\top$
IntAct2: $\texttt{Does}_{i:\alpha}\,\top \rightarrow \texttt{Pref}_i\,\texttt{Does}_{i:\alpha}\,\top$

Axiom **Active** ensures that the world is never static, i.e. at every moment there exists an agent i and action α such that i performs α. This is the reason why the operator $\texttt{X}$ for *next* of LTL (linear temporal logic) can be defined as follows:
$\texttt{X}\varphi \stackrel{\texttt{def}}{=} \bigvee_{i\in AGT,\alpha\in ACT} \texttt{Does}_{i:\alpha}\,\varphi$. According to **Inc**$_{Act,PAct}$, if i is going to do α and φ will be true afterward, then it is not the case that $\neg\varphi$ will be true after i does α. Axioms **IntAct1** and **IntAct2** relate preferences with actions. Note that $\neg\texttt{Does}_{i:\alpha}\,\varphi \rightarrow \texttt{After}_{i:\alpha}\,\neg\varphi$ is not valid. According to **IntAct1**, if i has the preference to perform action α and can do action α then, i is going to do α. According to **IntAct2**, an agent is going to do action α only if it has the preference to perform action α: an agent's *doing* is by definition intentional. Similar axioms have been studied in [11] in which a logical model of the relationships between intention and action performance is proposed.

2.2 The logical definition of trust

The five predicates on the right hand side of the definition of occurrent trust (definition (1)) can be specified in the logic $\mathcal{L}$ as follows:

[5] We here typographically distinguish the informal predicates $Goal$, $Believes$, $OccCap$, $OccPower$ and $OccIntends$ in the definition of occurrent trust from the modal operators of the logic $\mathcal{L}$ written in typewriter font.

$$
\begin{aligned}
Believes(i,\varphi) &\stackrel{\text{def}}{=} \texttt{Bel}_i\,\varphi \\
Goal(i,\varphi) &\stackrel{\text{def}}{=} \texttt{Pref}_i\,\texttt{X}\varphi \\
OccCap(i,\alpha) &\stackrel{\text{def}}{=} \neg\texttt{After}_{i:\alpha}\,\bot \\
OccPower(i,\alpha,\varphi) &\stackrel{\text{def}}{=} \texttt{After}_{i:\alpha}\,\varphi \\
OccIntends(i,\alpha) &\stackrel{\text{def}}{=} \texttt{Pref}_i\,\texttt{Does}_{i:\alpha}\,\top
\end{aligned}
$$

Thus, agent i has the goal that φ, if and only if i prefers φ to be true in the next state; i has the capability to do α if and only if, i can do α (i.e. at the actual world there exists a possible occurrence of α performed by i); i intends to do α if and only if, i prefers to do α.

It is worth noting that, from Axioms **IntAct1**, **IntAct2**, and $\mathbf{Inc}_{Act,PAct}$ it follows that the following logical equivalence is a theorem of the logic $\mathcal{L}$: $(\neg\texttt{After}_{i:\alpha}\,\bot \wedge \texttt{Pref}_i\,\texttt{Does}_{i:\alpha}\,\top) \leftrightarrow \texttt{Does}_{i:\alpha}\,\top$. Therefore, i's occurrent capability and i's occurrent intention to perform action α are together equivalent to the fact that i performs action α, that is:

$$
(OccCap(i,\alpha) \wedge OccIntends(i,\alpha)) \leftrightarrow \texttt{Does}_{i:\alpha}\,\top \tag{2}
$$

This is the reason why the definition of occurrent trust given in the previous section can be simplified as follows:

$$
\begin{aligned}
OccTrust(i,j,\alpha,\varphi) \stackrel{\text{def}}{=}\ & Goal(i,\varphi) \wedge \\
& Believes(i, OccAct(j,\alpha)) \wedge \\
& Believes(i, OccPower(j,\alpha,\varphi))
\end{aligned} \tag{3}
$$

where $OccAct$ is a predicate used to express action occurrence defined by:
$OccAct(j,\alpha) \stackrel{\text{def}}{=} \texttt{Does}_{j:\alpha}\,\top$.

