An Interaction-oriented Model of Trust Alignment

Andrew Koster, Jordi Sabater-Mir, and Marco Schorlemmer

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Bellaterra, Spain
{andrew, jsabater, marco}@iiia.csic.es

Abstract. We present a mathematical framework and an implementation of a proof of concept for communicating about trust in terms of interactions. We argue that sharing an ontology about trust is not enough and that interactions are the building blocks that all trust- and reputation models use to form their evaluations. Thus, a way of talking about these interactions is essential to gossiping in open heterogeneous environments. We give a brief overview of the formal framework we propose for aligning trust and discuss an example implementation, which uses inductive learning methods to form a trust alignment. We highlight the strengths and weaknesses of this approach.

1 Introduction

In complex, distributed systems, such as multi-agent systems, the artificial entities have to cooperate, negotiate, compete, etc. amongst themselves. Thus the social aspect of these systems plays a crucial role in their functioning. One of the issues in such a social system is the question of whom to trust and how to find this out. There are several systems already in development that model trust and reputation [1], ranging from a straightforward listing of evaluations (such as eBay's [2] reputation system), to complex cognitive models (such as Repage [3]). We anticipate that in an open multi-agent system, different programmers and users will have different wishes; leading to a large diversity of trust models. However, even if there is consensus on some model, we argue that in a heterogeneous environment it is inevitable that, if the trust model an agent uses is based on cognitive principles, the way different agents interpret their environment will still lead to differences in trust. We will show how, despite agreeing on the ontological underpinnings of the concepts, there is the need to align trust models.

The main question we address in this article is:

What information would be useful for agents to assess the reliability of communication about trust and what methods can be used for this assessment?

2 Related Work and Our Approach

We are not the only ones to consider the communication between agents about trust as a problem and some work has been done in defining common ontologies

for trust [4, 5], however in practice these ontologies do not have the support of many of the different trust methodologies in development. Even if support were added for all systems and a common ontology emerged, we could still not use it to communicate effectively. Trust is an inherently personal phenomenon and has subjective components which cannot be captured in an ontology. An adaptable approach that takes the different agents' points of view into account is needed, which will allow agents to learn an alignment even when the other agents don't share the ontology.

Abdul-Rahman and Hailes' reputation model [6] approaches the problem from another direction, by defining the trust evaluations based on the actual communications. The interpretation of communicated trust evaluations is based on previous interactions with the same sender. The problem with this, however, is that it is incomplete: firstly it assumes all other agents in the system use the same model, which in a heterogeneous environment will hardly ever be the case. Secondly, it uses a heuristic based on prior experiences, to "bias" received messages. This bias is an average of all previous experiences. They do not differentiate between different kinds of experiences, which are based on different types of interactions.

We propose to enrich the model of communication by considering it separate from the actual trust model. By doing this, we can allow for different trust models. We note, however, that while trust is modeled in disparate ways, all definitions do agree on the fact that trust is a social phenomenon. Just as any social phenomenon, it arises from the complex relationships between the agents in the environment and, without losing generality, we say these relationships are based on any number of interactions between the agents. These interactions can have many different forms, such as playing squash with someone, buying a bicycle on eBay or telling Alice that Dave is a trustworthy keynote speaker. Note that not all interactions are perceived equally by all participants. Due to having different goals, agents may observe different things, or even more obviously: by having a different vantage point. Simply by having more (or different) information available, agents may perceive the interaction itself differently. In addition, interactions may be accompanied by some kind of social evaluation of the interaction. These can range from an emotional response, such as outrage at being cheated in a trade, to a rational analysis. Thus, we see that how an agent experiences an interaction is unique and personal. This only adds to the problem we are considering. To be able to align, there needs to be some common ground from which to start the alignment, but any agent's experience of an interaction is subjective, and thus not shared. We call this personal interpretation of the interaction an *observation*. We say an agent's observations support its trust evaluations of other agents.

Now that we have discussed what interactions mean to a single agent, we will return to the focus of communicating about trust. One interaction may be observed by any number of agents, each making different observations, which support different trust evaluations of different targets performing different roles. However, to communicate about trust evaluations, the agents need to have a

starting point: some basic building blocks they implicitly agree they share. We note that the interactions provide precisely such a starting point. While all the agents' observations are different, they do share one specific thing: *the interaction itself*. We therefore argue that to find a reliable alignment between two agents they can align based on these interactions.

