A Formal Argumentation Dialogue for Personalised Trust Communication

Andrew Koster[1,2], Jordi Sabater-Mir[1], and Marco Schorlemmer[1]

[1] IIIA-CSIC
Bellaterra, Spain
{andrew, jsabater, marco}@iiia.csic.es
[2] Universitat Autònoma de Barcelona
Bellaterra, Spain

Abstract. In order to allow agents in a multi-agent systems to effectively communicate about trust, we propose to personalise the communication about trust. In previous work we described AdapTrust and an argumentation framework; the former to adapt the trust model to the needs of the receiving agent and the latter to formalise a language in which to communicate. In this paper we propose a dialogue protocol for this communication, formalising it as an extension of Prakken's dialogue game for an information-seeking dialogue.

1 Introduction

Trust models for computational agents often center around the processing of evidence from direct interactions with a target. Nevertheless, in sufficiently large multi-agent systems it is unfeasible for agents to have had previous direct interactions with all targets they may need to interact with. It is thus essential that they can communicate about their trust evaluations, in order to decide whether or not a target is trustworthy. Trust, however, is a personal and subjective evaluation of a target for the fulfillment of a specific goal, and it is not straightforward to communicate this.

So far, communication methods for trust have relied mainly on machine-learning techniques to translate another agent's evaluation into the receiver's own frame of reference [1,5,12]. All of these methods learn a translation based on interactions that the recommendation-seeking agent shares with the recommendation-supplier, and for this to succeed the agents must share a large number of interactions. If this is not the case, the agents cannot learn an accurate translation and another method is required.

In previous work we proposed that the recommendation-supplier adapts, or *personalises*, its trust evaluation to the specific needs of the recommendation-seeker, rather than the seeker attempting to translate the unadapted recommendation from the supplier [4]. We presented an argumentation framework for communicating the beliefs that influence the agent's trust evaluation and the goal for which it evaluates the target. In this paper we propose a formal dialogue protocol for this argumentation. This dialogue protocol clearly defines the moves an agent is allowed to make when communicating in order to personalise communication about trust.

In Section 3 we present a dialogue protocol for the argumentation and personalisation of trust, but first we briefly summarise our argumentation framework and a prerequisite for it: a method for incorporating the trust model into an intelligent agent. AdapTrust is an extension to the BDI-agent model that allows an agent to adapt its trust model according to its beliefs and goals.

2 Preliminaries

Our method for enabling personalised communication about trust is based on three capabilities an agent must have. Firstly, an agent must be able to adapt its trust model in order to personalise its evaluations to the other agent's needs. Secondly, an agent must be capable of communicating its criteria for evaluating trust, as well as the underlying beliefs and goals leading to these criteria. Finally, 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. The ability to perform this adaptation requires that the trust model is incorporated into the cognitive reasoning process of the agent and we propose to use AdapTrust for this [6]. To communicate about how the trust model needs adapting in order to personalise a trust recommendation we use an argumentation language [4] and the dialogue protocol that we present in this paper. We start with a brief explanation of AdapTrust and the argumentation language.

2.1 AdapTrust

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. Moreover, this computation depends 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 [6]. AdapTrust works by changing the parameters of the trust model in accordance with the beliefs an agent has about the environment and the goal the agent wants to achieve, and for which it needs the trust evaluation.

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 use a subset of first-order logic, with a family of special 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 language uses a set of constants to represent 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.

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 in turn 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 emphasised: the goal the agent is trying to achieve influences the priority system and thus the trust model. For instance, in an eCommerce example, an agent might need to buy an item urgently. It then has the goal $buy\_urgent(item)$. For this goal, delivery time is more important than the price, so it has the priority rule $buy\_urgent(item) \rightsquigarrow_{Goal} (delivery\_time \succ_w price)$. This *adapts* its trust model to the requirements of the goal. The priority rules are atomic predicates, rather than implications in $\mathcal{L}_{Rules}$, because standard first-order semantics for material implication should not hold. For instance, if the agent does not have $delivery\_time \succ_w price$ in its priority system, this does not mean that the agent does not have the goal $buy\_urgent(item)$. There may just be other, conflicting priority rules that have precedence over this rule.

