An Interaction-oriented Model of Trust Alignment *

Andrew Koster and Marco Schorlemmer

IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Bellaterra, Spain
{andrew, 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 an 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. These activities may expose them to risks if they choose the wrong agent to partner with for any such activity. One proposed method of selecting the right partner is based on the concept of trust and reputation to create a network of social control for the agents. There are already quite a large number of computational models for trust and reputation [2] in use; each with a slightly different interpretation on what trust means. This contrasts with one of the main reasons for using a trust based approach: it is easily communicable and agents can warn each other for fraudulent agents or can help each other in their selection of a good partner. If the different agents use diverse models of trust, this communication becomes problematic. What does it mean to one agent when another agent communicates a trust evaluation?

Lets consider this in an example: *Alice wants to know if Dave would be a good keynote speaker for the conference she is organizing. However, she does not know enough about him. She asks Bob. Bob has never collaborated with Dave directly, but they work at the same institute and play squash together. Alice has a trust model which takes the specific roles agents are involved in into account. Bob does not and therefore his trust in Dave is universal, as a collegue, a squash*

* Part of this article was published at CCIA '09 [1]

player and also as a keynote speaker. If Bob therefore answers that he trusts Dave, what does this mean to Alice?

This simple example shows that it is important for Alice to know *why* Bob trusts Dave. Trust cannot be seen independent from the observations which support that trust. In this case we considered two different computational models: one simple one which ignores roles and one slightly more complicated which takes the roles agents play in the environment into account. However we argue that even if the agents do use the same computational model, if that model is sufficiently expressive and based on cognitive principles [3, 4], the goals agents have and the way they observe their environment will still lead to different interpretations of what trust means. Various models allow for the specification of parameters such as context-dependence, preference for risk-avoiding or risk-taking behaviour, etc. While the models are the same, agents with different values for these parameters will have different trust evaluations given exactly the same information. Thus even using the same model trust can mean different things for different agents.

This article discusses the problem of aligning trust models. In Section 3 we present a mathematical framework to describe the problem and in Section 4 we present a prototype implementation to align trust models. First we will discuss some related work.

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 [5, 6], however in practice these ontologies do not have the support of many of the different trust methodologies in development. An ontology alignment service is presented in [7], but it requires a translation of all specific trust model ontologies into a general ontology. In addition, as we argued in the introduction, even if support were added for all systems and a common ontology emerged, a cognitive agent will still have its own interpretation of the world on which it bases its trust evaluations: thus trust must always be considered in the light of *why* agents trust each other. This is based on interactions they have had, or are told about and thus can already be talked about for most domains in the domain-specific ontologies. We wish to consider an adaptable trust alignment method which builds on top of the domain ontologies.

Abdul-Rahman and Hailes' reputation model [8] 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 dif-

ferentiate 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, based on the gossip they send each other. The gossip specifies certain interactions, which each agent observes differently. These observations form the support for an agent's trust evaluation. If another agent communicates this trust evaluation, the interpretation should be based on the underlying interactions. An alignment of the trust models gives a way of doing this by gossiping about the agents' trust evaluations and the observations (and thus interactions) they base these on.

3 Theoretical Foundations

Before we consider possible solutions we need a clear definition of the problem we are considering. Firstly we are considering 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 need to define a language. Finally, the agents need to have some method of forming an alignment based on the statements in this language. Throughout this section we will illustrate the main definitions with an example. We will first give the basic scenario, which is a modified version of the example in the introduction:

> 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. The agent's first choice is Bob. It sends Bob's agent a message with an invitation to the conference, but he can't make it. Instead his agent recommends Zack. However, Alice and Bob's agents have never aligned their models and therefore Alice's agent doesn't know how to assess the reliability of this gossip. It asks Bob's agent to start the alignment process. There are various other people who both Bob and Alice have interacted with. Their agents contain knowledge about this and they gossip about these to form the alignment.

3.1 A formal representation of trust models

As argued in Section 2, interactions form the building blocks for talking about trust. While these interactions have a wealth of properties, we will start with the bare minimum we need to know to start the trust alignment. We define an interaction just as the set of observing agents. In practice we will use a more descriptive interpretation of what an interaction is. This we will introduce in Section 3.2.

