Trust as Dependence: A Logical Approach

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
singh@ncsu.edu

ABSTRACT

We propose that the trust an agent places in another agent declaratively captures an architectural *connector* between the two agents. We formulate trust as a *generic* modality expressing a relationship between a truster and a trustee. Specifically, trust here is *definitionally independent* of, albeit constrained by, other relevant modalities such as commitments and beliefs. Trust applies to a variety of attributes of the relationship between truster and trustee. For example, an agent may trust someone to possess an important capability, exercise good judgment, or to intend to help it. Although such varieties of trust are hugely different, they respect common logical patterns. We present a logic of trust that expresses such patterns as reasoning postulates concerning the static representation of trust, its dynamics, and its relationships with teamwork and other agent interactions. In this manner, the proposed logic illustrates the general properties of trust that reflect natural intuitions, and can facilitate the engineering of multiagent systems.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*; D.2.1 [**Software Engineering**]: Requirements Specifications—*Methodologies*

General Terms

Theory

Keywords

Trust, commitments, service-oriented computing

1. INTRODUCTION

We develop a novel approach to trust in multiagent systems that relates the intuition of trust as reliance with the notion of an architectural *connector* [17]. When the components of a software architecture are agents (understood as active, autonomous entities), each connector between any two agents is naturally understood in terms of the trust they place in each other. In this manner, we not only relate intuitions about two heretofore isolated subfields of multiagent systems (trust and agent-based software engineering), but also provide a new basis for formalizing those intuitions to use as a basis for improved engineering methodologies.

Classically, following Castelfranchi and Falcone [1], one may understand an agent (the *truster*) as trusting another (the *trustee*) when the truster puts its plans in the hands of the trusted agent. In general terms, the above is a valuable intuition that we seek to preserve. However, Castelfranchi and Falcone take a staunchly cognitive stance wherein a "plan" is reflected in the intentions and beliefs of the truster with respect to the trustee.

In contrast, we take the position that the notion of "plan" in general multiagent settings is often, though not always, far removed from the cognitive view. Referrals, which are crucial for inducing trust in social settings, often involve plans that might be quite tenuous. In other cases, one may spot a plan only based on strong assumptions about the truster and trustee, the tasks involved, and the context. Therefore, we advocate here an architectural intuition where the parties may not have strongly cognitive plans either.

Trust arises in many settings. For this reason, we develop a modular, "minimalist" formalization of trust, which captures the essential properties that any model of trust would follow. Our approach does not demand agreement on the additional aspects of trust—such as belief, intentions, plans, similarity, probability, utility—that specific models might incorporate and specific applications may demand. Thus our approach can provide a conceptual basis for organizing systems without having to delve into the details of trust.

We treat trust as a high-level architectural connector. A truster's trust in a trustee expresses the expectations the truster holds of the trustee. This interpretation of an architectural connector as the dependence of a truster on a trustee generalizes the classical software architecture [15] idea of one component's "assumptions" about another. Traditionally, such assumptions reduce to operational details of control and data flow, but in agent-oriented software engineering we ought to treat them as interagent dependencies.

Singh and Chopra [19] propose to use commitments as a basis for multiagent systems architecture. Commitments are appropriate bases for interaction where a protocol specifies the commitments involved. However, in flexible, emergent settings, such specifications might be incomplete or even nonexistent. That is, the agents should be prepared to interact with others even in the absence of commitments. In such cases, the basis for their interactions would be the trust that each agent places in the other. Even when a commitment exists, the creditor of the commitment would need to trust

Cite as: Trust as Dependence: A Logical Approach, Munindar P. Singh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 863-870.

the debtor in order to rationally act on the assumption that the debtor will discharge the commitment in consideration.

When we apply our trust-based approach on traditional software components, any modeling of trust would be implicit and hard-coded in the components—reflected only in the minds of the designers. When we apply our approach on sophisticated intelligent agents, the modeling of trust would be more explicit and subject to reasoning by the agents themselves.

Notice that sometimes architecture is conflated with notations for expressing it, especially established notations such as UML. Such notations have no abstractions geared toward trust and other high-level concepts, so we avoid them here.

Contributions. We present a formal semantics of trust, motivating several reasoning postulates for trust and relating those postulates to architectural connectors. Our contributions bear relevance also to the study of commitments, which we treat as correlates of trust. Further, the notion of architecture pursued here, although far removed from traditional software architecture, is inspired by taking a truly agent-oriented stance. Not only are agents a natural abstraction but also the trust between them is core to their interactions.

Organization. The rest of this paper is organized as follows. Section 2 discusses some intuitions about trust as it relates to architecture. Section 3 introduces our technical framework for trust. Section 4 presents a variety of postulates for trust describing potential properties of relevance to active trust, integrity, structure, meaning, teamwork, and dynamics. Section 5 presents a case study demonstrating our approach in relation to both traditional and more recent commitment-based approaches. Section 6 places our work in the broader setting of architecture and brings out some directions for future work.