2.2 Argumentation Framework

Our argumentation framework extends Pinyol's framework for arguing about trust [8], and is explained in more detail in [4]. In this section we summarise the argumentation framework and language.

The first requirement for arguing about trust is that the agents have a common language in which to describe their trust evaluations. We use the $\mathcal{L}_{Rep}$ language, described by Pinyol et al. [9]. This is a first-order language about trust and reputation, defined by a taxonomy of predicates used for describing the process of computing trust. Some of the predicates describe what they refer to as *ground elements*, such as direct experiences and communications. We represent the ground elements as the set $ground(\mathcal{L}_{Rep})$. Other predicates describe "higher" concepts, such as the outcome of a direct experience or the reputation of a target. In Pinyol's argumentation framework, $\mathcal{L}_{Rep}$ is sufficiently expressive for all the communication, but we need to extend the language we use. First, however, we describe one thing our frameworks have in common: the interpretation of a trust model as an inference relation.

Trust Models and Inference A key point of both Pinyol's argumentation framework and our own is the focus on how to generate arguments. For this we build on the representation of any computational process as the application of a finite set of inference rules [3]. A trust model is a computational process and can thus be represented by a set of inference rules. The process of calculating a trust evaluation can be seen as the finite application of a number of inference rules $\mathcal{I}$ on a set of inputs $\Delta \subseteq ground(\mathcal{L}_{Rep})$ to obtain the output $\delta \in \mathcal{L}_{Rep}$. We write $\Delta \vdash_{\mathcal{I}} \delta$. 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. An example could be to infer the trust evaluation from reputation as follows:

$$\frac{rep(T,X)}{trust(T,X)}$$

The main difference between Pinyol's framework and our own, is that we assume the trust model is integrated into the agent's cognitive process by using AdapTrust, and it is therefore dependent on the agent's beliefs and goals: trust is an evaluation of a target for a specific goal, given the evaluator's beliefs about the environment. These beliefs and this goal influence how the trust model computes an evaluation and this must be represented as well in the inference rules. We assume the agent's beliefs and goals are represented in logical languages $\mathcal{L}_{Bel}$ and $\mathcal{L}_{Goal}$, as is the case in AdapTrust. For a set of beliefs $\Psi \subseteq \mathcal{L}_{Bel}$ and a goal $\gamma \in \mathcal{L}_{Goal}$ we have a set of inference rules $\mathcal{I}^{\Psi,\gamma}$, and we write $\Delta \vdash_{\Psi,\gamma} \delta$ to represent that input $\Delta \subseteq ground(\mathcal{L}_{Rep})$ results in trust evaluation $\delta \in \mathcal{L}_{Rep}$ for goal γ, given beliefs Ψ.

The way these inference rules are affected by the beliefs and goal is defined in AdapTrust: a set of beliefs and a goal cause certain priority rules to trigger, which leads to a set of priorities. A set of priorities describes a legal set of values for the parameters and in this way the trust model is adapted to the beliefs and goals. Not all inference rules are affected by the same priorities, because not all inference rules use the same parameters. We thus see that for a set of beliefs Ψ and a goal γ, we have that for any $\iota \in \mathcal{I}^{\Psi,\gamma}$ there is a (possibly empty) set of parameters $params(\iota)$. The values for the parameters, in turn, are prescribed by a set of priorities $\Pi_{\Psi,\gamma}$.

