Personalizing Communication about Trust

Andrew Koster
IIIA-CSIC,
Universitat Autònoma de Barcelona
Spain
andrew@iiia.csic.es

Jordi Sabater-Mir
IIIA-CSIC
Spain
jsabater@iiia.csic.es

Marco Schorlemmer
IIIA-CSIC,
Universitat Autònoma de Barcelona
Spain
marco@iiia.csic.es

ABSTRACT

Agents in open multi-agent systems must deal with the difficult problem of selecting interaction partners in the face of uncertainty about their behaviour. This is especially problematic if they have to interact with an agent they have not interacted with before. In this case they can turn to their peers for information about this potential partner. However, in scenarios where agents may be evaluated according to many different criteria for many different purposes, their peers' evaluations may be mismatched with regards to their own expectations. In this paper we present a novel method, using an argumentation framework, that allows agents to discuss and adapt their trust model. This allows agents to provide, and receive, personalized trust evaluations, better suited to the agent in need, as is shown in a prototypical experiment.

Categories and Subject Descriptors

Computing Methodologies [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Experimentation

Keywords

Trust, reliability and reputation; Argumentation

1. INTRODUCTION

In any society of intentional agents, trust is an essential tool for selecting interaction partners. Trust is a personal and subjective evaluation of a target for the fulfillment of a specific goal. However, to choose a partner based on trust, an agent needs information about it. In any environment where it does not have direct experiences with interaction partners, it turns to external sources to aid in making this selection. Reputation is one source of such information. Reputation is what a group of individuals say about an agent, regarding its behaviour. Computational models of reputation usually obtain this by aggregating reported evaluations from a large number of individual agents. If the aggregation is performed properly, this can be an effective estimate of an agent's trustworthiness, precisely because it is an aggregation of a large number of reported evaluations.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1.1 Reputation and Recommendation

The drawback of reputation that we focus on in this article is that, if an agent has specific requirements, reputation calculated as an aggregation of opinions can be an inadequate measure to decide whether or not a target is able to meet these requirements. This is especially true in any domain where there are many different requirements an agent may have for any specific task. A good reputation for fulfilling that task does not guarantee that the agent complies with a single agent's specific requirements, just that it is generally able to comply with agents' needs.

For tasks such as buying an item in an online marketplace these measures are generally good enough: the range of requirements between agents does not vary much, and a good reputation generally means an agent is good over the entire range of requirements. However, the same cannot be said when the range of requirements increases. When a choice depends on many different criteria, collaborative filtering mechanisms may offer a solution. These mechanisms provide personalized recommendations by relying on large numbers of agents and matching the requesting agent's profile to that of agents who have provided evaluations [17]. These recommendations are more tailored to an individual's needs than reputation, because the system only uses evaluations from agents similar to the requesting agent. Such mechanisms, however, have their own drawbacks. The first is that they require a large number of agents providing not only evaluations, but also a profile (representing the context in which they made the evaluation and the goal they were trying to achieve).

As multi-agent systems mature and gain in popularity, the domains in which they may be applied increase, and not all of these have a large number of agents providing profiles and evaluations. An approach that does not rely on a large network is to use an agent's own network of friends, in order to give personalized recommendations. This is the approach we adopt in this paper.

1.2 Personalized Trust Recommendations

We assume an agent has a computational trust model to aid with partner selection. Ultimately, computational trust models and recommender systems have a very similar aim. Both computational trust models and recommender systems aim to provide accurate evaluations of other agents in order for the agent to select a good partner, in the given context, to achieve its goal.

The fundamental difference is what agent the calculation centers on. Recommender systems aim to tailor a personalized recommendation to the requesting agent. Trust models, on the other hand, take only the calculating agent's goals and beliefs into account when calculating a trust evaluation. Upon receiving a trust evaluation this needs to be taken into account: it is the sender's subjective and personal evaluation and may, therefore, not be useful to the requesting agent.

In this paper we propose a novel approach to communicate about trust by learning a lesson from recommender systems: we propose to personalize trust evaluations to the requesting agent's requirements. Our proposal is two-pronged. Firstly, agents requesting information may communicate the goal, and corresponding criteria, for which they need a partner to perform a task. The supplying agent can then use this goal and criteria to tailor a trust evaluation to it. Secondly, agents may attempt to persuade each other that their beliefs about the environment, and the corresponding criteria in their trust model, are incorrect. By combining these techniques agents can communicate about personalized trust, allowing them to better estimate a potential partner's performance for fulfilling a specific goal, given the agent's beliefs about the environment.

2. RELATED WORK

Using other agents' trust evaluations directly is not a new idea. A long-standing problem with such communication has been lying or colluding agents. In such cases an agent intentionally communicates a wrong trust evaluation. A solution is to filter out communication from such agents [19]. The underlying assumption is that lying is the only reason other agents' trust evaluations can be mismatched and thus, by detecting agents whose evaluations do not match the own evaluations, the problem can be solved. However, in an environment in which agents may use many different criteria to evaluate each other, these methods will mark many agents as liars who simply have a different opinion.

Koster et al. address this problem by translating others' trust evaluations into the own frame of reference using a machine learning algorithm [8]. However, as with any such algorithm, this requires a large amount of data. In this case the data consists of targets that both agents have evaluated, and can thus be compared. This assumes both agents have already interacted with many agents in the system. Thus it does not work for agents who are new to the environment, or environments with few agents.

Pinyol et al. [13] address the problem of communication in a manner that does not require a large amount of data, by using argumentation about trust. This allows agents to exchange information about their trust model, and thus each agent can decide whether or not to accept a communicated evaluation. While this does not assume agents are lying if their trust evaluations do not match, it suffers a similar drawback to the methods for detecting lies: it will discard a large amount of information if there are many different criteria on which to base an evaluation, because it may only accept or reject a communicated evaluation. We demonstrate this drawback empirically in Section 6. There are other approaches that combine trust and argumentation [10, 6], but these focus on different aspects of the area.