2. INTUITIONS ABOUT TRUST

Trust is central to several disciplines. So it is not surprising that it has garnered a lot of research attention. Existing approaches differ a lot on the complexity of the conceptual model in which they consider trust. The following main lines of research reflect the intuition of dependence are relevant.

Subjective, which treat trust as a suitably structured set of beliefs and intentions [1]. Indeed, Demolombe [5] reduces trust to (graded) beliefs. Liau [12] and Dastani et al. [4] consider how a truster may absorb information from a trustee, e.g., by adopting a belief if a trusted sender says so.

Measured, which treat trust as a numeric weight based on heuristics [9], subjective probability [11, 23], a utility [3], or a grade [5]. These are subjective approaches albeit with representations geared toward numeric or ordinal values.

Social, which understand trust in terms of social relationships [20]. Falcone and Castelfranchi [7] distinguish objective and subjective dependence as well as unilateral, reciprocal, and mutual dependence. Our basic framework accounts for all of these, albeit with specific postulates describing different situations, e.g., teamwork. Johnson et al. [10] examine teamwork via social interdependence, which is crucial as a basis for trust.

A commonality of the existing approaches is that they conflate aspects of the *representation* of trust on the one hand with the complex of features that go into making a *judgment* of trustworthiness on the other. The latter involve reasoning techniques (often domain-specific and heuristic) for updating the extent of trust placed by a truster in a trustee. Indeed, there is a common confusion when talking about trust in that many researchers expect to see the above kinds of heuristics, and do not appreciate the value of a generic method, such as ours. As an analogy, one can think of rules of Bayesian inference or axioms of belief. Such rules and axioms do not in themselves produce an answer of what an agent should infer or believe, but constrain the probabilistic or binary truth values an agent may assign to various propositions. In the same way, our approach describes how an agent or a designer may reason soundly about trust.

We formalize a general-purpose semantically motivated representation of trust. Interestingly, this representation provides a basis for stating a variety of constraints on the modeling of trust with respect to the integrity and structure of architectural connectors, and of reasoning about trust. Although it is not focused on trust measures, it also provides a basis for such measures.

Conditionality of Trust. We posit that, in general, trust must be conditional. Each assignment of trust presupposes some preconditions (which we can capture as antecedents) and expectations (which we can capture as consequents). Blind trust is merely a boundary condition. This holds in normal usage: e.g., a customer may trust a merchant as follows "if I pay, (I trust) the merchant will deliver the goods," expressing the customer's expectation and presumably linking it to further plans of the customer.

Trust as Dependence, Architecturally. Let us consider an agent formulating and enacting a plan that relies upon the contributions of others—in essence, trusting the others to make their contributions to its plan. More generally, the interactions of an agent with other agents may be described at a high level in terms of the trust each of them places upon the others. We model further aspects of the interactions such as whether trusted agents are indeed trustworthy based on how the trust maps to relevant concepts.

Although the antecedent and consequent are generic, nominally, we associate them with the truster and trustee, respectively. When the antecedent becomes true, the connector *activates* and when the consequent becomes true, the connector *completes*. It is helpful to relate the antecedent and consequent of a trust expression to the structure of the connector it describes. Intuitively, a trust expression becomes stronger as its antecedent becomes weaker and its consequent becomes stronger. We can understand the antecedent becoming weaker with the connector becoming *broader* because it would activate more easily. Likewise, we can understand the consequent becoming stronger with the connector becoming *tighter* because it would complete with greater effort on part of the trustee, and thus sustain enhanced expectations on part of the truster.

This paper develops an organizational approach, especially from the standpoint of the connectors among autonomous agents understood conceptually. As explained above, in this view, a relationship from one agent to another can be understood as the trust the first agent places in the second agent.

An agent may implement such an interconnection based on concepts such as beliefs.

3. TECHNICAL FRAMEWORK

Our technical framework is based on modal logic with a possible worlds semantics. In addition to trust, we capture commitments as an abstraction because they help us state various important postulates reflecting dependence.

We include an explicit notion of *reality* in our model. That is, we identify a path (corresponding to a particular execution of the multiagent system) as being the real one. This is *not* to suggest that we have found a way to predict the future; rather, it is a way to accommodate nondeterminism by merely claiming (as appropriate) that whatever the real path might be, it satisfies some property, desirable or otherwise. For example, we might define trust as being well-placed if the proposition that is being trusted occurs on the real path. In this manner, incorporating reality explicitly enables us to state constraints that we cannot state otherwise.