This formalisation of occurrent trust expresses a fundamental aspect of the trust concept, namely the fact that the truster has a goal that φ and believes that the trustee is going to ensure φ by performing action α.

2.3 From binary trust to graded trust

In a recent extension of the previous logic of trust [9], the authors moved from binary trust (i.e. either i trusts j or does not) to graded trust (i.e. agent i trusts agent j with a certain strength x). To this aim, the doxastic operators of the form $\texttt{Bel}_i$ were generalised to normal operators for graded beliefs of the form $\texttt{Bel}_i^x$ where $i \in AGT$ and $x \in [0,1]$. A formula $\texttt{Bel}_i^x\,\varphi$ means: agent i believes φ at least with strength x. Therefore $\texttt{Bel}_i^1\,\varphi = \texttt{Bel}_i\,\varphi$.

At the semantic level, every operator $\texttt{Bel}_i^x$ is interpreted according to a corresponding accessibility relation R_i^x over possible worlds $w, w', ...$ It is supposed that, given two possible worlds w and w', if $x > y$ then $R_i^y \subseteq R_i^x$. Thus, for every agent i, the accessibility relations in $\{R_i^x | x \in [0,1]\}$ induce a so-called system of spheres [8]. This constraint on the accessibility relations R_i^x corresponds to the following logical axiom:

$\mathbf{Inc}_{Bel}$ $\quad \texttt{Bel}_i^x\,\varphi \rightarrow \texttt{Bel}_i^y\,\varphi$

That is, if $x > y$ and i believes φ at least with strength x then i also believes φ at least with strength y. More generally, the logic of graded beliefs validates:

$$(\mathtt{Bel}_i^{x_1}\,\varphi_1 \wedge ... \wedge \mathtt{Bel}_i^{x_m}\,\varphi_m) \rightarrow \mathtt{Bel}_i^{min(x_1...,x_m)}\,(\varphi_1 \wedge ... \wedge \varphi_m)$$

Such operators of graded belief can be used to represent truster's beliefs with different strengths about different properties of the trustee. As we will show in Section 4, this aspect is important when moving from the abstract model of trust reasoning to the implementation in *Jason*. For example, one would like to say that i (the truster) believes at least with strength x that j (the trustee) will perform action α, or that i believes at least with strength y that j has the power to achieve φ by doing α. These two facts are respectively represented by the formulas $\mathtt{Bel}_i^x\,\mathtt{Does}_{j:\alpha}\,\top$ and $\mathtt{Bel}_i^y\,\mathtt{After}_{j:\alpha}\,\varphi$.

3 Occurrent trust applied to ART scenario

We apply the definition of trust presented in Section 2 to the ART testbed scenario (`http://art-testbed.net`). This scenario is proposed by the trust community as a common testbed for experimentation and evaluation of multi-agent trust models. The ART scenario consists in a simulation of painting appraisals. Several agents are in competition and each agent has a few paintings to evaluate. A painting belongs to a given era and the agents have different level of expertise allowing them to be more or less skilled in the evaluation of a painting's rating according to its era. At each time-step, an agent receives from simulated clients a set of paintings to evaluate. An agent cannot evaluate all the paintings from its own clients and it has to rely on other agents to do it. This is called the *opinion* protocol, where an agent asks other agents an appraisal (or an opinion) for its paintings. In order to choose who to ask for opinions, an agent can use its past direct experiences, and/or two interaction protocols: (i) the *certainty* protocol, according to which, the agent directly asks other agents about their own expertise; (ii) the *reputation* protocol, according to which, the agent asks other agents what is the reputation of a third agent. Every agent has the possibility to lie when communicating in these protocols. Agents are payed when appraising paintings and if they were accurate, they receive more clients at the next step so that the accurate appraisers earn more money.