3. ADAPTRUST

Our method for enabling personalized communication about trust is based on three capabilities an agent must have:

1. An agent must be able to adapt its trust model in order to personalize its evaluations to the other agent's needs.
2. An agent must be capable of communicating its criteria for evaluating trust, as well as the underlying beliefs and goals leading to these criteria.
3. An agent must be willing and able to change its trust model, if it is persuaded that its beliefs about the environment, and thus the criteria for calculating trust are wrong.

We assume that agents are willing to adapt their model if they are convinced it is inaccurate. We use AdapTrust to enable this, and, additionally so agents can adapt their trust model to another agent's needs. AdapTrust is an extension of the BDI framework for intelligent agents [16]. As the name implies, AdapTrust allows a trust model to be adapted, according to an agent's goals and beliefs. We present the method in full detail in [9] and summarize it here.

Computational trust models are, fundamentally, methods of aggregation: they combine and merge data from several different sources into a single value, the trustworthiness of a target. As argued in the introduction, the evaluation of a target is dependent on the *beliefs* the evaluator has about the world, as well as the *goal* it is trying to achieve. Luckily most computational trust models come equipped with a way of implementing this dependency: they have parameters that can be used to adjust the behaviour of the trust model. The aim of AdapTrust is not to present another trust model, but to incorporate existing trust models into an intelligent agent. This can be used to deal with the multifaceted aspects of trust or, as we show in this article, adapt the trust model to improve communication about trust.

In any computational trust model, there are parameters that represent criteria for evaluating trustworthiness. For instance, many trust models use a parameter to give less importance to old information than new. This is useful if old information can become outdated and thus new information is more accurate than old. However, in a largely static environment this is not the case. The value of this parameter should be adjusted to the dynamicity of the environment. In general, the parameters of the trust model should be influenced by an agent's changing criteria for evaluating trustworthiness in a changing environment.

3.1 Priority System

The parameters of a trust model describe the importance of the different criteria for evaluating trustworthiness. However, it is more useful to consider this the other way round: the relative importance between the different criteria define a set of parameters for the trust model. These criteria are directly under an intelligent agent's control, and thus an agent is able to adapt its trust model. AdapTrust describes the specific techniques necessary to do this. The first of these is $\mathcal{L}_{PL}$, a language to describe the relative importance of any two criteria that influence a parameter of the trust model. We chose a subset of first-order logic with a family of predicates to define this importance relation, also called a priority ordering. For each parameter p of the trust model, the binary predicates $\succ_p$ and $=_p$ are defined with the expected properties of strict ordering and equality, respectively. The terms of the language are a set of elements representing the criteria that influence how the trust model should work. A Priority System is defined as a satisfiable theory in this language. For instance, consider an eCommerce environment. If an agent uses a weight w to calculate its evaluation of a sale and it finds the price of an item to be more important than its delivery time, it can have the priority $price \succ_w delivery\_time$ in its Priority System.

3.2 Priority Rules

The second technique of AdapTrust is to create the link between, on the one hand, an agent's beliefs and goals and, on the other hand, the priority between the different criteria for evaluating trust. This link makes explicit the adaptive process: a change in an agent's beliefs or goals effects a change in the priorities over the criteria, which changes the parameters of the trust model. The connection between the beliefs or goals and the priorities is made through what we call *priority rules.* The priority rules are specified using another first-order language, $\mathcal{L}_{Rules}$, with predicates $\rightsquigarrow_{Belief}$ and $\rightsquigarrow_{Goal}$ specifying how a set of beliefs, or a goal, respectively, leads to a specific priority relation between two criteria. By using these rules, we see that when the belief base changes the priorities can change. Additionally this is how the multifaceted aspect of trust is emphasized: the goal the agent is trying to achieve influences the priority system and thus the trust model. For instance, in the eCommerce example above, our agent might need to buy a bicycle urgently. It then has the goal $buy\_urgent(bicycle)$. For this goal, delivery time is more important than the price, so it has the priority rule $buy\_urgent(bicycle) \rightsquigarrow_{Goal} (delivery\_time \succ_w price)$ and therewith *adapts* its trust model to the requirements of the goal.

We do not go into detail on how these priority rules come to be. They can be programmed by a designer, or generated dynamically by a machine learning algorithm. However, what we are interested in here, is that they can also be incorporated through communication with another agent. We will return to this in Section 5, but first we describe the basic argumentation framework that we extend to allow for this communication. For a full description of the AdapTrust mechanism we refer an interested reader to [9].

4. PINYOL'S ARGUMENTATION METHOD

In the previous section we addressed two of the three requirements for agents to provide personal communications about trust. The last is that they are able to communicate about their criteria for evaluating trust. These criteria are given by an agent's beliefs and goals. What we need is thus a communication language that allows agents to talk about trust evaluations, the beliefs and goals these depend on, and the causal relationship between the two. We present this in Section 5, however Pinyol proposed a partial solution to this problem: an information-seeking dialogue for communication about trust [14]. The main aim of this framework is to allow the receiver of a communicated evaluation to decide whether or not to accept it. The framework creates an argument abstracting away from the computational process of the trust model, thereby allowing agents to discover what the original sources for evaluating a trust evaluation are. However, when asked *why* an aggregation of sources resulted in a specific evaluation, the model can only repeat itself as this is modeled as a ground element of the argumentation language. Our proposal extends the framework and allows agents to answer such questions, but first we summarize Pinyol's argumentation framework.