3.1 Syntax and Formal Model

Putting together the intuitions about architectural connectors and the inherent conditionality of trust, we propose to formalize trust-as-dependence as a modal operator that takes two parties and two propositions, as in

$$\mathsf{T}_{\mathsf{truster,trustee}}(\mathsf{antecedent}, \mathsf{consequent})$$

The first two arguments describe the end points of the given connector, and the last two its logical structure. In logical terms, trust bears a syntactic similarity with commitments but the two are independent concepts. More generally, we can view trust and commitment as correlates of each other. Some of the postulates below relate trust and commitments.

$\mathcal{L}$, our formal language, takes a linear-time logic enhanced with a modality C for commitments [18] with a modality T for trust. Below, *Atom* is a set of atomic propositions and $\mathcal{X}$ is a set of agent names. We further define agents that are composed from other agents; in other words, an agent may be a simplistic multiagent system. L and X are nonterminals corresponding to $\mathcal{L}$ and $\mathcal{X}$, respectively.

L_1. $L \longrightarrow \mathit{Trust} \mid \mathit{Commit} \mid \mathit{Atom} \mid L \wedge L \mid \neg L \mid \mathsf{R}L \mid L \mathsf{U} L$

L_2. $\mathit{Trust} \longrightarrow \mathsf{T}_{\mathit{Agent},\mathit{Agent}}(L, L)$

L_3. $\mathit{Commit} \longrightarrow \mathsf{C}_{\mathit{Agent},\mathit{Agent}}(L, L)$

L_4. $\mathit{Agent} \longrightarrow X \mid \langle\{\mathit{Agent}\}\rangle$

We use the following conventions: x, etc. are agents, ψ, etc. are atomic propositions, p, q, r, etc. are formulae in $\mathcal{L}$, t, etc. are moments, and P, etc. are paths. We drop agent subscripts when they can be understood. A model for $\mathcal{L}$ is a tuple, $M = \langle \mathbb{S}, <, \mathbb{R}, \mathbb{I}, \mathbb{T}, \mathbb{C} \rangle$:

- $\mathbb{S}$ is a set of possible moments, each a possible snapshot (i.e., a state) of the world.
- $< \subseteq \mathbb{S} \times \mathbb{S}$ is a discrete linear order on $\mathbb{S}$, which induces *paths* at each moment. A path is a contiguous set of moments beginning at a moment. Two paths are either disjoint or one is a subset of the other. $[P; t, t']$ denotes a *period* on path P from t to t'. Formally, $[P; t, t']$ is the intersection of P with the set of moments between t and t', both inclusive. $\mathbb{P}$ is the set of all periods and $\mathbb{P}_t$ of periods that begin at t ($\mathbb{P}_t \neq \emptyset$).
- $\mathbb{R}$ identifies the *real path* that initiates from a moment. A real path must be self-consistent in that if a moment initiates a real path τ, every subsequent moment that occurs on path τ initiates a suffix of τ as its real path.
- The interpretation, $\mathbb{I}$, of an atomic proposition is the set of moments at which it is true. That is, $\mathbb{I} : \mathit{Atom} \mapsto \wp(\mathbb{S})$. We show below, through the definition of moment-intension (which lifts $\mathbb{I}$ to all propositions), that the denotations of all propositions are sets of moments.
- At each moment, $\mathbb{T} : \mathbb{S} \times \mathcal{X} \times \mathcal{X} \times \wp(\mathbb{S}) \mapsto \wp(\wp(\mathbb{P}))$ yields a set of periods for each moment and proposition for each truster-trustee (ordered) pair of agents.
- At each moment, $\mathbb{C} : \mathbb{S} \times \mathcal{X} \times \mathcal{X} \times \wp(\mathbb{S}) \mapsto \wp(\wp(\mathbb{P}))$ yields a set of periods for each moment and proposition for each debtor-creditor (ordered) pair of agents.

Models for modal logics are commonly based on Kripke structures, which define a set of possible worlds along with an accessibility relation that maps each world to a set of worlds. The semantics of a modal operator tests for *inclusion* in that set of worlds. The models proposed here are *not* Kripke structures and do not involve an accessibility relation. Instead they are based on the Montague (and Scott) approach [14] to define a "standard" of correctness by mapping each world to a set of sets of worlds. The semantics of a modal operator tests for *membership* in the set of sets of worlds. Montague's approach offers greater flexibility in allowing or denying some inferences that the Kripke approach requires. In many (though not all) cases, it is straightforward to map this semantics to a Kripke semantics but we find the proposed formulation more natural and modular.

$\mathbb{T}$ and $\mathbb{C}$ capture the standards for trust and commitments, respectively, for each moment and truster-trustee pair. Given an antecedent proposition, $\mathbb{T}$ yields a set, each of whose members is a set of periods. Each set of periods is the representation in the model of a consequent proposition, specifically, the proposition whose period-intension (defined below as the set of periods at whose culmination it holds) equals that set of periods. The truster trusts the trustee to bring about any such consequent if the antecedent holds. Likewise, $\mathbb{C}$ yields a set each of whose members is a set of periods, each culminating in the consequent proposition that the debtor commits to bringing about. As in many (arguably most) logics of intention and obligation, we do not model actions explicitly: $\mathbb{T}$ and $\mathbb{C}$ are simply understood as describing the conditions an agent would bring about.