Our approach uses these shared interactions as building blocks to align the agents' trust models. The agents specify which interactions were observed to support a certain trust evaluation. Another agent can then calculate the own trust based on its observations of those interactions. This allows them to assemble information about the different trust evaluations agents support with the same set of interactions. An alignment of trust models, based on these relations, gives a way of interpreting other agents' communicated trust evaluations by starting from the set of interactions they share. In the following section we will give a brief overview of the formalization of this idea.

3 Theoretical Foundations

Before we consider possible solutions we need a clear definition of the problem we are considering. We follow the formalization we described in [7] and will summarize it briefly in the following sections. Firstly we consider agents with heterogeneous trust models, but we have no clear description of what a trust model is in the first place. Furthermore, to align, the agents need to communicate. For this we will need to define a language. And finally, the agents need to have some method of forming an alignment based on the statements in this language.

3.1 A Formal Representation of Trust Models

As argued in Section 2, interactions form the building blocks for talking about trust. An interaction is observed by different agents and represented internally by them. These observations then lead to trust evaluations of the various agents involved. Any trust model can therefore be described as a binary relation between an agent's observations and its trust evaluations. In addition, trust always has a target: any form of representing trust will have a trusting agent and a target agent. It is assumed that any agent's trust evaluations can be represented in some formal language $\mathcal{L}_{Trust}$. Note that because trust is a subjective phenomenon, the semantics of this language aren't shared, but by sharing the syntax the agents can communicate about it. A trust model is therefore a binary relation $\models$, such that $X \models \varphi$ means that there is a set of observations X which support trust evaluation $\varphi \in \mathcal{L}_{Trust}$. The observations X are unknown as they are an internal representation of the agent. However, we know these are based on some set of interactions. If $\mathfrak{O}$ is the set of an agent's possible observations and $\mathfrak{I}$ is the set of all interactions in the environment, then each agent has a function $observe : \mathfrak{I} \rightarrow \mathfrak{O}$ which associates interactions with observations. The observations X in the trust model are therefore generated (with the *observe*-function) from some

set of interactions I. These interactions are facts in the environment and we assume all agents may know about them and can use them as the basis of an alignment.

3.2 Formalizing gossip

We already mentioned $\mathcal{L}_{Trust}$, the language to talk about trust evaluations. However, a second component is needed for effective alignment: a language in which to talk about the interactions. Because knowing which information about the interactions is relevant is dependent on the domain, it is called $\mathcal{L}_{Domain}$. This language allows the agents to talk about interactions and sets of interactions. More specifically they can now send messages to each other giving their trust evaluation of an agent, as well as a description of the set of interactions this evaluation is based upon.

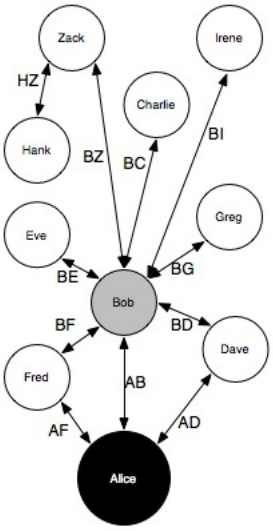
The agents align by gossiping about different targets: communicating their trust evaluations of a target in $\mathcal{L}_{Trust}$ and about the interactions these evaluations are based on in $\mathcal{L}_{Domain}$. Gossip from agent B to agent A is defined as a message $\texttt{gossip}(T, \beta, \psi)$, with T the target of the trust evaluation $\beta \in \mathcal{L}_{Trust}$ and $\psi \in \mathcal{L}_{Domain}$ pinpointing the interactions I such that $observe_B(I) \models_B \beta$.

The receiving agent A can now use the own trust model to find an $\alpha \in \mathcal{L}_{Trust}$, such that $observe_A(I) \models_A \alpha$ and the resulting rule $\langle \alpha, \beta, \psi \rangle$ will form the basis of our alignment. What this rule means is: the interactions which support ψ, support trust evaluation α for agent A and β for agent B. The goal is now to find a way of generalizing from such rules to a more general, predictive model, such that, for example, agent A can know what trust evaluation α' it should associate with a certain $\beta' \in \mathcal{L}_{Trust}$ given ψ, despite not knowing either the interactions which support ψ or not being able to conclude an own trust evaluation from the observation of those interactions.