Arguing about Trust To be able to communicate about the trust process we must describe a formal language. We use the argumentation framework presented by Chesñevar and Simari, LDS_{ar} [2], which provides an intuitive way for representing the inference rules $\mathcal{I}$ in a communication language $\mathcal{L}_{Arg}$. $\mathcal{L}_{Arg}$ is a labelled language for defeasible reasoning, but for simplicity we omit the labels (for the full formalisation see [4]). We interpret it as a non-monotonic propositional language, in which we allow the connectives $\wedge$ as conjunction, and $\rightarrow$ as non-monotonic implication with semantics as in logic programming (for the formal semantics, see [2]). The language has three deduction rules, which are:

$$\text{Intro-BDU: } \frac{}{\alpha} \quad \text{for any } \alpha \in \mathcal{L}_{KR}$$

$$\text{Intro-AND: } \frac{\alpha_1, \ldots, \alpha_n}{\alpha_1 \wedge \cdots \wedge \alpha_n}$$

$$\text{Elim-IMP: } \frac{\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta, \alpha_1 \wedge \cdots \wedge \alpha_n}{\beta}$$

These deduction rules are used to deduce the conclusion of an argument from the *argumentative theory*, which is a set of basic sentences in $\mathcal{L}_{Arg}$ that are called *basic declarative units* (bdus). These bdus are ground sentences in an underlying language for knowledge representation $\mathcal{L}_{KR}$. In Pinyol's framework that was $\mathcal{L}_{Rep}$, but we will extend this.

Let the agent have beliefs Ψ and goal γ for which a trust model infers trust evaluation δ from input Δ. We write $\Delta \vdash_{\Psi,\gamma} \delta$ using inference rules $\mathcal{I}^{\Psi,\gamma}$. Let $\iota \in \mathcal{I}^{\Psi,\gamma}$ be an inference rule such that $\alpha_1, \ldots, \alpha_n \vdash_\iota \beta$, with $\alpha_1, \ldots, \alpha_n, \beta \in \mathcal{L}_{Rep}$, and the values of the parameters $params(\iota)$ are prescribed by the priorities $\Pi_{\Psi,\gamma}$; then we add a bdu $(\bigwedge_{\pi \in \Pi_{\Psi,\gamma}} \pi) \rightarrow (\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow \beta)$ in $\mathcal{L}_{Arg}$. We do this for all $\iota \in \mathcal{I}^{\Psi,\gamma}$.

Furthermore we add a bdu for each priority rule: if $\Phi \rightsquigarrow_{Belief} \pi$ is a priority rule, then $\Phi \rightarrow \pi$ is a bdu. The same for any goal $\gamma' \rightsquigarrow_{Goal} \pi$, we have $\gamma' \rightarrow \pi \in \mathcal{L}_{Arg}$. We also add all $\delta' \in \Delta$, all the agent's beliefs $\psi \in \Psi$ and the agent's goal γ as bdus. This means that the knowledge representation language $\mathcal{L}_{KR}$ that underlies $\mathcal{L}_{Arg}$ must be extended too. We have $\mathcal{L}_{KR} = \mathcal{L}_{Rep} \cup \mathcal{L}_{PL} \cup \mathcal{L}_{Rules} \cup \mathcal{L}_{Bel} \cup \mathcal{L}_{Goal}$.

The set of bdus generated in this manner gives a way for an agent to justify its trust evaluation in $\mathcal{L}_{Arg}$. While the above description seems to imply that an agent starts from the ground elements in $\mathcal{L}_{Rep}$, its beliefs and its goals, to generate bdus in $\mathcal{L}_{Arg}$, in actual fact the reverse is true. An agent uses its trust model, influenced by its beliefs and goal, to calculate a trust evaluation. It then traces this process back and encodes its calculation process, and the inputs it used, as bdus in $\mathcal{L}_{Arg}$. Because sentences in $\mathcal{L}_{Arg}$ are communicable between agents, any agent can follow another's reasoning and deduce the trust evaluation from the inputs without knowing any of the details of the other agent's trust model. By following this deduction process an agent can also reason about whether it agrees, or disagrees, with the other agent and why. We note the main reasons agents may disagree about the trust model:

- The agents disagree about (some of) the ground elements $\Delta \subseteq \mathcal{L}_{Rep}$ that are introduced into $\mathcal{L}_{Arg}$ as bdus. This was dealt with in Pinyol's argumentation

framework and we do not go into detail about that [8]. In general we will assume that if agents disagree about the ground elements, then communication fails, but it should not happen often. In general the recommendation-seeking agent is asking for advice about a target that it has no, or little, knowledge of, and it can accept that the recommendation-supplier has had a number of direct experiences with the target.
- The agents disagree about (some of) the beliefs $\Psi \subseteq \mathcal{L}_{Bel}$ that are introduced as bdus. In this case the agents can enter a persuasion dialogue to try to reach an agreement about beliefs.
- The agents have different goals. The recommendation-seeking agent should make it clear from the start that the recommendation is needed for a specific goal, and the recommendation-supplier should use this goal in its trust computation. If this does not happen, the recommendation-seeker should reject the recommendation.
- The agents disagree about a set of priorities that beliefs Ψ and goal γ lead to. In this case, the agents have different priority rules in AdapTrust. They can communicate these priority rules between each other and adapt their trust model.
- Despite having the same priorities, the agents disagree on the evaluation that can be inferred from a set sentences $\alpha_1, \ldots, \alpha_n \in \mathcal{L}_{Rep}$. In this case the agents' computational process is too different to be able to adapt: they agree on all the premises, but not on the conclusion. The recommendation-seeker should reject the recommendation from that supplier and try to communicate with a different agent.

In the next section we present a formal dialogue protocol in which two agents can argue about a trust evaluation and find places that they disagree. It then allows them to deal with this disagreement in the way we described above.

3 Dialogue Protocol for Personalising Trust

The argumentation in the previous section can be used by an individual agent to justify its trust evaluation in a language that the other agents understand. We now specify a protocol that allows agents to argue back and forth in order for the requesting agent to receive a personalised trust recommendation from the witness. We illustrate this protocol with an example, and start with explaining this example.

3.1 An example of argumentation

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 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: $img(Jim, 5) \ \wedge \ rep(Jim, 1) \rightarrow trust(Jim, 5)$, where img and rep are predicate symbols in $\mathcal{L}_{Rep}$, and are short for the agent's

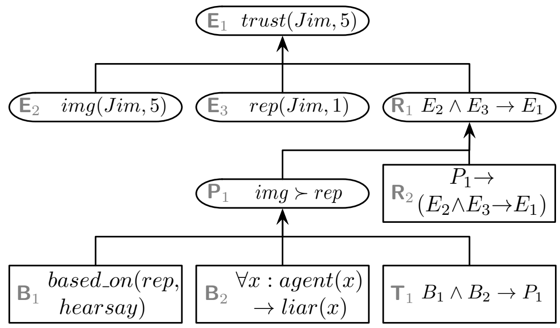

Fig. 1. An example of an argument. The rectangular nodes are bdus.

image, and reputation of the target. An argumentation tree, such as this one, is used in a dialogue to communicate personalised trust evaluations.

We do not explore all the paths in the tree and leave the nodes E_2 and E_3 unexplored, because their unfolding results in a similar structure to the unfolding of the root (E_1).

3.2 A formal dialogue protocol

We can now define a formal dialogue system for communication about personalised trust recommendations in which the argumentation can be communicated. The system we need is, for a large part, an information-seeking dialogue system, according to the classification by Walton and Krabbe [13]. It thus stands to reason that we use a protocol similar to the one presented by Parsons et al. [7]. However, while our dialogue is for a large part information-seeking, it also incorporates some aspects of persuasion dialogues. We thus present the formal system in a similar structure to the dialogue system presented by Prakken [10] for persuasion dialogues, in order to allow for some locutions in addition to the "question", "assert" and "challenge" locutions proposed by Parsons et al.