An agent i may use the concept of trust presented in Section 2 to select a partner j to whom to ask for an appraisal for a painting. To use that conceptualisation, we need to identify the actions and goals in the context of ART. All agents share the same set of possible actions: to appraise paintings of a specific era. The goal of each agent is to give the best possible evaluation for its clients' paintings. In order to achieve this, an agent must select partners to ask for appraisals. Thus, the sentence 'agent i trusts j to appraise a painting p' can be written as follows:

$$\begin{aligned} &OccTrust(i, j, \text{appraise(p)}, \text{good\_eval(p)}, min(x, y)) \\ &\stackrel{\mathtt{def}}{=} Goal(i, \text{good\_eval(p)}) \wedge \\ &\quad Believes(i, OccAct(j, \text{appraise(p)}), x) \wedge \\ &\quad Believes(i, OccPower(j, \text{appraise(p)}, \text{evaluate(p)}), y) \end{aligned}$$

where x and y are the strengths of the two beliefs used in the formula; and $Believes(i, \varphi, x) = \texttt{Bel}_i^x\, \varphi$.

Having identified the actions and goals for ART, the next and more complex step is to develop some mechanisms which allow agent i to infer those beliefs about the properties of j which are relevant for the achievement of its goal of giving the best possible evaluation for the paintings. Namely, these mechanisms should allow i to evaluate whether the predicates $OccPower(j, \alpha, \varphi)$ and $OccAct(j, \alpha)$ hold in such a way that i can assess the trustworthiness of j. The former predicate denotes j's power to appraise a painting that will help i's goal to give the best possible evaluation for its paintings, whereas the latter denotes that j is going to provide its opinion about the paintings.

A first mechanism is to obtain information about the power of j by means of the *certainty* protocol available in ART. Of course, agents may lie about their expertise possibly leading to incorrectness in i's belief about the predicate $OccPower$. A second mechanism, that can also be applied for $OccAct$, consists in using previous experiences of interaction with j. For instance, if in previous collaborations (when j was asked to respond), j has provided appraisals for i's paintings, then it is concluded that now j has the power to provide appraisals and is going to respond (given that i has asked him). While the first mechanism concerns sincerity issues, the second mechanism concerns all problems related with learning. A third mechanism is to ask other agents their opinions about j's properties, that is, to ask other agents whether the predicates $OccPower(j, \alpha)$ and $OccAct(j, \alpha, \varphi)$ hold. In other words, this third mechanism consists in discovering the reputation of j. However, issues related to reputation are not considered yet in the current stage of our work.

To sum up, in the ART scenario an agent i can exploit various sources of information in order to assess the trustworthiness of some target agent j: communication with j, direct experiences with j, and the reputation of j.

4 From the abstract model to an agent implementation

This section describes how the definition of trust presented in the previous section can be designed and implemented for an agent that participates in the ART scenario. Once the concept of trust is defined on the basis of cognitive ingredients (beliefs, goals, etc.), a suitable agent architecture and programming language should be chosen. For this work, the BDI architecture and the *Jason* programming language were chosen [1]. The main reason to select this language is that it is perfectly suitable for an implementation of the formal definition of trust discussed in Section 2. *Jason* is selected since it is both based on logic programming and on the BDI architecture. Other kinds of architecture and language could be chosen. However, the goal here is to concretely show that the concept can be implemented in at least one configuration.

Figure 1 illustrates the main components of the agent architecture. Briefly, there are data structures for the agent's beliefs, goals, plan library (set of possible plans to achieve goals), and intentions (current plans in execution to achieve the goals of the agent). The *perceive process* updates the belief base from the incoming messages and the *act process* selects an action to be performed from the current set of intentions. The *trust inference* has to decide whether to trust an agent or not. For that purpose a *theoretical*

reasoning may be enough, in the case where a conclusion can be draw from the current beliefs (for instance, from past experiences). However, in some circumstances a kind of *practical reasoning* may be necessary, i.e. some sequence of actions are required to obtain the necessary information for the trust decision (as in the case where the reputation of an agent has to be asked to others). In this latter case, a new intention is created to perform those actions and obtain the required information.