3.2 Semantics

The semantics of $\mathcal{L}$ is given relative to a model, a path, and a moment on the path. $M \models_{P,t} p$ expresses "M satisfies p at t on path P." The truth of several constructs is independent of the path and depends only on the moment. An expression p is *satisfiable* (respectively, *valid*) iff for some (respectively, all) M, P, and $t \in P$, $M \models_{P,t} p$. Formally, we have:

M_1. $M \models_{P,t} \psi$ iff $t \in \mathbb{I}(\psi)$, where $\psi \in \mathit{Atom}$

M_2. $M \models_{P,t} p \wedge q$ iff $M \models_{P,t} p$ and $M \models_{P,t} q$

M_3. $M \models_{P,t} \neg p$ iff $M \not\models_{P,t} p$

M_4. $M \models_{P,t} \mathsf{R}p$ iff $M \models_{\mathbb{R}_t,t} p$

M_5. $M \models_{P,t} p\mathsf{U}q$ iff $(\exists t'' \in P : t \leq t''$ and $M \models_{P,t''} q$ and $(\forall t' : t \leq t' < t'' \Rightarrow M \models_{P,t'} p))$

Disjunction ($\vee$), implication ($\rightarrow$), equivalence ($\equiv$), false, and true are the usual abbreviations. $p\mathsf{U}q$ means "p holds until q": thus $\mathsf{true}\mathsf{U}q$ (abbreviated $\mathsf{F}q$) means "eventually q." And, $\mathsf{R}p$ means that p holds on the real path of the current moment.

We define the *moment-intension* of formula p as the set of moments where it is true: $[\![p]\!] = \{t | M \models_{P,t} p\}$. We define *period-intension* of formula p as the set of periods culminating in its becoming true: $\langle\!\langle p \rangle\!\rangle = \{[P; t, t'] | M \models_{P,t'} p\}$. In these periods, p occurs at the last moment but may possibly occur earlier as well. Thus these are all possible ways in which p may be brought about. Based on these, we can now specify the formal semantics of trust and commitments. As explained in connection with $\mathbb{T}$ above, $\mathsf{T}_{x,y}(r, u)$ holds precisely at points where the period-intension of u belongs to the standard for trust. (Likewise, for commitments).

M_6. $M \models_{P,t} \mathsf{T}_{x,y}(r, u)$ iff $\langle\!\langle u \rangle\!\rangle \in \mathbb{T}_{x,y}(t, [\![r]\!])$

M_7. $M \models_{P,t} \mathsf{C}_{x,y}(r, u)$ iff $\langle\!\langle u \rangle\!\rangle \in \mathbb{C}_{x,y}(t, [\![r]\!])$

4. REASONING POSTULATES

Let's now consider several postulates that reflect common reasoning patterns that apply uniformly to trust. It is worth emphasizing that we consider atomic propositions that are *stable*, meaning that they include any temporal requirements within them. Thus a proposition that is true is generally true forever. For example, let *pay* mean the agent pays by noon on May 1. If *pay* is true at one point on a run, it is true on all points on the run. Consequently, most of our postulates do not involve any temporal operators. Trust and commitments (which can become active and then inactive) are themselves not stable; thus some postulates that deal with them involve the until operator. We expand the notion of agents to treat simplified multiagent systems.

4.1 Postulates for Active Trust

We treat trust in the sense of a living, functioning architectural connector. That is, we consider the case of *active* trust. When a truster places trust in a trustee, the corresponding connector is activated. When the trustee has performed as expected, there is no more for the truster to expect of the trustee based solely upon the given connector. In such a case, the connector is no longer active.

Our approach helps distinguish between a connector that is inactive and one that which has been activated but not completed. The former is perfect; the latter is worrisome. As a result, often, we would formulate trust expressions as including the possibility of success. As a specific example, an agent x may deal with an agent y because it trusts y to deliver the goods if it pays. That is, we would have $\mathsf{T}_{x,y}(\mathit{pay}, \mathit{deliver})$. But to accommodate the unknown or early performance of *deliver*, we might instead formulate the trust expression as $\mathit{deliver} \vee \mathsf{T}_{x,y}(\mathit{pay}, \mathit{deliver})$

For each postulate below that uses truster x and trustee y, for brevity, we write $\mathsf{T}(r, u)$ instead of $\mathsf{T}_{x,y}(r, u)$.