Definition 1 (Dialogue System for Personalised Trust (adapted from Prakken's Definition 3 [10])). *A* dialogue system *for personalised trust is a tuple* $\mathcal{D} = \langle \mathcal{L}_C, P, CR \rangle$ *where* $\mathcal{L}_C$ *(the communication language) is a set of* locutions*,* P *is a* protocol *for* $\mathcal{L}_C$*, and* CR *is a set of* effect rules *of locutions in* $\mathcal{L}_C$*, specifying the effects of the locutions on the participants' commitments.*

The three parts are described below, but first we must define some of the basic elements of a dialogue. The first of these is the set of participants themselves. These participants of the dialogue are the recommendation-seeker and recommendation-supplier, and we denote them with Q and R, respectively. Both of these agents have a *commitment store*, a set of sentences in $\mathcal{L}_{Arg}$ that they have committed themselves to [13]. Commitment is a complicated concept, but we use it in a very specific way: an agent's commitment store contains beliefs it has voiced during the dialogue and is *committed* to justify and defend. Because the dialogue is essentially an information-seeking dialogue, the recommendation-supplying agent R will mainly be the one committing itself to sentences in the dialogue. As the dialogue progresses, the recommendation-supplier will justify, in increasing detail, why the initially communicated trust evaluation holds. Every justification of this kind adds to the recommendation-supplier's commitment store. The agents' commitment stores are denoted C_Q and C_R for agents Q and R, respectively. Initially both agents' commitment stores are empty.

With these concepts in place we can move on to the definition of the locutions and protocol of a dialogue system. We start with the locutions.

Definition 2 (Locutions for Personalised trust). *The* locutions *allowed in the dialogue for personalised trust are specified by $\mathcal{L}_C$ and include the basic locutions for information-seeking, specified by Parsons et al. [7]. The locutions are explained in Table 1.*

Some of the locutions have an effect on an agent's commitment store. We usually denote the agent (either Q or R) that is sending a message, also called making a move, with s and the other agent with $\overline{s}$. We take C'_s to be the new commitment store of agent s after sending the locution, and C_s is the old commitment store prior to sending. The way the commitment store is updated for each locution is detailed in Table 2, which thus defines the rules CR of the dialogue.

Locution	*Use*
request_recommendation(t, γ)	The initial request for a recommendation, with $t \in Agents$ and γ the goal that Q wants the recommendation for.
assert(p)	Assert that p is true, where $p \in \mathcal{L}_{Arg}$.
justify(p, S)	Assert that $S \subset \mathcal{L}_{Arg}$ is the (direct) support for p in $\mathcal{L}_{Arg}$.
challenge(p)	Challenge a sentence $p \in \mathcal{L}_{Arg}$ in the other agent's commitment store. An agent may challenge a sentence p if it wants the other agent to justify p.
counter(π_R, π_Q)	Propose an alternative priority π_Q to priority π_R with $\pi_Q, \pi_R \in \mathcal{L}_{PL}$. Note that this switches roles: **counter** is similar in use to **assert**, so the agent Q, that has thus far only been challenging assertions, now proposes its own priority, that R can now challenge.
argue(ψ)	Propose to enter into a separate persuasion dialogue about beliefs $\psi \subset \mathcal{L}_{Bel}$. The details of this dialogue are outside the scope of this paper, but we propose to use the dialogue system proposed by Prakken [11].
end	Indicate that the dialogue has concluded.

Table 1. Locutions in $\mathcal{L}_C$, the communication language for personalised trust recommendation dialogues

The locutions **request_recommendation**, **assert** and **challenge** correspond directly to "question", "assert" and "challenge" in Parsons et al.'s system.