We will denote with $\mathfrak{I}$ the set of all interactions in the environment and with Ag the set of all agents. $\mathfrak{I}_{|A} \equiv \{\langle id, Ag\rangle \in \mathfrak{I} \mid A \in Ag\}$ is the set of interactions observed by agent A; id is a unique identifer for an interaction. Agent A's observations of these interactions form a separate set.

Definition 1 (Observation). *An agent A's observation is given by the function $observe_A : \mathfrak{I}_{|A} \rightarrow \mathfrak{O}_A$, which associates each interaction $i \in \mathfrak{I}_{|A}$ with an observation $o \in \mathfrak{O}_A$. The set $\mathfrak{O}_A$ is the entire set of observations of agent A and is in the form of the agent's internal representation.*

Example: Alice's agent (Observations). *Alice's agent stores the observations in its belief base. We give an example observation of an interaction where Bob and Dave play racquetball together.*
The observe-function simply maps the interaction itself, with id BD*, onto a belief: $observe_{Alice}(BD) = (play\_racquetball(Bob, Dave))$.*

Because we will generally work with sets of observations, it is useful to extend the function $observe_A : \mathfrak{I}_{|A} \rightarrow \mathfrak{O}_A$ to $Observe_A : \mathcal{P}(\mathfrak{I}_{|A}) \rightarrow \mathcal{P}(\mathfrak{O}_A)$ such that for $I \subseteq \mathfrak{I}_{|A}$: $Observe_A(I) = \{o \in \mathfrak{O}_A | \exists i \in I : observe_A(i) = o\}$.

An agent A's observations *support* some trust evaluation. This is the essence of the trust model. As we argued in Section 2 there are many different computational trust models, but all compute trust evaluations, based on observations of interactions. We say these observations support the trust evaluation. These trust evaluations are statements in a language $\mathcal{L}_{Trust}$. The agents share the syntax of this language, but each agent can have its own semantics, defined by its trust model. While this shared syntax is not strictly necessary it makes the framework more comprehensible. Because the semantics will differ anyway, whether an agent calls reputation "reputation" or "jackhammer" doesn't really matter from a conceptual point of view, each agent will have their own semantics and will have to distinguish between the syntax used by each agent anyway. It makes things easier on a computational level, however, if we can define a language in which agents agree on the syntax. even though they need to align the semantics. It also makes it easier to understand for humans, who will ultimately be directing agents equipped with the system. Thus this assumption may be just as important from the perspective of making the framework "explicable".

$\mathcal{L}_{Trust}$ is a standard predicate language, with one restriction: because trust is always about some target agent (all trust predicates have an object), we will only consider those predicates which give an evaluation of exactly one such target T:

$$\mathcal{L}_{Trust}[T] \equiv \{\varphi \in \mathcal{L}_{Trust} \mid \text{all predicates in } \varphi \text{ have target T}\}$$

We can now give a very abstract model of trust, which we can use as a basis for communication. We ground this framework in a mathematical model of information flow, introduced by Barwise & Seligman [9]. This is a very general model of how information flows and has been shown to be a good foundation for alignment [10]. We use this same framework to formalize our earlier assertion that a trust model gives an evaluation of target agents *supported* by the agent's observations.

Definition 2 (Trust model). *A trust model $\mathcal{M}_A$ of agent A is given as the tuple $\langle \mathcal{P}(\mathfrak{O}_A), Evals_A, \models_A \rangle$, with the following definitions:*

- *$\mathcal{P}(\mathfrak{O}_A)$ the power set of the observations of agent A.*
- *$Evals_A \subseteq \mathcal{L}_{Trust}$ the trust evaluations of agent A.*
- *$\models_A$ a binary relation: $\models_A \subseteq \mathcal{P}(\mathfrak{O}_A) \times Evals_A$, such that if $O \subseteq \mathfrak{O}_A$ (in other words, $O \in \mathcal{P}(\mathfrak{O}_A)$) and $\varphi \in Evals_A$, then $O \models_A \varphi$ represents that for agent A, the trust evaluation φ is supported by O.*