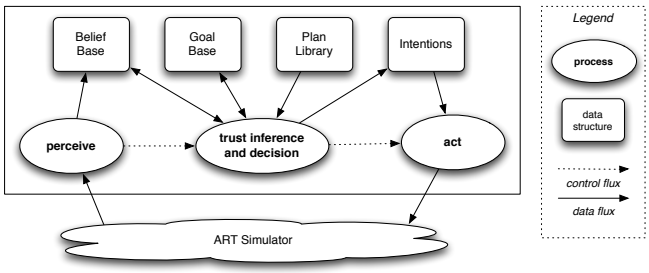

Fig. 1. General agent architecture for trust

The first requirement for the development of our agent is the integration of the ART testbed agent architecture (where the agent have to be coded in Java) and a *Jason* agent architecture that allows the programming of the agent using BDI primitives. *Jason* provides a suitable support to allow this kind of customisation. Roughly, this component provides as perception all data that come from the ART simulator and translates the agent's actions into suitable messages to the simulator. When some particular decision is required for the agent, a new goal is introduced into the reasoning cycle of the agent. For instance, when the simulator requires that the agent performs all the certainty requests, a new goal `!prepareCertaintyRequests` is created. During the agent reasoning process, a suitable plan will be selected to try to achieve this goal resulting in the execution of actions that correspond to a reputation request.

The perception provided by the architecture is translated to first-order predicates and included in the belief base with a special annotation that indicates that they correspond to the agent's perception. Older belief-perceptions are removed accordingly. Among these beliefs, the following are given by the ART simulator and used in the sequel:

- $painting(e, p, t)$: represent that the painting p of era e is allocated to the agent at the current step t of the simulation. The agent's paintings are perceived at the begin of each simulation step.
- $opinion(j, e, v_g, v_r, t)$: represents the appraisal produced by partner j for a painting of era e; the real value of the painting as defined by the simulator is v_r and the opinion provided by agent j is v_g. The quality of the opinion provided by j is based on the difference between v_g and v_r. This sort of information is provided to the agents at the end of each simulation step so that they can evaluate their selection of partners.

All beliefs described above are considered as having strength 1.

In each simulation step for the ART scenario, our agent receives several paintings to evaluate. For each painting it initially assigns n partners using exploitation and exploration strategies (n is the maximum number of opinions an agent can ask for a painting). The exploitation strategy tries to select the n most trustworthy agents in the correspond-

ing era of the painting. If there are not enough trustful agents, partners are randomly selected among the sincere agents (exploration strategy).

The identification of trustful agents uses the definition of occurrent trust (definition 3), i.e. trust is inferred from the agent's goals and beliefs (see the code of Figure 2).[6] The first component of the trust definition is $Goal(i, \varphi)$. In order to check whether this predicate holds, we simply consult all the intentions of the agent. The second component, the belief about $OccAct(j, \alpha)$, is inferred using the following implication ($\alpha = appraise(p)$):

$$Believes(i, OccAct(j, \alpha), x) \leftarrow Believes(i, opinions\_count(j, a, g), 1) \wedge a > 0 \ \wedge \ x = \frac{g}{a} \ \wedge \ x > \epsilon \quad (4)$$

where $opinions\_count(j, a, g)$ is the fact that there were a opinions that were asked to j, and g opinions provided by j. Thus, agent i believes that j is going to collaborate if i has previously interacted with j ($a > 0$) and the percentage of answers provided by j is greater than ϵ ($\epsilon = 0.9$ in our experiments). The strength x of the belief about $OccAct$ is $x = \frac{g}{a}$. Although we use only direct experiences to infer $OccAct(j, \alpha)$, the very particular mechanism used for that could be more complex and efficient. The goal in this paper however is not to optimise the mechanism, but rather to illustrate the implementation of the concept and to compare its influence in the agent performance differentiating agents that consider the predicate $OccAct(j, \alpha)$ in their trust reasoning from those that do not consider it.