4.1 Trust as an inferential process

Pinyol starts by modeling the trust model as an inference relation between sentences in $\mathcal{L}_{Rep}$, a first-order language about trust and reputation [14]. This language is defined by a taxonomy of terms used for describing the process of computing trust. A trust model is considered as a computational process: given a finite set of inputs, such as beliefs about direct experiences or reputation, it calculates a trust evaluation for a target. The semantics of a computational process can be given by the application of a set of inference rules [7]. We define this as follows:

DEFINITION 1 (SEMANTICS OF A TRUST MODEL). *We say that a set of inference rules $\mathcal{I}$ is a specification of a trust model if, given input Δ and the resulting trust computation δ, we have that $\Delta \vdash \delta$, i.e., there exists a finite number of applications of inference rules $\iota \in \mathcal{I}$ by which we may infer δ from Δ.*

The inference rules themselves depend on the specifics of the computational process and thus the actual trust model being used, but for any computational trust model, such an inference relation exists. For instance, a trust model might have a rule:

$$\frac{img(T,X), rep(T,Y)}{trust(T, \frac{X+Y}{2})}$$

With img, rep and $trust$ predicate symbols in $\mathcal{L}_{Rep}$ and T, X and Y variables. For a specific target Jim, an agent knows $\{img(Jim,3),\ rep(Jim,5)\}$. It can thus infer $trust(Jim,4)$ using the rule above. For a full example of representing a trust model in inference rules, we refer to [12].

4.2 Arguing about trust

Arguments are sentences in the $\mathcal{L}_{Arg}$ language. This language is defined over another language $\mathcal{L}_{KR}$, that represents object-level knowledge . In Pinyol's framework $\mathcal{L}_{KR} = \mathcal{L}_{Rep}$, but in Section 5 we will supplement this language in order to extend the argumentation. A sentence in $\mathcal{L}_{Arg}$ is a formula $(\Phi : \alpha)$ with $\alpha \in \mathcal{L}_{KR}$ and $\Phi \subseteq \mathcal{L}_{KR}$. This definition is based on the framework for defeasible reasoning through argumentation, given by Chesñevar and Simari [4]. This framework of argumentation provides a clear manner for constructing arguments from an underlying language, rather than just providing a way for resolving what set of arguments fulfill certain criteria, which is the usual role of an argumentation framework [5, 2]. An alternative could be to model the trust model using a bipolar argumentation framework [1], however we choose to follow Pinyol's approach, which we explain here. Intuitively Φ is the defeasible knowledge required to deduce α. Defeasible knowledge is the knowledge that is rationally compelling, but not deductively valid. The meaning here, is that using the defeasible knowledge Φ and a number of deduction rules, we can deduce α. The defeasible knowledge is introduced in a set of elementary argumentative formulas. These are called *basic declarative units.*

DEFINITION 2 (BASIC DECLARATIVE UNITS). *A basic declarative unit (bdu) is a formula $(\{\alpha\} : \alpha) \in \mathcal{L}_{Arg}$. A finite set of bdus is an argumentative theory.*

Arguments are constructed using an argumentative theory Γ and the inference relation $\vdash_{Arg}$, characterized by the deduction rules *Intro-BDU, Intro-AND* and *Elim-IMP.*

DEFINITION 3 (DEDUCTION RULES OF $\mathcal{L}_{Arg}$).

$$\textit{Intro-BDU:}\frac{}{(\{\alpha\} : \alpha)}$$

$$\textit{Intro-AND:}\frac{(\Phi_1 : \alpha_1), \ldots, (\Phi_n : \alpha_n)}{(\bigcup_{i=1}^{n} \Phi_i : \alpha_1 \wedge \cdots \wedge \alpha_n)}$$

$$\textit{Elim-IMP:}\frac{(\Phi_1 : \alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta), (\Phi_2 : \alpha_1 \wedge \cdots \wedge \alpha_n)}{(\Phi_1 \cup \Phi_2 : \beta)}$$

An argument $(\Phi : \alpha)$ is valid on the basis of argumentative theory Γ iff $\Gamma \vdash_{Arg} (\Phi : \alpha)$. Because the deduction rules, and thus $\vdash_{Arg}$, are the same for all agents, they can all agree on the validity of such a deduction, however each agent builds its own argumentative theory, using its own trust model. Let $\mathcal{I}$ be the set of inference rules that specify an agent's trust model. Its bdus are generated from a set of $\mathcal{L}_{Rep}$ sentences Δ as follows:

- For any ground element α in Δ, there is a corresponding bdu $(\{\alpha\} : \alpha)$ in $\mathcal{L}_{Arg}$.
- For all $\alpha_1, \dots, \alpha_n$ such that $\Delta \vdash \alpha_k$ for all $k \in [1, n]$, if there exists an applicaton of an inference rule $\iota \in \mathcal{I}$, such that $\frac{\alpha_1,\dots,\alpha_n}{\beta}$, then there is a bdu $(\{\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta\} : \alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta)$, i.e., there is a bdu for every instantiated inference rule for the model specified by $\mathcal{I}$.

$\mathcal{L}_{Arg}$ is a non-monotonic logic and implication is defined in a similar manner to implication in logic programming. For details on the semantics, we refer to Chesñevar and Simari's work [4]. Continuing the example from above, our agent might have bdus:
$(\{img(Jim, 3)\} : \ img(Jim, 3))$,
$(\{rep(Jim, 5)\} : \ rep(Jim, 5))$ and
$(\{img(Jim, 3) \ \wedge \ rep(Jim, 5) \rightarrow trust(Jim, 4)\} :$
$\quad img(Jim, 3) \ \wedge \ rep(Jim, 5) \rightarrow trust(Jim, 4))$.
These bdus constitute an argumentative theory, from which $(\Phi : trust(Jim, 4))$ can be inferred, with Φ the union of the defeasible knowledge of the argumentative theory. Similarly, working backwards, an agent can build a valid argument supporting a trust evaluation it believes. Moreover, it can communicate this argument. This forms the first part of the information-seeking dialogue we need, in order to enable personalized trust communications. The problem, however, is that the trust model's functioning is introduced into the argumentation language in the form of bdus (see above). This means agents cannot explain why their trust model performs a specific calculation, because it is treated as defeasible knowledge. In the next section we present our extension to this framework, that allows agents to explain the reasons for their trust model's functioning.