Example: Alice's agent (Alice's trust model). *Alice's agent's beliefbase contains the following observations:*

$\{play\_racquetball(Bob, Dave),\ co\_author(Alice, Dave)\}$

and $\mathcal{L}_{Trust}$ is the set of predicates formed by the predicate $trustworthy(T)$ and its negation, with $T \in \{Alice, Bob, Dave\}$. Some examples of support relations

its trust model can have:
$\{co\_author(Alice, Dave)\} \models_{Alice} trustworthy(Dave)$
$\{play\_racquetball(Bob, Dave\} \models_{Alice} \neg trustworthy(Dave)$

This framework models the entire space of potential observations and their supported trust evaluations. In practice only one of these evaluations will be the actual evaluation of the target, namely the one supported by the observations in the actual state of the agent. However, to assess gossip it is important to base this on a larger amount of data than just the real trust evaluation.

We will now use this formal model to define a channel, following the mathematical model put forth in [9]. The trust alignment between two agents is grounded in this channel, which models the relation of two different trust models.

3.2 Formalizing gossip

Now that we have described trust models algebraically, we can focus on the formal model of how to assess each others' gossip. To do so, the agents should form an alignment and for that the agents A and B establish some set of shared interactions $\mathfrak{I}_{|AB} = \mathfrak{I}_{|A} \cap \mathfrak{I}_{|B}$, the set of interactions they have both observed. This results in subsets of observations for both agents: $\mathfrak{O}_{A|B} \subseteq \mathfrak{O}_A$ are A's observations of the interactions it shared with B. These are observations based on interactions they know they've shared. Due to the assumption that all the observers know the other observers in interactions, each agent can find this set. We justify this assumption by noting that even if this is not the case a priori, the set of shared interactions is easy to establish by first communicating about these, before entering into the alignment process.

The agents can now talk about their trust evaluations based *only* on the interactions they *both* observed to come to their evaluations. To talk about trust, they can use $\mathcal{L}_{Trust}$, but we have so far not specified how to talk about the interactions or observations. To do so, we introduce another, separate, language: $\mathcal{L}_{Domain}$. This should be a domain dependent language in which the agents can choose to relay objective properties of the interactions they have used to form their trust evaluations.

Example: Alice's agent ($\mathcal{L}_{Domain}$). *An example of a domain language in which agents can talk about interactions is the one given in Figure 1. Playing racquetball can be modeled as an activity which is performed in a personal interaction. The other interactions that can be talked about are of a more professional nature.*

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 sent by an agent B is defined as the tuple $\langle T, \beta, \psi \rangle$, with $\beta \in Evals_B[T]$ and $\psi \in \mathcal{L}_{Domain}$. Just as $Evals_B$ is defined as

a subset of $\mathcal{L}_{Trust}$, we define $Evals_B[T] \subseteq \mathcal{L}_{Trust}[T]$. Thus gossip between two agents consists of a part about trust and a part about the agents' observations which support this trust predicate. This ψ can be used to pinpoint what interactions comprise I, the set of interactions that B used to compute β. Agent A uses I to find its own trust evaluation $\alpha \in Evals_A[T]$, such that $Observe_A(I) \models_A \alpha$, which gives us the basis for a targeted combined trust model: a shared set of interactions I supporting a trust evaluation for both A and B with regards to target T. It is possible that in some situations $Observe_A(I)$ will not support any trust evaluation for agent A. In this case we cannot use the related gossip in forming the targeted alignment. However, due to the requirement that the agents only take shared interactions into account this should not happen often.