The third component of the trust definition, the belief about the property $OccPower(j, \alpha, \varphi)$, is inferred by the following implication when the goal is to have a good evaluation for a painting p ($\varphi = good\_eval(p)$) and the action is to appraise the painting ($\alpha = appraise(p)$):

$$Believes(i, OccPower(j, \alpha, \varphi), y) \leftarrow Believes(i, sincere(j), 1) \wedge Believes(i, painting(e, p), 1) \wedge y = image_t(j, e) \ \wedge \ y > \delta \quad (5)$$

where $sincere(j)$ holds when j is believed to be sincere (based on previous interactions with j); $painting(e, p)$ is given as perception by the simulator and is used here to retrieve the era e of painting p; and $image_t(\alpha, e)$ is a function ($image_t : AGT \times ERA \rightarrow [0, 1]$) that maps each agent and era of the simulation step t to the corresponding agent's image. The strength of j power is the same value as its ($y = image_t(j, e)$). Thus, agent i believes that j has power to give a good evaluation on some era if j is sincere and currently has an image greater than δ ($\delta = 0.5$ in our experiments).

The definition of the image function is inspired by reinforcement learning techniques and the Q-Learning algorithm [15]. The reward of asking opinions to j in a simulation step t is given by the mean of all errors in j's opinions:

$$r_t(j, e) = \frac{1}{\#O_t^{j,e}} \sum_{(v_g, v_r) \in O_t^{j,e}} 1 - \frac{|v_g - v_r|}{v_r}$$

[6] The purpose of adding this excerpt of code is twofold: to provide some details of the functioning of the agent and to show how our proposal is implemented in a BDI approach. We do not have the space here to introduce the language; however, we added comments in the code to explain the meaning of the main parts.

```
// trust inference rule, e.g. Act=appraise(p1), Goal=good_eval(p1)
trust(J,Act,Goal)[strength(C)] :-
   .intend(Goal) &                        // I have the goal
   occ_act(J,Act)[strength(X)] &          // J is capable and intend
   occ_power(J,Act,Goal)[strength(Y)] & // J has the power
   C = math.min(X,Y).                     // computes the strength of the trust
   // the strength of beliefs are represented by annotations, enclosed by [ and ]

// when a painting is allocated to me, to evaluate it is a goal
+painting(Era,P) <- !good_eval(P).

// capability and intention are based on the percentage of responses to requests
occ_act(J,appraise(P))[strength(X)] :-
   opinions_count(J,Asked,Provided) & Asked > 0 & X = Provided/Asked & X > 0.9.

// power is based on image and sincerity
occ_power(J, appraise(P), _)[strength(Y)] :-
   sincere(J) & painting(Era,P) & image(J, Era, Y) & Y > 0.5.
   // the image function is implemented as a belief where the third term is
   // the value of the image of agent J

// whenever I receive an opinion from J
+opinion(J, Era, GivenValue, RealValue)
   <- Error = math.abs(RealValue - GivenValue) / RealValue;
      if (Error > 10) {  // huge errors means insincerity
         +~sincere(J)    // add a belief that J is not sincere
      };
      N = .count(opinion(J,Era,_,_)); // number of opinions
      R = (1-Error)/N;                // reward for the opinion
      ?image(J, Era, Img);            // consult current image
      NewImg = 0.5*Img + 0.5*R;       // compute new image
      -+image(J, Era, NewImg).        // update image belief
```

Fig. 2. Excerpt of the implementation of the trustfulness evaluation in *Jason*

where $O_t^{j,e}$ is the set of all opinions provided by agent j to our agent in paintings of era e and simulation step t; $\#O_t^{j,e}$ is the cardinality of this set; and each element of the set is a pair (v_g, v_r) where v_g is the value provided by j and v_r the real value of the painting.