3.3 Generalizations and coverage

Now that we have a way of describing the relationship (alignment) of two agents' trust models with regards to a specific target, we wish to expand this idea to a more predictive model: we wish to find the more general alignment between the trust models. This problem is considered as an inductive learning problem [8]. Given a number of targeted alignments with regards to different agents, is there an alignment that describes all (or most) of them?

To use inductive learning, it is necessary to define what the solution should look like. This should be a generalisation of the abovementioned rules $\langle \alpha, \beta, \psi \rangle$. We note that both $\mathcal{L}_{Trust}$ and $\mathcal{L}_{Domain}$ should be representable in a standard first order logic. Thus it is possible to use θ-subsumption to generalise these rules. The way to do this is by structuring the search space. The solution should be the least general alignment, which covers all the rules given in the messages. A hypothetical alignment $\mathfrak{T}$ is said to cover a rule $\langle \alpha, \beta, \psi \rangle$ if there is a rule $\langle \Gamma, \Delta, \Psi \rangle \in \mathfrak{T}$ such that all sets of interactions I which support $\langle \alpha, \beta, \psi \rangle$ also support $\langle \Gamma, \Delta, \Psi \rangle$. One hypothetical alignment $\mathfrak{T}$ is more general than another



Interaction	Observation
AB	Bob has written some excellent articles.
AD	Alice and Dave co-authored some papers.
AF	Fred is a good friend of Alice's husband.
BC	Charlie won a scholarship over Bob a long time ago.
BD	Bob and Dave play squash together.
BE	Bob thinks Eve is lazy.
BF	Fred was Bob's PhD supervisor.
BG	Bob and Greg worked together on a successful project.
BI	Irene and Bob are in the same university swimming team.
BZ	Zack and Bob co-authored some papers.
HZ	Hank and Zack co-organized a workshop.

Fig. 1. The interactions observable by both Alice and Bob and Alice's observations

$\mathfrak{T}'$ if its coverage is greater: $\mathbf{c}(\mathfrak{T}) \supseteq \mathbf{c}(\mathfrak{T}')$. We write this $\mathfrak{T} \succeq \mathfrak{T}'$. The overall trust alignment $\mathfrak{T}^*$ between two agents can now be found by finding a minimally general generalization, which covers all the communicated rules. It is therefore a set of rules of the form $\langle \Gamma^*, \Delta^*, \Psi^* \rangle$, such that for any targeted alignment $\langle \gamma, \delta, \psi \rangle$ there is a rule in $\mathfrak{T}^*$ which θ-subsumes it.

4 Implementing the model

The formal framework outlined in the previous section is the roadmap we use to guide an implementation.This implementation must focus on the same three points as before. We will need to describe a robust language for $\mathcal{L}_{Domain}$ and a sufficiently expressive syntax for $\mathcal{L}_{Trust}$. These trust evaluations must be generated from observations with a different trust model for all agents. Lastly we must develop a process for finding the alignment based on inductive learning. As a proof of concept we used a simple scenario described below and focused on displaying the functionality, rather than computational limitations of the approach. There are heuristics we can use to optimize its response time, but this implementation is set up to show that automation of the mathematical framework is a real possibility.

4.1 Finding a Keynote Speaker

Alice is organizing a conference and needs to invite a keynote speaker. She assigns the task of finding this person to her personal computational agent. It must contact the other agents in the system. Bob's agent recommends Zack. However, Alice and Bob's agents have never aligned their models and therefore Alice's

agent does not know how to interpret this gossip. It asks Bob's agent to start the alignment process.

We give the network of interactions in Figure 1. It is a fairly small network, so as not to lose the oversight. The table in Figure 1 gives high level descriptions of Alice's observations of the interactions, which are stored in her agent's belief base. This description can, fairly easily, be interpreted in an actual modal logic for BDI-agents, but we opt for readability, rather than formality here.