5. PERSONALIZED TRUST

In this section we first present the extension of the argumentation framework, allowing agents to fully express the importance of criteria in their trust model. Subsequently we provide a dialogue protocol, allowing two agents to compare their trust evaluations, in order to discover where their trust models diverge. We provide a range of options to allow for the adaptation of their trust models, so that one agent can compute a trust evaluation tailored to the personal requirements of the other agent.

5.1 Extending the Argumentation Language

Pinyol's argumentation framework does not allow us to completely address the question of what criteria play a role in computing a trust evaluation, let alone connect these to underlying beliefs and goals. AdapTrust can answer this, but does not provide a language in which to do so. We propose to extend the argumentation framework presented in Section 4 with concepts from AdapTrust. In AdapTrust the *reason* an agent performs this computation and not some other one, is twofold: firstly the trust model follows an algorithmic method for aggregating the input. Secondly, the agent's beliefs and goals fix the parameters of this algorithm.

We do not propose to explain the algorithmic processes in the trust model, but the criteria, given by beliefs and goals, that define the trust model's parameters can be incorporated into the argumentative theory. For this, we need to represent the dependency of the trust model on the beliefs and goal of an agent in $\mathcal{L}_{Arg}$. In $\mathcal{L}_{Rep}$, the inference rules $\mathcal{I}$ specify a trust model algorithm. However, in AdapTrust this algorithm has parameters that depend on the agent's beliefs and goal. The inference rules should reflect this. Let $\Delta \subseteq \mathcal{L}_{Rep}$ and $\delta \in \mathcal{L}_{Rep}$, such that $\Delta \vdash \delta$. From Definition 1 we know there is a proof applying a finite number of inference rules $\iota \in \mathcal{I}$ for deducing δ from Δ. However, this deduction in AdapTrust depends on a set of the parameters, which we denote $Params$. Therefore, the inference rules must also depend on these parameters. For each $\iota \in \mathcal{I}$, we have $Params_\iota \subseteq Params$, the (possibly empty) subset of parameters corresponding to the inference rule and the set of parameters corresponding to a proof $\Delta \vdash \delta$ is simply the union of all parameters of the inference rules used in the deduction. Let the beliefs Ψ and goal γ determine the values for all these parameters. We denote this as $\Delta \vdash^{\Psi,\gamma} \delta$, which states that the trust model infers δ from Δ, given beliefs Ψ and goal γ. Similarly we have $\iota^{\Psi,\gamma} \in \mathcal{I}^{\Psi,\gamma}$ to denote a specific instantiation of the parameters $Params_\iota$ using beliefs Ψ and goal γ.

This allows us to redefine the set of bdus and thus the argumentative theory in such a way that the argumentation supporting a trust evaluation can be followed all the way down to the agent's beliefs and goal. $\mathcal{L}_{KR}$ must thus also be extended to encompass the various languages in AdapTrust, namely $\mathcal{L}_{KR} = \mathcal{L}_{Rep} \cup \mathcal{L}_{PL} \cup \mathcal{L}_{Rules} \cup \mathcal{L}_{Bel} \cup \mathcal{L}_{Goal}$, where $\mathcal{L}_{PL}$ is the language of priorities, $\mathcal{L}_{Rules}$ the language describing Priority Rules, $\mathcal{L}_{Bel}$ the language of the agent's beliefs and $\mathcal{L}_{Goal}$ that of the agent's goals. Using this $\mathcal{L}_{KR}$, the bdus for $\mathcal{L}_{Arg}$ are defined as follows:

DEFINITION 4 (BASIC DECLARATIVE UNITS FOR $\mathcal{L}_{Arg}$). *Let $\delta \in \mathcal{L}_{Rep}$ be an agent's trust evaluation based on inference rules $\mathcal{I}^{\Psi,\gamma}$, such that $\Delta \vdash^{\Psi,\gamma} \delta$ with $\Delta \subseteq \mathcal{L}_{Rep}$, $\Psi \subseteq \mathcal{L}_{Bel}$ and $\gamma \in \mathcal{L}_{Goal}$. For each $\iota \in \mathcal{I}^{\Psi,\gamma}$, let $Params_\iota$ be the corresponding sets of parameters. Let* labels *be a function that, given a set of parameters, returns a set of constants in $\mathcal{L}_{PL}$, the language of the priority system. Additionally let $\Xi \subseteq \mathcal{L}_{Rules}$ be the agent's set of trust priority rules and $\Pi \subseteq \mathcal{L}_{PL}$ be its priority system based on Ψ and γ, then:*

1. *For any sentence $\psi \in \Psi$, there is a corresponding bdu $(\{\psi\} : \psi)$ in $\mathcal{L}_{Arg}$.*
2. *The goal γ has a corresponding bdu $(\{\gamma\} : \gamma)$ in $\mathcal{L}_{Arg}$*
3. *For all priorities $\pi \in \Pi$ and all the rules $\xi \in \Xi$ the following bdus are generated:*
    - *if ξ has the form $\Phi \rightsquigarrow_{Belief} \pi$ and $\Phi \subseteq \Psi$ then $(\{(\bigwedge_{\varphi\in\Phi} \varphi) \rightarrow \pi\} : (\bigwedge_{\varphi\in\Phi} \varphi) \rightarrow \pi)$ is a bdu in $\mathcal{L}_{Arg}$*
    - *if ξ has the form $\gamma \rightsquigarrow_{Goal} \pi$ then $(\{\gamma \rightarrow \pi\} : \gamma \rightarrow \pi)$ is a bdu in $\mathcal{L}_{Arg}$*
4. *For all $\alpha_1, \dots, \alpha_n$ such that $\Delta \vdash^{\Phi,\gamma} \alpha_k$ for all $k \in [1,n]$, if there exists an application of an inference rule $\iota^{\Psi,\gamma} \in \mathcal{I}^{\Psi,\gamma}$, such that $\frac{\alpha_1,\dots,\alpha_n}{\beta}$ and* $\mathsf{labels}(Params_{\iota^{\Psi,\gamma}}) = L$ *then $(\{(\bigwedge_{\pi\in\Pi_L} \pi) \rightarrow (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta)\} : (\bigwedge_{\pi\in\Pi_L} \pi) \rightarrow (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta))$ is a bdu of $\mathcal{L}_{Arg}$. With $\Pi_L \subseteq \Pi$ the set of priorities corresponding to labels L.*