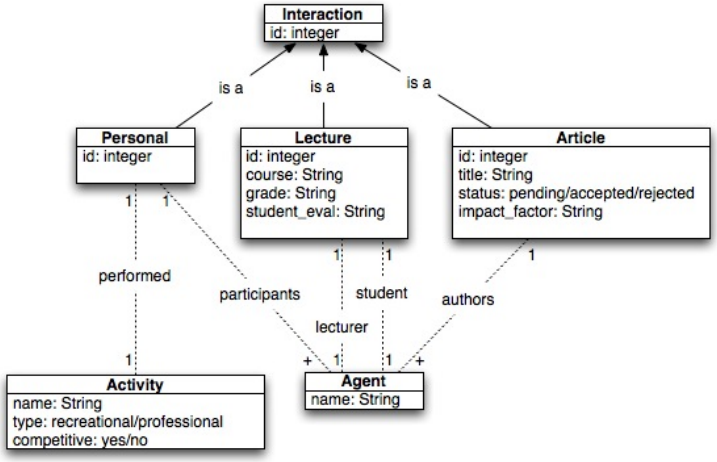


Fig. 1. The ontology for an example $\mathcal{L}_{Domain}$, in a UML-like representation

Definition 3 (Targeted combined trust models). *A targeted combined trust model has the same structure as a trust model, defined in Definition 2. It is the tuple* $\langle \mathcal{P}(\mathfrak{I}_{|AB}), (Evals_A[T] \cup \mathcal{L}_{Domain} \times Evals_B[T] \cup \mathcal{L}_{Domain}), \models_{AB} \rangle$
with $\models_{AB}$ *a binary relation:*

$\models_{AB} \subseteq \mathcal{P}(\mathfrak{I}_{|AB}) \times ((Evals_A[T] \cup \mathcal{L}_{Domain}) \times (Evals_B[T] \cup \mathcal{L}_{Domain}))$*, such that if* $I \subseteq \mathfrak{I}_{|AB}$ *and* $\langle \alpha, \beta \rangle \in (Evals_A[T] \cup \mathcal{L}_{Domain} \times Evals_B[T] \cup \mathcal{L}_{Domain})$*, then* $I \models_{AB} \langle \alpha, \beta \rangle$ *if and only if* $Observe_A(I) \models_A \alpha$ *and* $Observe_B(I) \models_B \beta$*.*

Example: Alice's agent (Combined trust model with regards to Dave). *We recall the interaction BD and Alice's observation* $observe_{Alice}(BD)=play\_racquetball(Bob,Dave)$*.*
Furthermore, Alice's trust model contains the relation: $\{play\_racquetball(Bob,Dave)\} \models_{Alice} \neg trustworthy(Dave)$*. Lets assume Bob has a similar observation:* $observe_{Bob}(BD)=good\_racquetball\_match(Bob,Dave)$ *and the relation:* $\{good\_racquetball\_match(Bob,Dave)\} \models_{Bob} trustworthy(Dave)$

Then the combined trust model for Dave contains the relation (with the agent's names in subscript for clarity):

$BD \models_{Alice,Bob} \langle \neg trustworthy_{Alice}(Dave),\ trustworthy_{Bob}(Dave) \rangle$.

Additionally, the model could contain further information communicated by Bob's agent about the interaction in $\mathcal{L}_{Domain}$. Thus the relation could look like:

$BD \models_{Alice,Bob} \langle \neg trustworthy_{Alice}(Dave),\ trustworthy_{Bob}(Dave)$
$\wedge\ personal(BD)\ \wedge\ activity(BD,racquetball)\ \wedge\ participants(BD,\ (Bob,Dave)) \rangle$

Note that neither agent knows everything, just those parts of the model which are gossiped about. From these partial models they must extrapolate the underlying model, so as to form an alignment. The rules we have such that $I \models_{AB} \langle \alpha, \beta \rangle$ in the targeted combined trust models form the basic building blocks of this alignment.

Definition 4 (Targeted alignment). *Given a combined trust model for agents A and B with regards to target T, we define $\Rightarrow_A^T$ as a binary relation $\Rightarrow_A^T \subseteq Evals_B[T] \cup \mathcal{L}_{Domain} \times Evals_A[T] \cup \mathcal{L}_{Domain}$, with $\Gamma[T] \Rightarrow_A^T \Delta[T]$ such that:*

$\Gamma[T] \subseteq Evals_B[T] \cup \mathcal{L}_{Domain}$ — *where $\Gamma[T]$ means all trust predicates in Γ have target T*