4.2 A communication language

In Section 3 we argued that to align agents need 2 languages in which to communicate. We will start with a description of $\mathcal{L}_{Trust}$. This will be a very simple language consisting of two predicates: $trustworthy(X)$ and $untrustworthy(X)$. This obviously glosses over the complexity of trust, but even with such simple predicates, we can give different semantics for the concepts to the separate agents.

Secondly we need to have a language in which to describe the observations. Firstly we need to distinguish between objective and subjective observations. From now on we will call the objective "observations" *facts*, while reserving *observation* for just the subjective ones. We want the agents to be able to communicate about the facts underlying a trust evaluation. We will rely on the restrictions of a language, $\mathcal{L}_{Domain}$, to limit the communication to shared, objective facts and not the subjective observations.

In our example Alice is searching for a keynote speaker. The environment is comprised of a diverse set of interactions. Both academic evaluations and personal relations between the scientists play a role in the trust the agents put in each other, so this must be reflected in any language suitable for them to communicate about this. We keep it simple and define the language as a simple ontology for interactions, as seen in Figure 2. Each property of an object is either objective, or can be objectified by using a shared benchmark, such as the impact factor of an article: this can be measured by a common standard, for instance the citation index. We note that these objective descriptions are easily locked down in an ontology and are the sort of definitions that are usually already fixed in available ontologies for agent domains.

4.3 Prolog and Aleph

Alice bases her evaluations of a keynote speaker on academic qualities only, while Bob also takes personal qualities into account. Both of the models will be represented as Prolog programs, rather than using a specific trust modeling methodology, which would allow for more complex models than we wish to align in this initial approach. Alice has three reasons to evaluate an agent as a trustworthy keynote speaker. Firstly they have published a good article together, which we objectively describe as having a high impact factor. Alternatively she attended a good lecture, given by that person. This is objectified by the average students' evaluation. Finally, if a trustworthy person published a good article

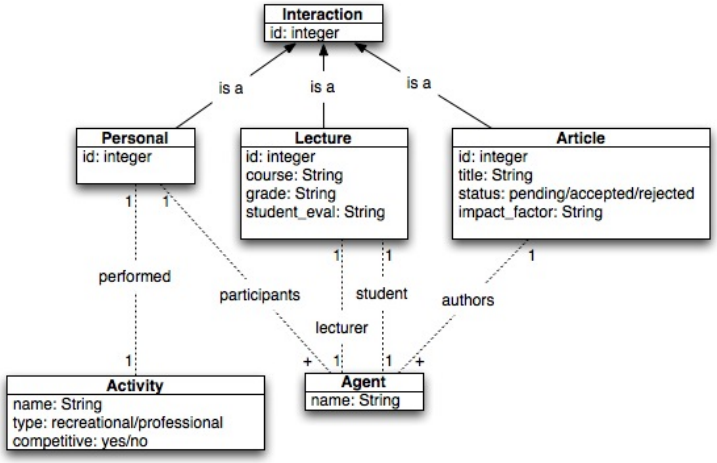


Fig. 2. The ontology for $\mathcal{L}_{Domain}$, in a UML-like representation

with a third person, that third person is also trusted. Bob has different reasons to trust an agent as a keynote speaker, based more on personal observations. He also trusts someone if they published together, but his criteria of a good article is that it was not rejected by the journal. For attended lectures it is a similar situation. The student evaluation does not play a role in his evaluation. Finally, he trusts a person based on its ability to entertain, which he evaluates through interactions on a recreational basis.

We specify the trust models for the agents representing Alice and Bob in the following table and use their observed interactions to calculate the trust evaluations they will align on.

Alice	**Bob**

`trustworthy(X) ← article(I), authors(I, List), member(X, List), member(alice, List), impact_factor(I, high)`
`trustworthy(X) ← lectured(I), lecturer(I, X), student(I, alice), ¬ student_evaluation(I, bad)`
`trustworthy(X) ← article(I), authors(I, List), member(X, List), member(Y, List), trustworthy(Y)`
	`trustworthy(X) ← article(I), authors(I, List), member(X, List),member(bob, List), ¬ status(I, rejected)`
	`trustworthy(X) ← lectured(I), lecturer(I, X), student(I, bob)`
	`trustworthy(X) ← personal(I), participants(I, List), member(X, List), member(bob, List), activity(I, Act), type(Act, recreational)`

Both agents also have the rule that if a target agent is not `trustworthy` then he is `untrustworthy`.