T_1. COMPLETE A CONNECTOR. $u \rightarrow \neg\mathsf{T}(r, u)$

When u holds, the trust in u is completed and is, therefore, no longer *active* (this treatment is neutral as to whether u is the provision of information or the performance of a domain action). Notice that the above yields $\neg\mathsf{T}(r, \mathsf{true})$ for any r.

T_2. ACTIVATE A CONNECTOR. $\mathsf{T}(r \wedge s, u) \wedge r \rightarrow \mathsf{T}(s, u)$

A typical case is when a truster performs part or all of what it needs to do to activate a connector. For example, if you push money over a coffee counter you trust that the barista would push back a cup of coffee for you. If you trusted the barista to give you a cup of coffee upon your paying $1, upon handing over $1 you trust the barista to give you the cup of coffee without further ado.

More generally, a connector may be activated piecemeal. When "part of" the antecedent of a connector holds, the connector strengthens to one for the "remainder" of the antecedent and with the original consequent comes into being. Notice that this postulate means that a connector does not need to be activated in a single shot: as more and more of its antecedent becomes true, the connector becomes incrementally closer to being activated. When the connector is of the form $\mathsf{T}(\mathsf{true}, u)$, then it is fully activated. For such a connector, failure by the trustee to complete the connector is tantamount to a betrayal of trust.

T_3. PARTITION A CONNECTOR. $\mathsf{T}(r, u \wedge v) \wedge \neg u \rightarrow \mathsf{T}(r, u)$

In general, if you trust a trustee for two propositions, you trust it for each of the propositions. In other words, you would expect to be able to partition a connector into its components. However, the obvious formulation $\mathsf{T}(r, u \wedge v) \rightarrow \mathsf{T}(r, u)$ is inconsistent with T_1, because if u holds, T_1 would eliminate $\mathsf{T}(r, u)$. Since T_1 is fundamental to capturing an active connector, we include $\neg u$ on the left-hand side in T_3. Thus a connector partitions into component connectors as long as none of the components have already been completed. For example, if you trust a merchant to send both the goods you ordered and a warranty, then you trust the merchant to send you the goods—unless the goods are already sent.

4.2 Postulates for Connector Integrity

These postulates describe the integrity of connectors.

T_4. AVOID CONFLICT. $\mathsf{T}(r, u) \rightarrow \neg\mathsf{T}(r, \neg u)$

A connector cannot both ask for and prevent the same thing. This postulate is stronger than merely stating that a connector for a logical impossibility cannot exist, which would be formalized as $\neg\mathsf{T}(r, \mathsf{false})$. However, in the presence of T_8, AVOID CONFLICT is the same as $\neg\mathsf{T}(r, \mathsf{false})$.

T_5. NONVACUITY. From $r \vdash u$ infer $\neg\mathsf{T}(r, u)$

Since $r \vdash u$, if r holds so does u. Or, $\mathsf{T}(r, u)$ completes as soon as it is activated, and is thus vacuous. Because $r \vdash r$, we have $\neg\mathsf{T}(r, r)$. The intuition is that a nonvacuous connector must not require an antecedent stronger than its consequent. The architectural implication of a vacuous connector is that we might as well disconnect the two agents, because the trustee would deliver no value to the truster.

T_6. TIGHTEN. From $\mathsf{T}(r, u), s \vdash r, s \nvdash u$ infer $\mathsf{T}(s, u)$

Any connector that holds for a weaker antecedent also holds for a stronger antecedent. In other words, we can always broaden a connector in the logical ways specified. For example, if you trust your customer will pay you $1 if you give them a coffee, then you can safely trust they will

pay you \$1 if you give them a coffee and a cookie. Some useful consequences are $\mathsf{T}(r \vee s, u) \rightarrow \mathsf{T}(r, u)$, $\mathsf{T}(r, u) \rightarrow \mathsf{T}(r \wedge s, u)$, and $\mathsf{T}(\mathsf{true}, u) \rightarrow \mathsf{T}(r, u)$.

Note that $p \vdash q$ means we can prove q from p: this is stronger than implication $p \rightarrow q$, which holds merely if p is false. Clearly, $\mathsf{T}(r, u) \wedge \neg s \rightarrow \mathsf{T}(s, u)$ is bogus, i.e., we would not conclude $\mathsf{T}(s, u)$ simply because s happens to be false.

4.3 Postulates for Connector Structure

These postulates describe structural properties.

T_7. COMBINE ANTECEDENTS. $\mathsf{T}(r, u) \wedge \mathsf{T}(s, u) \rightarrow \mathsf{T}(r \vee s, u)$

To the left of the $\rightarrow$ are two connectors, together meaning that the truster expects the trustee to do u if r or if s hold, which is the connector on the right. Hence, this broadens a connector, in contrast with T_6.