Locution	*Effect on commitment store*
request_recommendation(t, γ)	$C_Q = \emptyset$, $C_R = \emptyset$
assert(p)	$C'_s = C_s \cup \{p\}$
justify$(p,\ S)$	$C'_s = C_s \cup S$
challenge(p)	$C'_s = C_s$
counter$(\pi_1,\ \pi_2)$	$C'_s = C_s \cup \{\pi_2\}$
argue(ψ)	$C'_s = C_s$
end	$C'_s = C_s$

Table 2. The effect that the various locutions in $\mathcal{L}_C$ have on the sender's commitment store

Moreover, **justify** also corresponds to "assert" in Parsons et al.'s framework, but because they do not allow agents to backtrack, the sentence being justified is always immediately clear from the previous dialogue steps. The locutions **counter** and **argue** are not present in regular information-seeking dialogues. We add these so that agents can propose alternative priority systems for AdapTrust or attempt to persuade each other about their beliefs — thereby facilitating the adaptation of the agents' trust models.

If we look at the argumentation tree of Figure 1, then we see that, first of all, the recommendation-seeking agent must communicate its request, using **request_recommendation**(Jim, γ) with γ the goal for which it wants to know whether Jim is trustworthy or not.

The recommendation-supplier answers with **assert**$(trust(Jim, 5))$, which adds the atomic sentence $trust(Jim, 5)$ to its commitment store: $C_R = \{trust(Jim, 5)\}$ and $C_Q = \emptyset$. The seeker can now choose a next action, but there is only one action that makes sense: **challenge**$(trust(Jim, 5))$, resulting in the recommendation-supplier answering **justify**$(trust(Jim, 5), \{E_2, E_3, R_1\})$. The sentences it uses as justification are also added to its commitment store, so we have: $C_R = \{E_1, E_2, E_3, R_1\}$ Now the recommendation-seeker really does have a choice in its next move in the dialogue. We define *moves* and *legal moves* in the dialogue next.

Not all locutions can be uttered at any moment, there are rules to the dialogue. These are defined by the protocol P in terms of the moves allowed.

Definition 3 (Moves and dialogues (adapted from Prakken's Definition 5 [10])). *The set M of* moves *in a dialogue is defined as $\mathbb{N} \times \{R, Q\} \times \mathcal{L}_C$, where the three elements of a move m are denoted by, respectively:*

- *$id(m)$, the numerical identifier of the move*
- *$player(m)$, the agent performing in the move*
- *$speech(m)$, the speech act performed in the move*

The set of dialogues*, denoted by $M^{\leq\infty}$, is the set of all sequences $m_1, \ldots$ from M, such that each i^{th} element in the sequence has identifier i and for any $i > 1$, $player(m_i) \neq player(m_{i-1})$*[3]*. The set of* finite *dialogues is denoted by $M^{<\infty}$. For any dialogue $d = m_1, \ldots, m_i, \ldots$, the sequence $m_1, \ldots, m_i$ is denoted by d_i,*

[3] Note that this is a specific implementation of the turn-taking function in Prakken's dialogue system [10].