$\Delta[T] \subseteq Evals_A[T] \cup \mathcal{L}_{Domain}$

$\exists I \subseteq \mathfrak{I}_{|AB} : \forall \gamma \in \Gamma[T] : \exists \delta \in \Delta[T] :\quad I \models_{AB} \langle \gamma, \delta \rangle$

$\mathfrak{T}_A[T] = \{\langle \Gamma[T], \Delta[T] \rangle \mid \Gamma[T] \Rightarrow_A^T \Delta[T]\}$ is agent A's targeted alignment with target T. We call $\Gamma[T] \Rightarrow_A^T \Delta[T]$ a rule in this targeted alignment . The relation $\Rightarrow_A^T$ is not symmetrical, while the combined trust model is. It therefore stands to reason there is a similar targeted alignment $\mathfrak{T}_B[T]$ with its binary relation $\Rightarrow_B^T$.

A targeted alignment can be interpreted as the set of relations between an agent's own model and the communication partner's model with regards to some specific target. Each rule states that if there is a set of interactions I which support all trust evaluations by agent B as well as all statements in $\mathcal{L}_{Domain}$ about the interactions, then agent A has a trust evaluation with corresponding statements in $\mathcal{L}_{Domain}$ which is also supported by I. We consider agents gossiping untruthful information as outside the scope of this work and suppose that any rule in a targeted alignment is a "true" rule.

Example: Alice's agent (Aligning about Dave). *The rule in the alignment with regards to Dave, based on the relation in above would be:*

$personal(BD),\ activity(BD,racquetball),\ participants(BD,(Bob,Dave)) \wedge trustworthy_{Bob}(Dave)$
$\Rightarrow_{Alice}^{Dave} \neg trustworthy_{Alice}(Dave)$

3.3 Generalization and coverage

Now that we have a way of describing the relationship between two agents' trust models with regards to a specific target, we wish to expand this idea to

encompass multiple targets. We consider this problem as an inductive learning problem [11]. 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, we need to define what our solution should look like. This is an untargeted alignment: similar to the targeted alignment in Definition 4, but not restricted to just one target. A natural way of forming an untargeted alignment is by simply replacing all instances of the target agent in a targeted alignment with a free variable. In general we will say an untargeted alignment is a θ-subsumption of one or more targeted alignments. We introduce the notion of coverage to specify which targeted alignments these are.

Definition 5 (Coverage of alignments). *For an agent A, we say an alignment $\mathfrak{T}_A$ covers a targeted alignment $\mathfrak{T}_A[T]$, if for every rule $\Gamma[T] \Rightarrow^T_A \Delta[T] \in \mathfrak{T}_A[T]$, there is a rule $\Gamma \Rightarrow_A \Delta \in \mathfrak{T}_A$, such that Γ θ-subsumes $\Gamma[T]$ and Δ θ-subsumes $\Delta[T]$ for some θ. We introduce the function* $\mathbf{c}$ *which returns the set of targeted alignments* covered *by a given untargeted alignment.*

We can now use inductive learning to find a trust alignment that covers all the targeted alignments. The way to do this is by structuring the search space. We do this with the generality relationship.

Definition 6 (Generality relation). *We say an alignment $\mathfrak{T}$ is more general than an alignment $\mathfrak{T}'$ iff $\mathbf{c}(\mathfrak{T}) \supseteq \mathbf{c}(\mathfrak{T}')$. We write this: $\mathfrak{T} \succeq \mathfrak{T}'$. If $\mathbf{c}(\mathfrak{T}) \supset \mathbf{c}(\mathfrak{T}')$ we say $\mathfrak{T}$ is strictly more general and write $\mathfrak{T} \succ \mathfrak{T}'$*

The overall trust alignment between two agents can now be found by finding a minimally general generalization, which covers all targeted alignments.