In items 1 and 2 the relevant elements of the agent's reasoning are added to the argumentation language. In items 3

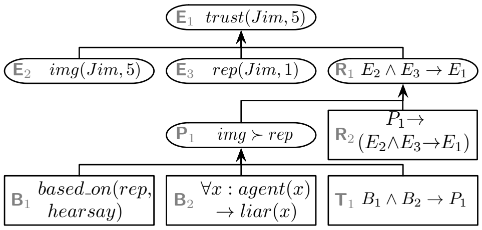

Figure 1: An example of an argument. The rectangular nodes are bdus.

and 4 the implements for reasoning about trust are added: in 3 the trust priority rules of AdapTrust, which link beliefs and goals to priorities, and in 4 the rules of the trust model. The bdus added in 4 contain a double implication: they state that if an agent has the priorities in Π_L then a *trust rule* (which was a bdu in Pinyol's argumentative theory) holds. In practice what this accomplishes, is to allow the argumentation to go a level deeper: agents can now argue about *why* a trust rule, representing an application of a deduction rule in the trust model, holds. An argument for a trust evaluation can be represented in a tree. We call this an argumentation tree and give an example of one in Figure 1. The argumentation tree can be followed by applying the deduction rules of $\mathcal{L}_{Arg}$ at each level. In order to be succinct we have omitted the defeasible knowledge part of the sentences in each node. Furthermore, we use shorthand in the tree by referring to nodes, rather than repeating the content of a node. For instance in node R_1 we can expand $E_2 \wedge E_3 \rightarrow E_1$ to its meaning: $img(Jim,5) \wedge rep(Jim,1) \rightarrow trust(Jim,5)$. An argumentation tree, such as this one, is used in a dialogue to communicate personalized trust evaluations.

5.2 Dialogue Protocol for Personalizing Trust

The argumentation in the previous section can be used by an individual agent to construe the reasons for having a trust evaluation in a language that the other agents understand. We now specify a protocol that allows agents to argue back and forth in such a way that an agent is assured that, if the dialogue completes successfully, it receives a personalized recommendation at the end. The protocol is summarized in the diagram of Figure 2.

The protocol defines a dialogue for two agents; a recommendation-seeker and a recommendation-supplier. If, at any point either of the agents does not want to continue conversing, it may end the dialogue immediately. If this happens, the seeker can use any information it has obtained during the dialogue, but there is no guarantee the trust evaluations communicated use the seeker's criteria for calculating trust. In the rest of this section, we describe the other options both participants have at each point in the dialogue.

The dialogue starts with the seeker contacting the supplier to request its recommendation of a partner to achieve the seeker's goal. The supplier provides a recommendation, at which point the dialogue begins in earnest. The guiding principle in the dialogue is that the seeker agent is trying to decide whether the recommendation is acceptable or what further information and adaptation is required for this. Thus, in the diagram of Figure 2 the first decision is whether or not to accept the argument. If the argument is not immediately accepted, or rejected, the next step is to decide which of the nodes of the argumentation tree is most likely to expedite this decision. This choice is made in the "Select node in argument" action of the diagram. In the description of the

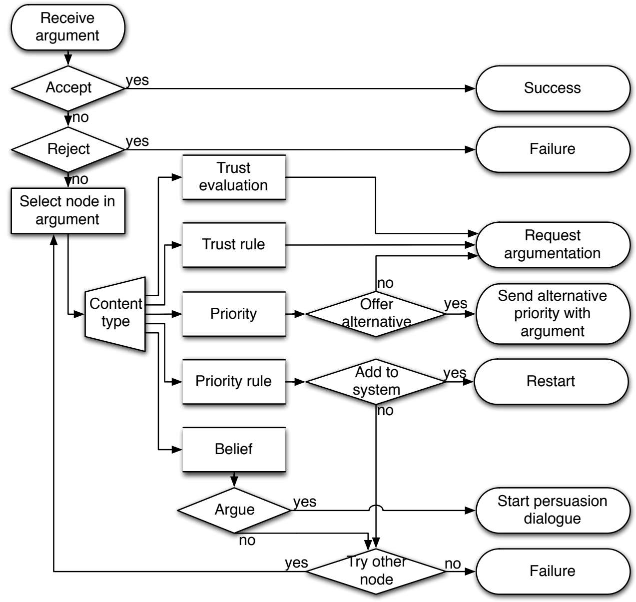

Figure 2: Diagram of the dialogue protocol

protocol below, we also describe this selection process. After selecting a node, the protocol determines what courses of action are available to an agent, based on the type of the node.

The example we use to describe the dialogue is the same as in the previous section, with the argumentation tree in Figure 1. The supplier does not reveal the entire argumentation tree at once. It only discloses information when the seeker asks for it. At the start of the dialogue, the supplier provides its trust evaluation E_1 and the direct reasons for having this evaluation (E_2, E_3 and R_1).

The seeker receives the initial argument. In order to decide whether it can accept the trust evaluation, it must decide whether it can accept the premises of the argument. This decision depends on the type of premises.

In $\mathcal{L}_{Arg}$, a trust evaluation is based on a trust rule and a number of inputs for the trust model. In the example these are trust rule R_1 and the trust evaluations E_2 and E_3. To decide whether or not to accept a trust rule, the seeker can compare it to the output of its own trust model, by using this with the inputs in the argument. In the example, the seeker can use trust evaluations E_2 and E_3 as inputs in its own trust model: if the output is equal to evaluation E_1 it accepts that the underlying criteria for the supplier's trust model are similar to the criteria in its own trust model. If not, it knows that the supplier's trust model is different from its own. The protocol gives a single course of action for trust rules: ask the supplier's reasons for it.