T_8. COMBINE CONSEQUENTS. $\mathsf{T}(r, u) \wedge \mathsf{T}(r, v) \rightarrow \mathsf{T}(r, u \wedge v)$

Combine consequents of connectors between the same truster and trustee with the same antecedent. The truster would become committed to u and to v if r holds, which is the meaning of the connector on the right. For example, if you trust a merchant to give you an item for your payment and a warranty for the same payment, then you can expect both the item and the warranty for your payment. This postulate relies upon the propositions being not temporally indexed, as Section 4 explains.

T_9. INFERENCE CHAIN. From $\mathsf{T}(r, u), u \vdash s, \mathsf{T}(s, v)$ infer $\mathsf{T}(r, v)$

Assume you trust someone to bring about u if r and to bring about v if u. Then, you trust them to bring about v if r. T_9 generalizes the above intuition to when $u \neq s$. Here we have a situation where the connectors being chained exist between the same truster and trustee pair. The situation becomes more interesting with teamwork, as in T_{17}.

4.4 Postulates for Connector Meaning

These postulates pertain to the content of trust, especially as it relates to commitments [18]. These are important because in some respects commitments are the flip side of trust.

T_{10}. EXPOSURE. $\mathsf{C}_{x,y}(r, u) \rightarrow \mathsf{T}_{y,x}(r, u)$

A debtor is exposed when the creditor of the commitment trusts the debtor for the same content as the given commitment. Now the debtor cannot cancel the commitment without betraying the trust the creditor placed in it. This signifies architectural minimality in that a commitment is being included in a multiagent system only if there is a trust relationship that relies upon the commitment.

T_{11}. TRANSIENT ALIGNMENT. $\mathsf{T}_{x,y}(r, u) \rightarrow \mathsf{C}_{y,x}(r, u)$

A creditor and debtor of a commitment are aligned when if the creditor trusts the debtor for something, the debtor is committed to bringing it about. That is, the connector between the debtor and creditor is covered. This postulate relates to Chopra and Singh's [2] notion of commitment alignment, although their notion considers commitments alone.

T_{12}. WELL-PLACED TRUST. $\mathsf{T}_{x,y}(\mathsf{true}, u) \rightarrow \mathsf{R}u$

This says that whenever a truster trusts a trustee, the consequent comes true on the real path. The success may be incidental, but the trust is not betrayed.

T_{13}. WHOLE-HEARTED ALIGNMENT.

$\mathsf{T}_{x,y}(s, v) \rightarrow \mathsf{R}(s \rightarrow (\mathsf{C}_{y,x}(s, v) \mathsf{U} v))$

When a truster connects to a trustee, the trustee commits (as debtor) to the truster for the relevant propositions *and* remains committed until success. Thus success is achieved, but as an outcome of the debtor's persistent commitment, not incidentally. Thus, this postulate describes a stronger connector than does TRANSIENT ALIGNMENT.

The formulas below are not suitable to be asserted as constraints, but describe important situations. They could be used for problem diagnosis or in engineering effective systems.

Unexercised connector. $\mathsf{T}(r, u) \wedge \mathsf{R}\neg r$. This indicates a connector that is never activated. For example, you may trust that your banker will loan you money if you apply for one, but you may never file the requisite application.

Misplaced trust. $\mathsf{T}(r, u) \wedge \mathsf{R}\neg u$ A connector may fail because when it is activated, the trustee fails to deliver the consequent. Notice that the trustee may never have committed with respect to this connector: therefore, the trustee cannot be faulted for noncompliance.

4.5 Postulates Involving Multiple Agents

These postulates provide a basis for architecting multiagent settings such as teams. They can be thought of as specifying the structures of different types of teams in logical terms, based on the trust relationships among the members. Since, in intuitive terms, trust is an important aspect of teams, we take this to be a promising theme. Below, $\langle x, y \rangle$ represents a simplified team consisting of x and y.

T_{14}. MUTUAL PROGRESS.

$\mathsf{T}_{x,y}(r, u) \wedge \mathsf{T}_{y,x}(u, r) \rightarrow \mathsf{T}_{x,\langle x,y \rangle}(\top, r \wedge u)$

When two agents trust each other reciprocally, each of them trusts their team to make progress on both propositions. This postulate arises commonly in instances of teamwork, including successful business interactions, where each participant concedes to the other, thereby achieving progress. We can think of it as a strengthening of reciprocal dependence [7]. Trust in this sense also provides a complementary aspect to commitments in understanding concession [24].

T_{15}. TRUSTEE'S TEAM. $\mathsf{T}_{x,y}(r, u) \rightarrow \mathsf{T}_{x,\langle y,z \rangle}(r, u)$

Participation by the trustee in a team does not alter the truster's placement of trust in it. This can be thought of as describing cooperative teams in which any conflicts are resolved. For example, if z conflicted with y and prevented y from being trustworthy for u, then the above postulate would not hold for the team $\langle y, z \rangle$. In other words, the connector between the truster and trustee applies equally to the team including the trustee. For example, if you trust your local postman to deliver your mail, you can trust the local post office to deliver your mail. This inference applies when participation in the team does not alter the nature of the connection. For example, you can trust your friend to take your side in a dispute, but not against his employer.