Locution	*Precondition. d is the dialogue so far and s the player*
request_recommendation(t, γ)	A recommendation-seeker may only request a recommendation in the first move, t must be a target and γ a goal. Formally: $d = d_0$, $t \in \mathit{Agents}$ and $\gamma \in \mathcal{L}_{Goal}$
assert(p)	A recommendation-supplier may only assert a trust evaluation in the second move, and the goal and target of the recommended trust evaluation must be equal to the goal and target for which it was requested. Formally: $d = m_1$, $\mathit{player}(m_1) = \overline{s}$, $p \in \mathcal{L}_{Rep}$, $\mathit{target}(p) = t$ and $\mathit{goal}(p) = \gamma$, with γ and t the goal and target in $\mathit{speech}(m_1)$, and functions *goal* and *target* return the goal and target for which an evaluation is made.
justify(p, S)	A sentence p can be justified, if it is in the current player's commitment store and the other player challenged it in a previous move. Formally: let $d = d_{i-1}; m_i$ and $\overline{s} = \mathit{player}(m_i)$, then there is a move m in d, such that $\mathit{player}(m) = \overline{s}$ and $\mathit{speech}(m) = \textbf{challenge}(p)$. Furthermore $p \in C_s$, $S \vdash_{Arg} p$ and $S \nsubseteq C_s$.
challenge(p)	A sentence p can be challenged, if it is in the other player's commitment store and the current player has not previously challenged it. Formally: let $d = d_{i-1}; m_i$ and $\overline{s} = \mathit{player}(m_i)$, then there is no move m in d such that $\mathit{player}(m) = s$ and $\mathit{speech}(m) = \textbf{challenge}(p)$. Furthermore $p \in C_{\overline{s}}$.
counter(π_1, π_2)	A priority π_1 can be countered by priority π_2, if it is in the other player's commitment store and π_2 is not yet in the current player's commitment store. Formally: let $d = d_{i-1}; m_i$ and $\overline{s} = \mathit{player}(m_i)$, then $\pi_1 \in C_{\overline{s}}$, $\pi_2 \notin C_s$ and $\pi_1, \pi_2 \in \mathcal{L}_{PL}$.
argue(ψ)	The current player may propose to argue about belief ψ if ψ is in the other player's commitment store and the player has not previously proposed to argue about ψ. Formally: let $d = d_{i-1}; m_i$ and $\overline{s} = \mathit{player}(m_i)$, then there is no move m in d such that $\mathit{speech}(m) = \textbf{argue}(\psi)$. Furthermore $\psi \in C_{\overline{s}}$ and $\psi \in \mathcal{L}_{Bel}$.
end	A player may always choose to end the dialogue after the first move. Formally: $d \neq d_0$

Table 3. The preconditions, in terms of the dialogue, for the various locutions in $\mathcal{L}_C$

where d_0 denotes the empty dialogue. When d is a dialogue and m a move, then $d; m$ denotes the continuation of d with m.

A protocol P on a set of moves M is a set $P \subseteq M^{<\infty}$ satisfying the condition that whenever $d \in P$, so are all initial sequences of d. We define a partial function $Pr : M^{<\infty} \rightarrow 2^M$ for personalised trust dialogues, that allows us to derive the protocol P. Prakken defines this in the opposite manner: with the protocol defining the function [10]. In practice, however, it is easier to define the function than all possible sequences of legal moves.

Definition 4 (Protocol function for Dialogues for Recommending Trust). $Pr : M^{<\infty} \rightarrow 2^M$ *defines the set of legal moves in a dialogue, and thus by induction defines the* protocol P *of a dialogue. We do this, by first defining the* preconditions *for each of the possible speech acts. These are listed in Table 3. We define the function pre that, given a speech act, a player and a dialogue, returns whether the preconditions are true or false. This allows us to define a function that returns all legal moves, given the dialogue so far:*

– $Pr(d_0) = \{(1, Q, \textit{\textbf{request\_recommendation}}(t, \gamma, r))\}$
– $Pr(d; m_i) = \{(i+1, s, lm) |\ s = \overline{player(m_i)}\ \wedge\ lm \in \mathcal{L}_C\ \wedge\ pre(lm, s, d; m_i)\}$

If the persuasion dialogue about argumentation is guaranteed to terminate, then the dialogue for recommending trust is guaranteed to terminate. The proof of this is trivial, given that $\mathcal{L}_{Arg}$ contains a finite number of elements and the protocol guarantees no steps are repeated. It depends, however, on the agents' choices of the legal moves how fast it reaches a desirable outcome. Such a desirable outcome is furthermore dependent on the agents actually adapting their trust models when necessary. This is not treated in the actual dialogue: if either agent receives a trust priority rule as the justification for a priority, it may choose to *add* this to its own rule base. This is a choice made outside of the dialogue, and if this happens then the argumentative theories change. This means the logic for the current dialogue no longer represents the agents' stances, and therefore the agent should choose to end the current dialogue. The seeker should restart with a new request for recommendations. We describe the choices an agent can make in more detail in [4].