To align these trust models, the agents need to share a set of interactions. The initial setup contains this set of shared interactions as well as each agent's observations thereof. Both agents observe only the shared facts of the interactions and there are no subjective observations. The alignment process starts with Bob's agent sending gossip messages to Alice's, regarding all other agents in the system. An example of such a message is:

`gossip(fred, trustworthy(fred), lectured(BF) ∧ lecturer(BF, fred) ∧ student(BF, bob))`

These messages allow Alice's agent to form the targeted alignments by computing the own trust based on the interactions pinpointed in the gossip message. The

targeted alignments have this trust evaluation as the head of the rule and the gossip message in the body.

```
untrustworthy(fred) ←  trustworthy_bob(fred), lectured(BF),
                       lecturer(BF, fred),
                       student(BF, bob)
```

Learning as search. Alice's agent can form a trust alignment with Bob's agent by generalising from targeted alignments such as above. We look at this as the problem of finding a hypothesis that covers the targeted alignments. This is considered a search problem through the "hypothesis space". We use Aleph [9], an implementation of the Progol algorithm [10] to perform this "search". It searches for sets of Horn clauses which cover the examples, but requires us to give some basic information about the boundaries of the search space: which predicates it should learn to put in the head of the clause and which predicates it can use in the body of the clauses. In our example all this information is available: we want the trust evaluation in the head and the predicates in the gossip in the body. The main drawback of the algorithm is that it can only learn *two-valued* concepts. For our example we have a trust model that is two-valued, but in most models currently in use this is not the case. In the case of discrete-value trust models the algorithm could learn each value separately. In the case of continuous-value trust models it would require some pre-processing to be able to use an ILP algorithm. For our example, however, a search for two-valued concepts is all we need. Even in this case, though, we need to reformulate the problem. What we want to find are alignment rules, which may not be a binary concept. We know that Bob's trust model *is* two-valued. We therefore use this algorithm to learn Bob's trust model, based on the gossip.

The algorithm attempts to learn a hypothesis that covers all *positive* examples and excludes all *negative* examples. For us a positive example is an agent that is *trustworthy*, while being *untrustworthy* is obviously a negative example for this concept. In our scenario, Charlie, Hank and Eve are untrustworthy and thus negative examples for the predicate we are trying to learn.

The algorithm performs a heuristic search of the hypotheses and gives us the minimally general generalization.

4.4 Results

For our example, Aleph found the following trust model for Bob:

```
trustworthy(fred)
trustworthy(greg)
trustworthy(X) ← personal(I), participants(I, List),  member(X, List),
                 activity(I, Act), type(Act, recreational)
```

The first thing we notice is that the trustworthiness of Fred and Greg are given as facts. This is because there are not enough examples to learn further rules. While Aleph can generalize the rules, the hypotheses generated do not cover any further examples. Its best solution is therefore the plain fact. We note therefore that to learn anything sensible we need more examples. By adding more agents and interactions, we obtain:

```
trustworthy(X) ← article(I), author(I, List), member(X, List),
                 impact_factor(I, high)
trustworthy(X) ← lectured(I), lecturer(I, X)
trustworthy(X) ← personal(I), participants(I, List),  member(X, List),
                 activity(I, Act), type(Act, recreational)
```

This is a better approximation of Bob's trust model. We still see some notable differences. Firstly the clause that Bob needs to be a member of the interactions has been dropped: all the interactions taken into account had Bob as a member and there were no negative examples where the same held and Bob was *not* a member. The same happens for taking the positive predicate `impact_factor(I, high)` rather than the negation ¬`status(I, rejected)`. Once again, due to a lack of examples. This, however, is completely within the expectations of induction. We can never know for sure our alignment is complete; all we can do is find the best approximation given the data we have. Now that we have an approximation of Bob's trust model, we can use this as a predictive model. If Bob's agent gossips to Alice's that it trusts Zack, based on interaction `article(BZ)`, Alice's agent can trace the model to find that the first rule in the approximated model covers that. It can compare that with Alice's own model and find that they are very similar. The reliability of this gossip is high. If, however, it had been based on a different, *personal*, interaction and used the third rule in the approximated model, then she would be able to find few similarities to her own model and conclude a low reliability. We see, even in such a simple example, the significance of this approach: whereas in both cases Bob's agent gossips that Zack is trustworthy, Alice's agent can distinguish between the two situations.