T_{16}. TRUSTER'S TEAM. $\mathsf{T}_{x,y}(r,u) \rightarrow \mathsf{T}_{\langle x,z\rangle,y}(r,u)$

In contrast with T_{15}, here the connector applies to any team that the truster may belong to.

T_{17}. PARALLEL TEAMWORK.

$$\mathsf{T}_{x,y}(r,u) \wedge \mathsf{T}_{x,z}(u,v) \rightarrow \mathsf{T}_{x,\langle y,z\rangle}(r, u \wedge v)$$

When a truster connects to two trustees, the truster connects to their team as a composite trustee. For example, if you trust one friend to bring you bread and one to bring you soup, you trust them as a team to bring you bread and soup. This postulate is an alternative to T_9 (INFERENCE CHAIN) and shows how the connectors to two trustees can be combined.

T_{18}. PROPAGATE.

$$\text{From } \mathsf{T}_{x,y}(r,u), \mathsf{T}_{y,z}(s,v), v \vdash u, r \vdash s \text{ infer } \mathsf{T}_{x,\langle y,z\rangle}(r,v)$$

Here, x trusts y and y trusts z. Because of how the antecedents and consequents mutually relate, x trusts $\langle y,z\rangle$.

4.6 Postulates Involving Dynamism

The postulates involving updates are largely heuristic in nature. The following illustrate three aspects of dynamism: these deal with persistence when nothing changes; reduction in trust ratings when trust is betrayed; and enhancement in ratings when trust is kept. The intuition behind these is based on the notion of relational or trust capital [7], which agents can build up through trustworthy behavior and drain through untrustworthy behavior.

T_{19}. PERSISTENCE. $\mathsf{T}(r,u) \rightarrow \mathsf{T}(r,u)\mathsf{U}(u \vee r)$

A truster persists in its connector unless it acquires evidence that the connector has failed or completed. That is, a connector persists at the same strength as long as the connector is not activated (until r holds), meaning that the substantive aspect of the trust has not been exercised, or the connector has not been completed (until u holds). Assume you trust a merchant to deliver if you pay, i.e., as $\mathsf{T}(\mathsf{pay}, \mathsf{deliver})$. If you have not paid, then your not receiving a delivery should not affect your trust in the trustee.

Notice that the above postulate is silent about success or failure. Below, skepticism and faith identify domain-specific notions, outside our language, of how a truster respectively reduces or increases its level of trust in a trustee.

T_{20}. SKEPTICISM.

$$\mathsf{skepticism}_{x,y}(s,v) \rightarrow (\mathsf{T}(r,u) \wedge r \wedge \neg u) \rightarrow \neg\mathsf{T}(s,v)$$

A truster lowers its trust in a trustee if the trustee fails for an activated connector, i.e., one whose antecedent has been achieved. This can be thought as an agent narrowing or weakening its connectors with another agent based on the second agent's performance.

T_{21}. FAITH. $\mathsf{faith}_{x,y}(s,v) \rightarrow (\mathsf{T}(r,u)\mathsf{U}u) \rightarrow \mathsf{T}(s,v)$

A truster adjusts its trust in a trustee based on whether the trustee achieves the consequent. This can be thought of as an agent broadening or strengthening its connectors with another based on the second agent's performance.

In addition, we can compare trust ratings as follows.

Compare ratings. The expression $\mathsf{T}_{x,y}(r,u) \wedge \mathsf{T}_{x,w}(r, u \wedge v)$ signifies that x trusts y less than it trusts w. This reflects some intuitions of Falcone et al.'s [8] contracting approach. The deeper underlying intuition is that sets of possible paths (being different outcomes) map naturally to probabilities.

5. APPLYING THE THEORY

Let us consider a cross-organizational scenario of auto insurance claims [21], which relates naturally to multiagent systems. Figure 1 (from [21]) describes the intended operations in this scenario, which deals with auto insurance claims processing by AGFIL, an insurance company. Interestingly, this figure omits the policy holder whom the scenario serves. A policy holder, John Doe, is in an accident and files a claim with Europ Assist, who runs AGFIL's call center. Europ Assist identifies a mechanic shop (garage) in consultation with Doe, sends Doe there, and forwards his claim to AGFIL. AGFIL passes the claim to Lee Consulting Services (Lee CS), which interacts with Doe to complete the claim, obtains estimates from the mechanic, and decides whether to honor Doe's claim. Skipping ahead a few steps, this episode would normally end with the mechanic repairing Doe's car and getting paid by AGFIL.