Another possibility is to ask about the trust evaluations used as input for the supplier's trust model. In the example these are trust evaluations E_2 and E_3. For trust evaluations, the protocol gives a single option: to ask the supplier for its reasons, which, if supplied, would expand those nodes in the argumentation tree. We omit these expansions, because the resulting subtrees are similar to the argumentation for the root, E_1. Instead we focus on the expansion of rule R_1. The seeker asks for the argument explaining why the supplier has trust rule R_1. Upon receiving this argument the seeker starts the decision process in Figure 2 again.

The reasons for a trust rule are clearly defined in $\mathcal{L}_{Arg}$. They are priorities over the criteria and a bdu that represents the dependency of the trust rule on these priorities. In the example there is only one priority, P_1, that influences the calculation of a trust evaluation from reputation and image. The first step in the protocol is once again to decide whether to accept or reject the argumentation, this time supporting node R_1. The seeker's trust model provides a way of deciding to reject the argument: if instantiating its trust model with priority P_1 does not allow it to compute E_1 from E_2 and E_3, then the agents' underlying algorithmic methods are too dissimilar for the supplier to be able to provide a personalized recommendation. Despite both agents using the same priorities to instantiate the parameters of their trust models, they compute different evaluations from the same input. In this case the dialogue ends in failure: the seeker should reject recommendations from the supplier and try another agent. Just as in Pinyol's framework, this is still useful information: the agents know that they disagree and that, in this situation, agreement is impossible.

If, in contrast, the seeker is able to emulate the supplier's trust calculation by using its priorities, then the only possible reason to not accept the trust rule outright is because the seeker disagrees with at least one of the priorities in the argumentation. The seeker can select such a priority and choose what to do. The protocol offers two options. The first is to ask *why* the supplier believes a priority holds. The second is to propose using its own priority instead. Note that the protocol allows an agent to explore both possibilities: if at a later stage it reaches "Try other node" it can try the alternative approach. The example continues using the former option for the only priority available, P_1, but the latter approach is equally valid and is described in Section 5.2.2.

5.2.1 Reasoning about the supplier's priorities

If the seeker asks why the supplier believes a priority holds, the dialogue continues. In the example, the reasons for the supplier having priority P_1 are in the 4th level of the argumentation tree of Figure 1.

The reasons for prioritizing one criterion over another, are given by the priority rules of AdapTrust, which are adopted as bdus in $\mathcal{L}_{Arg}$. These priority rules are supported by either beliefs or a goal. If the priority is supported by beliefs, as in the example, the protocol defines four possibilities:

1. The seeker chooses not to add the priority rule to its system. In this case its trust model will continue to be based on different criteria from the supplier's. It can try to convince the supplier to use its own priorities instead.

2. The seeker agent tries to add the priority rule to its system. This rule does not conflict with the rules it already knows. In this case it can be seen as a gap in the agent's knowledge and it can choose to adopt this rule.

3. The seeker agent tries to add the priority rule to its system and this rule does conflict with the rules it holds. In this case the agents have found a context in which agreement is impossible: the cognitive underpinnings of their trust models are different in this situation. The seeker agent should reject recommendations from the supplier in this context.

4. The agents enter a separate persuasion dialogue in order to convince each other about the validity of their beliefs. This can be done using a state-of-the-art argumentation framework for persuasion, such as the one proposed in [15].

Priority rules can also have goals in the antecedents, which are treated similarly, although the option for a persuasion dialogue is then not present. Conflicts between priority rules are defined as follows:

Definition 5 (Conflict of Priority Rules). *Let $U \subseteq \mathcal{L}_{Rules}$ be a set of priority rules such that:*

1. *$\Pi = \{\pi' | (\Phi' \leadsto_{beliefs} \pi') \in U\}$ is satisfiable in $\mathcal{L}_{PL}$*
2. *the set Φ, defined as the union of all Φ', such that $(\Phi' \leadsto_{Belief} \pi') \in U$, is satisfiable in $\mathcal{L}_{Bel}$*

Then a priority rule $\Psi \leadsto_{Belief} \pi$ conflicts directly with U iff $\Pi \cup \{\pi\}$ is unsatisfiable and $\Phi \models \Psi$.

A set of priority rules $\Xi \subseteq \mathcal{L}_{Rules}$ conflicts with a rule ξ if there is a set $U \subseteq \Xi$ that conflicts directly with ξ.

Note that two rules do not conflict if their antecedents are merely consistent, but only if one follows directly from the other. This is because two consistent antecedents with different conclusions might be designed to trigger in different situations, which is after all dependent on the beliefs and goals an agent has. In the case of two rules with conflicting conclusions triggering, AdapTrust contains a mechanism for choosing a consistent set of priorities. Definition 5 only defines conflicts for priority rules over beliefs. For goals it is the same, but then it is simply that a single goal leads to a conflicting set of priorities.

5.2.2 Reasoning about the seeker's priorities

If, instead of continuing the argument about the supplier's priority, the seeker proposes an alternative priority, the roles in the dialogue are switched. Now the supplier needs to discover why it should accept the seeker's priority. The same decision tree, in Figure 2, is used, but now the supplier performs the choices. Note that there are always less options, because using our $\mathcal{L}_{Arg}$, the reason for having a priority cannot be a trust evaluation, or a trust rule. Note that the supplier has the possibility to accept a priority rule into its knowledge base, but, unlike the seeker, can do this only temporarily: it may do this with the sole purpose of calculating a personalized trust evaluation for the seeker and its goal.

If at any point in the dialogue, either agent has adapted its trust model, they should restart the dialogue in order to verify that they have reached agreement and the supplier is able to provide personalized recommendations.