Definition 7 (General trust alignment). *The trust alignment $\mathfrak{T}^*_A$ of an agent A with another agent is a* minimally general generalization *of all the targeted alignments: $\forall T \in Targets : \mathfrak{T}_A[T] \in \mathbf{c}(\mathfrak{T}^*_A)$. A minimally general generalization means, that if there is any other alignment $\mathfrak{T}'_A$ that covers all targeted alignments, then: $\mathfrak{T}'_A \succeq \mathfrak{T}^*_A$.*

Example: Alice's agent (Trust alignment with Bob). *If Alice and Bob only gossip about Dave, with the targeted alignment above as a result, the general alignment could look something like this:*

$$personal(I) \wedge activity(I,racquetball) \wedge participants(I, (X,Z)) \wedge trustworthy_{Bob}(X) \Rightarrow \neg trustworthy_{Alice}(X)$$

This example has necessarily been very simplistic and therefore this result seems trivial. We base this alignment on only one interaction about one single agent. The generalization in this case is just the skolemnization of the targeted alignment. From this we learn that if Bob bases his evaluation "trustworthy" of a target agent on an interaction where they played racquetball together, Alice should consider this agent as "¬trustworthy". While simple, this is a good start: next time Bob recommends a possible keynote speaker based on his racquetball games, Alice knows that she should take this to mean the opposite. However, if

Bob recommends someone based on their joint experience in authoring papers, this rule says nothing about this. The alignment is not yet complete and there is no rule covering this type of interaction.

We will now discuss a prototype implementation of the alignment process, which will illustrate the computational model of this framework.

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 on the 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.

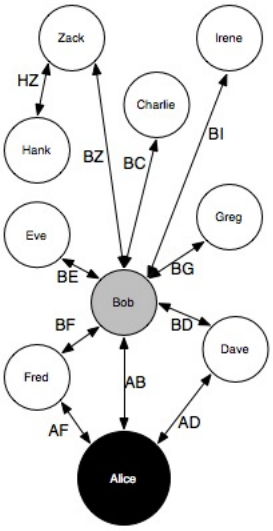
4.1 Finding a Keynote Speaker

We use the same example as in the previous section, but will use more realistic trust models as well as a small network of interactions. This network is given in Figure 2. It is a fairly small network, so as not to lose the oversight. The table in Figure 2 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.



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. 2. The interactions observable by both Alice and Bob and Alice's 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, which we have already seen in Figure 1. 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 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 [12], an implementation of the Progol algorithm [13] 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. [14].

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*. We'd like to thank Jordi Sabater-Mir for participating in this research.

References

1. Koster, A., Sabater-Mir, J., Schorlemmer, M.: A formalization of trust alignment. In: Twelfth International Congress of the Catalan Association of Artificial Intelligence (CCIA 2009), Cardona, Spain, IOS Press (2009)
2. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review **19**(1) (2004) 1–25
3. Falcone, R., Castelfranchi, C.: Social Trust: A Cognitive Approach. In: Trust and Deception in Virtual Societies. Kluwer Academic Publishers (2001) 55–90
4. Hübner, J.F., Lorini, E., Herzig, A., Vercouter, L.: From cognitive trust theories to computational trust. In: Proceedings of the 12th International Workshop on Trust in Agent Societies, Budapest, Hungary, 10/05/2009-11/05/2009. (In Press)
5. 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
6. 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
7. Nardin, L.G., Brandão, A.A.F., Muller, G., Sichman, J.S.: Effects of expressiveness and heterogeneity of reputation models in the art-testbed: Some preliminar experiments using the soari architecture. In: Proceedings of the Workshop "Trust in Agent Societies" at AAMAS '09, Budapest, Hungary (2009)
8. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. Proceedings of the 33rd Hawaii International Conference on System Sciences **6** (2000) 4–7
9. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press (1997)
10. Schorlemmer, M., Kalfoglou, Y., Atencia, M.: A formal foundation for ontology-alignment interaction models. International Journal on Semantic Web and Information Systems **3**(2) (2007) 50–68
11. De Raedt, L.: Logical and Relational Learning. Springer Verlag (2008)
12. Srinivasan, A.: The aleph manual. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/, retrieved February 9, 2009 (June 2004)
13. Muggleton, S.: Inverse entailment and progol. New Generation Computing Journal **13** (1995) 245–286
14. Lukácsy, G., Szeredi, P.: Plagiarism detection in source programs using structural similarities. Acta Cybernetica **19**(1) (2009) 191–216