This comparison between trust models is a fairly straightforward comparison process. There are many algorithms, using various metrics to measure the distance between two programs. We can use the same algorithms for calculating the distance between two program fragments. If the distance is large, then the trust models are dissimilar for the given interactions and the reliability is low. If the distance is small, then the models are similar and communication is reliable. In our example, using a lexical comparison is enough to give a distance measure: in the situation where Bob's trust is based on co-authoring an article, the distance between the approximation and Alice's model is smaller than in the case of a personal interaction. In more descriptive trust models, we propose using more sophisticated methods, such as the one developed by Lukacsy et al. [11].

5 Conclusion and Future Work

We have argued that for agents to understand communication about trust, the agents need an understanding of what observations the sender bases his gossip on. In Section 3 we outlined a mathematical framework for this purpose, which relies on 3 things:

- a language to talk about trust
- a language to talk about objective facts of interactions
- an algorithm to model predicates in the former based on the latter

In Section 4 we have presented a proof of concept for such a model. The trust language was left mostly out of the picture, but ongoing work on ontologies,

as mentioned in Section 2 could be used for this. We are mainly interested in developing useful algorithms to align the underlying concepts, based on communication about interactions. These go hand in hand: if our $\mathcal{L}_{Domain}$ gets more or less descriptive, different algorithms may be necessary for aligning the trust evaluations through it. Our initial implementation works with a very basic $\mathcal{L}_{Domain}$ and a naive use of a learning algorithm, but it shows the approach works. Future work will focus on finding sensible heuristic rules to apply the algorithm in a larger and more realistic environment. The framework itself also needs extending to allow for situations where agents can have multiple roles and interpret trust differently per role. Our framework also does not yet take dishonesty in the gossip into account. However, this model allows for agents with diverse trust models to gossip reliably about them and future progress can build on the framework.

Acknowledgements This work is supported by the Generalitat de Catalunya under the grant *2009-SGR-1434*, the Agreement Technologies Project *CONSOLIDER CSD2007-0022, INGENIO 2010* and the LiquidPub Project *CIT5-028575-STP*.

References

1. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review **19**(1) (2004) 1–25
2. Omidyar, P.: Ebay. http://www.ebay.com, retrieved September 26, 2008 (1995)
3. Sabater-Mir, J., Paolucci, M., Conte, R.: Repage: REPutation and imAGE among limited autonomous partners. JASSS - Journal of Artificial Societies and Social Simulation **9**(2) (2006)
4. Pinyol, I., Sabater-Mir, J.: Arguing about reputation. the lrep language. In: Proceedings of the 8th Annual International Workshop "Engineering Societies in the Agents World" (ESAW'07). Volume 4995. Springer LNCS (2007) 284–299
5. Casare, S., Sichman, J.: Towards a functional ontology of reputation. In: AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, New York, NY, USA, ACM (2005) 505–511
6. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. Proceedings of the 33rd Hawaii International Conference on System Sciences **6** (2000) 4–7
7. Koster, A., Sabater-Mir, J., Schorlemmer, M.: Formalization of the trust and reputation alignment problem. Technical Report TR-2009-03. See also Deliverable 5.1.1 of Agreement Technologies CONSOLIDER-INGENIO 2010, CSIC-IIIA (2009) http://www2.iiia.csic.es/∼andrew/files/techreport.pdf.
8. De Raedt, L.: Logical and Relational Learning. Springer Verlag (2008)
9. Srinivasan, A.: The aleph manual. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/, retrieved February 9, 2009 (June 2004)
10. Muggleton, S.: Inverse entailment and progol. New Generation Computing Journal **13** (1995) 245–286
11. Lukacsy, G., Szeredi, P.: Plagiarism detection in source programs using structural similarities. Acta Cybernetica (In Press)