6. EXPERIMENTS

In the previous section we described a new argumentation framework to be able to communicate personalized trust evaluations. We now compare this model of communication to Pinyol's argumentation framework [14]. We have implemented AdapTrust using Jason [3]. In order to make a fair comparison, we keep everything as similar as possible to the experimental evaluation in [14], so we use the trust model Repage [18] and run the experiment in a simulated eCommerce environment, in which we evaluate the accuracy of buyers' trust evaluations of the sellers by using three methods of communication: (1) accepting other agents' trust evaluations directly (no argumentation), (2) filtering out mismatched communication with argumentation (Pinyol's argumentation) and (3) our model for communicating personalized trust evaluations.

6.1 The Simulation Environment

The simulation environment initially runs 20 agents who need to buy an item from any one of the 40 sellers in the environment, as in Pinyol's simulation. The sellers in this environment offer products with a constant price, quality and delivery time. These aspects of the product are used to evaluate the trustworthiness of the seller. A buyer can be “frugal”, in which case it gives more importance to the quality of the product than to the price or delivery time. A buyer can also be “stingy”, in which case it evaluates price as being more important than delivery time or quality. Finally, a buyer can be “impatient”, in which case the delivery time is the most important. The buyer profiles are implemented using AdapTrust’s priority rules, based on the beliefs of the agent.

In addition to these basic profiles, the buyers can have different goals. We have implemented the goal to buy a bicycle, which is not associated with any priority rules, and the goal to buy milk, which must be delivered quickly and thus has an associated priority rule to prioritize delivery time over both quality and price.

These two types of priority rules and the different profiles and goals of the agents allow them to benefit from the full dialogue of Section 5.2. Agents can attempt to persuade each other to switch their basic profile. Because we rely on pre-existing persuasion dialogues for this, we have simply hard-coded the outcome. A frugal agent can persuade a stingy agent to change its profile (and thus become frugal as well): a good quality item allows one to save money in the longer term by not needing to replace it as soon. This serves both agents’ underlying motivation of saving money. Furthermore, the different goals, and associated priority rules allow recommendation-suppliers to personalize their recommendation to the seeker’s goal, as well as have the agents exchange priority rules for their goal.

The simulation environment runs for 40 rounds to initialize the environment. In each round the buyers buy an item from a random seller. To ensure that no single buyer can obtain full knowledge, by interacting with all the sellers, each buyer can only interact with a configurable percentage of the sellers. This percentage is thus a control on the amount of knowledge each individual buyer can obtain about the set of sellers. After buying, the buyers can communicate their trust evaluations to exactly one other buyer. Depending on the type of communication we wish to evaluate, they use no argumentation, Pinyol’s argumentation, or personalized trust evaluations to perform this communication.

After this initialization, we create a new agent, which is the one to be evaluated. This agent knows nothing about the environment. It is a frugal agent with a 50/50 chance to have either goal, to buy a bicycle or milk, the same as the other buyer agents in the system. However, this agent does not explore by interacting with random sellers, but rather needs to discover the sellers’ trustworthiness through communication with the established buyers. For this, it uses the configured communication model, no argumentation, Pinyol’s model, or ours.

6.2 Simulation Results

The results are plotted in Figure 3. The experiments were run with an equal distribution of sellers offering one of either good quality, price or delivery time. Similarly the buyers were equally distributed over frugal, stingy and impatient agents. The experiment agent was always frugal and had a 50/50 chance of having the goal to buy a bicycle or milk. On the x-axis is plotted the percentage of sellers each buyer can interact with directly during the initialization and is thus a measure of the knowledge each agent can obtain about the

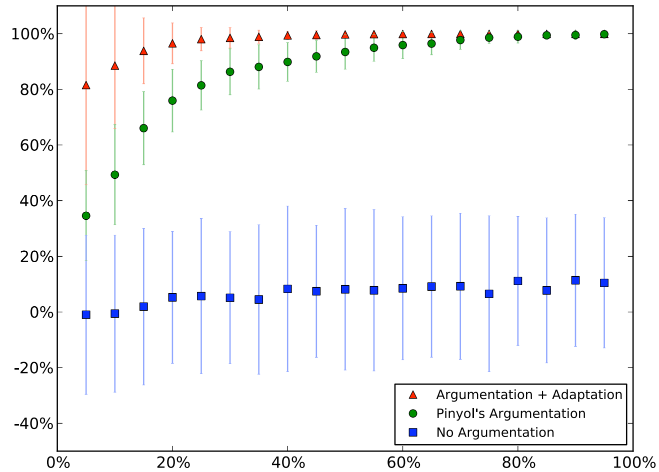

Figure 3: Experimental results. The x-axis represents the knowledge in the system and the y-axis the quality of the evaluation.

environment. With 20 buyers, 5% is the minimum to have all the sellers covered by at least one buyer. In this case, to obtain information about all sellers, information from all the buyers is needed. As the percentage of sellers each buyer can interact with increases, it becomes easier to obtain an accurate evaluation through communication, because the experiment agent needs to communicate with less of the established buyers to cover all the sellers. The y-axis plots the average accuracy of the evaluation of all the sellers in the system. After the experiment agent has finished its communication, it gives its evaluation of each of the seller agents. This “estimate” is compared to what its evaluation would be if it interacted with that seller. The difference of these two evaluations is the error of the experiment agent and we take the average of these errors as its score. To convert this to a percentage we compare this error to the expected error, if both the estimate and actual evaluations were chosen at random. This is equal to the expected difference between two standard uniform distributions, which is $\frac{1}{3}$. The y-axis thus plots the percentage increase in accuracy over this expected error of a random evaluation. Each point is the average of 100 experiments with the error bar being 1.96 standard deviations (representing an expected 95% of the population).