4 Discussion and Conclusions

The protocol we propose is specially designed for a dialogue with the aim to personalise a trust recommendation. It therefore includes the locutions to counter a priority with another one, and propose to argue about beliefs that underly a trust evaluation. These two locutions are the main points of difference between the dialogue we propose and a standard information-seeking dialogue, such as presented by Prakken [10]. Moreover, we have taken special care to allow agents to choose what information they disclose.

First, at any point in the dialogue, the agent may choose to end the dialogue. Unfortunately, this also means that we can give no formal guarantees about whether a dialogue ends in success or not; an agent may always choose not to disclose an important piece of information for the adaptation process.

Secondly, the dialogue moves through an argument step-by-step. If a sentence in the agent's commitment store is challenged, the agent only justifies it with the direct reason for believing the sentence: only one node of the argument tree is unfolded at a time, and the agent may choose to end the dialogue, rather than disclose the information at any point. These three properties of our dialogue make it quite different from Pinyol's dialogue protocol [8], which is, insofar as we know, the only other dialogue protocol specifically designed for communicating about trust. Pinyol's dialogue protocol is more reminiscent of one for a persuasion dialogue, because each move communicates an entire argumentation tree and the only legal move is to counter some part of that tree. The dialogue only ends when an agent can no longer attack the other's arguments, at which point the recommendation-seeker decides whether or not to accept the trust recommendation or not.

The dialogue protocol presented in this paper thus improves on the state of the art in two ways: first, it extends existing dialogue protocols specifically to allow for the adaptation of trust models and the communication of personalised trust recommendations; and second, it allows agents a more fine-grained control over how much information they communicate than the only other argumentation protocol for trust that we know about.

References

1. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. Proceedings of the 33rd Hawaii International Conference on System Sciences 6, 4–7 (2000)
2. Chesñevar, C., Simari, G.: Modelling inference in argumentation through labelled deduction: Formalization and logical properties. Logica Universalis 1(1), 93–124 (2007)
3. Jones, N.D.: Computability and Complexity: From a Programming Perspective. Foundations of Computing, MIT Press (1997)
4. Koster, A., Sabater-Mir, J., Schorlemmer, M.: Personalizing communication about trust. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012). IFAAMAS (Forthcoming 2012)
5. Koster, A., Schorlemmer, M., Sabater-Mir, J.: Engineering trust alignment: Theory, method and experimentation. International Journal of Human-Computer Studies (Forthcoming 2012)
6. Koster, A., Schorlemmer, M., Sabater-Mir, J.: Opening the black box of trust: Reasoning about trust models in a bdi agent. Journal of Logic and Computation (Forthcoming 2012)
7. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. Journal of Logic and Computation 13(3), 347–376 (2003)
8. Pinyol, I.: Milking the Reputation Cow: Argumentation, Reasoning and Cognitive Agents, Monografies de l'Institut d'Investigació en Intel·ligencia Artificial, vol. 44. Consell Superior d'Investigacions Científiques (2011)
9. Pinyol, I., Sabater-Mir, J.: Arguing about reputation. The LRep language. In: Artikis, A., O'Hare, G., Stathis, K., Vouros, G. (eds.) Engineering Societies in the Agents World VIII: 8th International Workshop, ESAW 2007. LNAI, vol. 4995, pp. 284–299. Springer (2007)

10. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. Journal of Logic and Computation 15(6), 1009–1040 (2005)
11. Prakken, H.: Models of persuasion dialogue. In: Rahwan, I., Simari, G. (eds.) Argumentation in Artificial Intelligence, chap. 14, pp. 281–300. Springer (2009)
12. Regan, K., Poupart, P., Cohen, R.: Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI). pp. 1206–1212. AAAI Press, Boston, MA, USA (2006)
13. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State of University of New York Press, Albany, NY, USA (1995)