Considering t as the current simulation step, the current image of j is calculated from the reward of asking opinions to j and the previous image of j:

$$image_t(j,e) = \begin{cases} 0.5 & \text{if } t = 0 \\ image_{t\text{-}1}(j,e) & \text{if } O_t^{j,e} = \emptyset \\ \gamma\ r_t(j,e) + (1\text{-}\gamma) image_{t\text{-}1}(j,e) & \text{otherwise} \end{cases}$$

The first case of the function, when $t = 0$, represents the initial image of j, i.e. 0.5. The second case is selected when no opinion was provided by j in step t, the image of the previous step is then used. The third case uses the reward of asking opinions to j and the previous image. The value of γ ($0 \leq \gamma \leq 1$) represents a discount for past images. We use $\gamma = 0.5$ meaning that the current experiences have the same importance than past experiences.

The above implementation is then used by our agent in each simulation step as follows. (1) For each painting that the agent has to evaluate, assign n partner agents. (2) Participate in reputation protocol. In this implementation, our agent does not ask for any reputation information. It simply answers to reputation requests using the internally

build image of others. (3) Participate in certainty protocol. Besides providing answers to requests, where our agent is always sincere, the certainty of the partners are requested. (4) Participate in opinion protocol. In this phase, our agent asks partners for opinions and accepts to provide opinions for every request. The accuracy of the opinion provided by our agent depends on the sincerity of the requester (more sincere agents receive more accurate opinions). (5) Update some beliefs based on the information available in the end of the simulation step: check whether the partners have provided or not an opinion for my paintings and update the $opinions\_count$ belief accordingly; update the $image$ of the partners based on the quality of the opinion they have produced; and update the sincerity property of the agents.

5 Experiments

Two experiments were done with our agent in the ART testbed. In both cases we used the configuration of the 2008 contest and, to produce the graphs, the mean of 10 executions is considered.

In the first experiment the four better placed agents of the 2008 AAMAS ART Contest were included (Uno, Connected, ForPrefect, and Next). We also added one cheating agent (that does not collaborate) and one honest agent (that always does the best for the partners). The result is shown in Figure 3. Our agent, identified by 'ForTrust', is in the group of agents placed second. Although it shows that our agent works quite well, the final performance of the agent is strongly dependent on the particular mechanism used to infer $OccAct$ and $OccPower$ and some parameters like ϵ, δ, and γ. As said before, we are not looking for the optimisation of those parameters here.

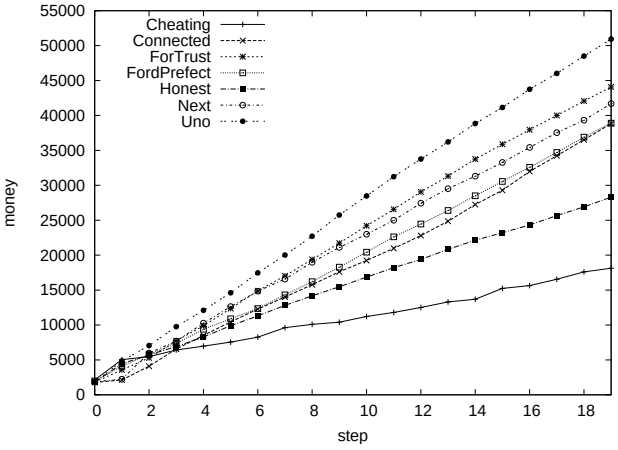

Fig. 3. Simulation results for our agent against some participants of the ART 2008 contest.

In the second set of experiments we intend to identify how the $OccAct$ and the $OccPower$ components of the trust definition interfere in the agent performance. For such an evaluation, four configurations of our agent were created:

Type1: this agent uses the complete definition of trust as presented in Sec 3.
Type2: the trust inference is based on $OccPower$.

Type3: the trust inference is based on $OccAct$.
Type4: this agent trusts in everybody.