7. DISCUSSION

The experiment in the previous section is a proof-of-concept demonstration of the presented method of personalized trust communications. Despite being a prototypical implementation, the experiment already displays some interesting features of this method. Our method displays the greatest gains over Pinyol’s argumentation in scenarios where each individual agent has little information about the entire scenario. When the amount of information available to each buyer is high, both Pinyol’s and our own method can obtain near perfect information, because an agent can afford to discard information from a greater number of agents. However, when buyers do not have a lot of information, it is necessary to obtain information from a larger number of agents to accurately assess the trustworthiness of the sellers in the system. When buyers can interact with 20% of the sellers, our method is still slightly over 20% more accurate than Pinyol’s method in the experiment scenario. We feel this increase justifies the greater complexity and communication of our method. Note that agents having had direct interactions with 20% of the providers of a service is already on the high side for many eCommerce, P2P or grid computing scenarios. Despite this, we do not claim that the results from this experiment carry over to other scenarios. We run the experiment with a uniform distribution of both sellers' qualities and buyers' criteria in a simplified representation of an eCommerce environment. Even in this simple environment, if we change the parameters, we see different results. Specifically, the less likely it is that the experiment agent finds agents who are like-minded, the more important it becomes for it to obtain personalized trust recommendations from agents whose evaluation would otherwise need to be discarded. More experimentation is needed in more diverse scenarios to decide when personalized communication about trust offers useful benefits to the agents. This experiment's purpose is to demonstrate the method's functional viability and sketch the general domain in which we expect agents could use it.

7.1 Applications

Despite the experiment being based on a small and simulated scenario, it is able to show that even with just three parameters for the agents and two different goals, a filtering method will be left with too little information to work with, necessitating the use of a method such as the one proposed in this paper. This simple scenario serves as a proof of concept for its application in more realistic scenarios, such as the following:

P2P routing problems – one of the problems encountered in P2P networks is that of routing information. Deciding which peer can be trusted to transfer the required information does not have a trivial answer, especially if the network is used for diverse purposes, such as streaming different types of media, for which different agents have different requirements. Current trust and reputation models offer a possible solution [11], but only if they can get enough information. Because the environment is generally considered open and highly dynamic, exchanging information is a necessity for trust models to work, and our method provides this.

Automated eCommerce agents – the scenario we presented in the experimentation was a simplified eCommerce environment, but as the scenario is extended with more items and more properties of these items, the probability of coinciding with another agent decreases correspondingly. Therefore, despite there also being a far larger number of agents in the system, those with similar backgrounds to the own will still be sparse, necessitating a communication model such as the one we describe. If the community of sellers and buyers is relatively stable, then it might be possible to use a translation approach, as described in [8], but if this is not the case then our method provides a solution.

7.2 Future work

We recognize that the method we presented requires more extensive experimental evaluation. We regard such an evaluation as future work and intend to use personalized trust communication in different, realistic, scenarios and compare it to contemporary recommender systems or the use of reputation to give a more precise indication of what applications will truly benefit from this model. The permitted options in the dialogue can also be extended, for instance with persuasion about the correctness of priority rules. We intend to provide a formal dialogue protocol and include such extensions in the near future.

Acknowledgements

This work is supported by the Generalitat de Catalunya grant 2009-SGR-1434, the Agreement Technologies project (CONSOLIDER CSD2007-0022, INGENIO 2010) and the CBIT project (TIN2010-16306).

8. REFERENCES

[1] L. Amgoud and H. Prade. Using arguments for making and explaining decisions. *Artificial Intelligence*, 173(3–4):413–436, 2009.

[2] T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *JLC*, 13(3):429–448, 2003.

[3] R. Bordini, J. Hübner, and M. Wooldridge. *Programming MAS in AgentSpeak using Jason*. Wiley, 2007.

[4] C. Chesñevar and G. Simari. Modelling inference in argumentation through labelled deduction: Formalization and logical properties. *Logica Universalis*, 1(1):93–124, 2007.

[5] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 7(2):321–358, 1995.

[6] E. Erriquez, W. van der Hoek, and M. Wooldridge. An abstract framework for reasoning about trust (extended abstract). In *AAMAS'11*, pages 1085–1086, 2011.

[7] N. D. Jones. *Computability and Complexity: From a Programming Perspective*. MIT Press, 1997.

[8] A. Koster, J. Sabater-Mir, and M. Schorlemmer. Inductively generated trust alignments based on shared interactions (extended abstract). In *AAMAS'10*, pages 1571–1572, 2010.

[9] A. Koster, M. Schorlemmer, and J. Sabater-Mir. Opening the black box of trust: Reasoning about trust models in a BDI agent. *JLC*, In Press.

[10] S. Parsons, Y. Tang, E. Sklar, P. McBurney, and K. Cai. Argumentation-basded reasoning in agents with varying degrees of trust. In *AAMAS'11*, pages 879–886, 2011.

[11] A. Perreau de Pinninck, M. Schorlemmer, C. Sierra, and S. Cranefield. A social network defence against whitewashing. In *AAMAS'10*, pages 1563–1564, 2010.

[12] I. Pinyol and J. Sabater-Mir. Towards the definition of an argumentation framework using reputation information. In *Proc. of TRUST@AAMAS'09*, pages 92–103, 2009.

[13] I. Pinyol and J. Sabater-Mir. An argumentation-based protocol for social evaluations exchange. In *ECAI'10*, Lisbon, Portugal, 2010.

[14] I. Pinyol Catadau. *Milking the Reputation Cow: Argumentation, Reasoning and Cognitive Agents*, volume 44 of *Monografies de l'IIIA*. CSIC, 2011.

[15] H. Prakken. Models of persuasion dialogue. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, chapter 14, pages 281–300. Springer, 2009.

[16] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In *Proc. of KR'91*, pages 473–484. Morgan Kaufmann, 1991.

[17] F. Ricci, R. Lior, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer, 2010.

[18] J. Sabater-Mir, M. Paolucci, and R. Conte. Repage: REPutation and imAGE among limited autonomous partners. *JASSS - Journal of Artificial Societies and Social Simulation*, 9(2), 2006.

[19] W. Teacy, J. Patel, N. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *JAAMAS*, 12(2):183–198, 2006.