Against these four agents, we put four honest agents, one cheating agent, and two lazy agents. Lazy agents are those that promise to provide an opinion (in ART we simulate this by the lazy agents asserting that they are experts), but that do not provide an opinion when someone ask them to do that. Briefly, lazy agents have the power to provide opinions but do not have the intention. The comparative performance of the four types of agents is shown in Figure 4. We can see that Type1, that uses the all the ingredients of the concept of trust performs better. To explain the result, we have to take a closer look at the partners this type of agent. Figure 5 shows how many requests of opinions were done at each simulation step by the agents of Type1 and Type2 respectively —the graph represents thus with whom the agent is interacting. After the exploration phase (around the step 8), the agents start to exploit their trust on other agents. While the agent Type1 rarely interacts with lazy agents since it also considers $OccAct$, the agent of Type2 continues to interact with lazy agent as often as with honest agents.

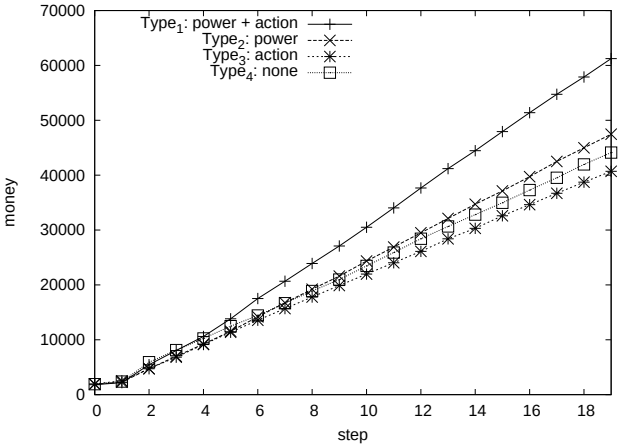

Fig. 4. Comparison of agents that use different ingredients of trust.

6 Discussion and related works

The ART scenario brings out some advantages for our experiments since it is well known by the community. It provides useful tools for the analysis of the experiments and other agents (from previous contests) to be included in the simulation and that we can then compare against our proposal. Nevertheless, some constraints might be cited. First, an important feature of the C&F definition of trust is to allow the truster to deal with different goals and actions, in ART however there is only one relevant type of action and it is the same for all agents. Second, the BDI architecture and the *Jason* language are suitable for environments where the agents have to be pro-active, while the ART simulator forces the agents to be just reactive to the protocols of each simulation step.

Although several trust models exist in the literature (a survey is presented in [14]), few of them are based on cognitive concepts. Not only few cognitive models of trust exist, but their integration into an agent architecture is rare. A first work in this direction is [12]. In that work, Pinyol and Sabater propose the integration of the concept of image from Repage reputation model with a BDI agent architecture. Their proposal consists of identifying how the reputation of other agents can influence the beliefs, desires, and intentions of the agent. Although we do not consider reputation in our proposal, our contribution is to use a general concept of trust (where the reputation can be integrated), propose an implementation, and evaluate the proposal in the ART scenario.

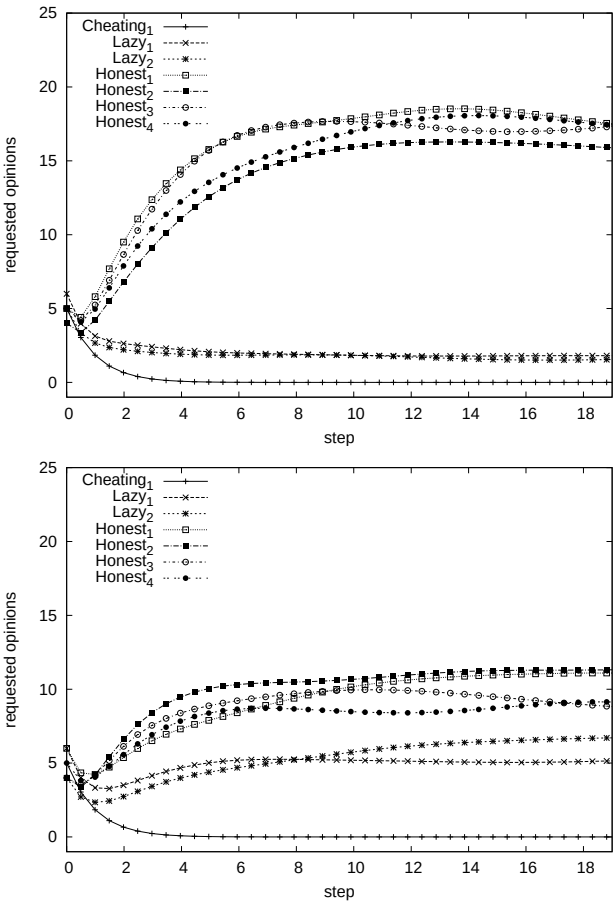

Fig. 5. Partners of the agent Type1 (top) and agent Type2 (bottom).

An important feature of our proposal is that the integration considers two directions: from trust to BDI and vice-versa. For example, when the agent intends to ask an opinion, the trust model is used; conversely, the trust reasoning may trigger new intentions to support the trust decision.

7 Conclusions

We conclude that the *cognitive* concept of trust as proposed by Castelfranchi and Falcone and formalised in Section 2 can be implemented and used by a concrete agent architecture. That concept is particularly suitable to be implemented in a BDI based language as provided by *Jason*. Although we do not take into account other BDI lan-

guages, the same conclusion may likely be drawn for similar languages like 2APL [5] and Jadex [13].

Our agent performed well against those of the 2008 ART Competition (2nd rank). The experiments in the ART testbed showed that, with certain types of agents (as the lazy agents used in the experiment), an agent that uses a concept of trust that considers all the ingredients proposed by C&F (goal, capability, power and intention), performs better than an agent that uses only a subset of these ingredients. Some features of our proposal are however not well explored and evaluated due to the limitations of the ART scenario. Future works will include the evaluation of our proposal in more complex scenarios. We also plan to include *reputation* as an important source of information to decide whether the trustee is going to act for the truster's goal and has to power to do that.

References

1. R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak using Jason*. Wiley Series in Agent Technology. John Wiley & Sons, 2007.
2. C. Castelfranchi and R. Falcone. Social trust: A cognitive approach. In C. Castelfranchi and Y. H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer, 2001.
3. B. F. Chellas. *Modal logic: an introduction*. Cambridge University Press, 1980.
4. P. R. Cohen and H. J. Levesque. Reasons: Belief support and goal dynamics. *Artificial Intelligence*, 42:213–61, 1990.
5. M. Dastani. 2APL: a practical agent programming language. *Autonomous Agent and Multi-Agent Systems*, 16:241–248, 2008.
6. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, 2000.
7. A. Herzig, E. Lorini, J. F. Hübner, J. Ben-Naim, C. Castelfranchi, R. Demolombe, D. Longin, L. Vercouter, and O. Boissier. Prolegomena for a logic of trust and reputation. In *Proc. of 3rd International Workshop on Normative Multiagent Systems (NorMAS 2008)*, pages 143–157, 2008.
8. D. Lewis. *Counterfactuals*. Basil Blackwell, 1973.
9. E. Lorini and R. Demolombe. From binary trust to graded trust in information sources: a logical perspective. In *Trust in Agent Societies 2008*, LNAI, pages 205–225. Springer-Verlag, 2008.
10. E. Lorini and R. Demolombe. Trust and norms in the context of computer security. In *Proc. of the Ninth International Conference on Deontic Logic in Computer Science (DEON'08)*, LNCS, pages 50–64. Springer-Verlag, 2008.
11. E. Lorini and A. Herzig. A logic of intention and attempt. *Synthese*, 163(1):45–77.
12. I. Pinyol and J. Sabater. Cognitive social evaluations for multi-context BDI agents. In *Proc. of Ninth Annual International Workshop Engineering Societies in the Agents World (ESAW'08)*, 2008.
13. A. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: A BDI reasoning engine. In R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, editors, *Multi-Agent Programming: Languages, Platforms, and Applications*, number 15 in Multiagent Systems, Artificial Societies, and Simulated Organizations, chapter 6, pages 149–174. Springer, 2005.
14. J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24:33–60, 2008.
15. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.