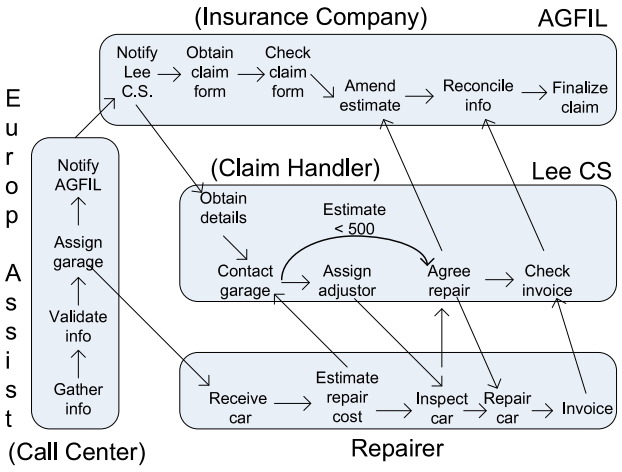

Figure 1: Insurance scenario modeled operationally

The traditional low-level representation emphasizes the steps performed by each party and their mutual control flow. It provides no support for meaning. Desai et al. [6] formalized this scenario in terms of commitments, identifying the contractual business relationships among the parties involved. However, such relationships are founded on a substrate of trust. An additional benefit of modeling trust is that it focuses on the architecture, which we can use as a basis (in an engineering methodology) for determining the necessary contractual relationships.

Let us consider the following examples. First, not only do Lee CS and AGFIL have commitments toward one another, they must also trust one another to perform accordingly. Second, the importance of trust becomes more important when we consider architectures that are not highly regimented. For example, when John Doe talks to Europ Assist, out of the many mechanics who are preapproved, Doe would select one of those whom he deemed trustworthy,

because the existence of commitments does not adequately characterize the outcomes, although the existence of a commitment by AGFIL to ameliorate a failed interaction with a preapproved mechanic may be a reason to place greater trust in the mechanic. Third, when the system in question is open, i.e., John Doe can have his car seen by any mechanic, the importance of trust goes up further.

In each of these cases, the participants would apply some of the above reasoning postulates. For example, Doe would ACTIVATE his dependence on the mechanic by bringing his car in for repairs (T_2); the mechanic would COMPLETE the dependence by repairing the car (T_1); the mechanic gives Doe a loaner car for a week: the loaner is PARTITIONED from the repair itself via (T_3); under T_7, Doe can COMBINE his dependence on the mechanic to trust the mechanic to repair the car whether Doe brings it in or asks the mechanic to tow it to his shop. Under PERSISTENCE (T_{19}), the mechanic holds his trust in being paid in a timely fashion by AGFIL until he submits a bill or gets paid. Doe and the mechanic demonstrate whole-hearted alignment (T_{13}) because the mechanic remains committed to completing the repairs until he does so. Doe applies PARALLEL TEAMWORK (T_{17}) to place his trust in the team consisting of AGFIL, Lee CS, and the mechanic to process his claim.

The foregoing points illustrate the kinds of reasoning involving trust, which can be used as criteria for judging specialized trust approaches. Existing approaches do not readily apply in the above kinds of settings: they either (1) make unrealistic assumptions about their models or (2) fail to support inferencing. In the first category we place approaches for adopting beliefs from reports [4, 12], which are simply inapplicable because trust here (and often) is about actions, not truthfulness; cognitive approaches, which presume deeper representations of beliefs and plans than may hold in practice [1, 5]; current heuristic [9] and probabilistic [11, 23] approaches, which do not provide the essential logical structure for this case (thus making it difficult to use them architecturally). In the second category, we place the social approaches to trust [7, 20] and dependence [10] which, though conceptually suited in principle to architecture, are mostly informal in their details.

More importantly, we can characterize the trust relationships among the parties with or without any contractual relationships among them. Specifically, in the above setting, we can define an auto repair ecosystem in which a party's dependencies can be expressed as trust, and reasoned about to determine if the ecosystem will prove effective: for example, if the respective dependencies are supported by capabilities or commitments of the agent's involved.

Architecture

More generally, an architecture is described not only by its components and connectors but also by its constraints and styles [17]. We propose an approach that enables specifying architectures for specific multiagent systems:

Components: Application-specific roles, such as mechanic and call center.

Connectors: The trust relationships between the roles: a connector better reflects a flow of trust not just a flow of information, as in traditional approaches. For example, the mechanic trusts AGFIL to pay for repairs.

Constraints: The reasoning postulates discussed in the foregoing. Of these, the integrity and structure constraints are of broad use; some of the others would apply in specific settings. For example, if Lee CS arranges to take care of Doe's car, the mechanic and Doe may have no direct connectors to each other.

Styles: The sets of constraints geared toward different applications. For instance, teamwork is a kind of architectural style. For example, the mechanic and policy holder may trust each other reciprocally; or the mechanic and policy holder may trust a common party, such as Lee CS or AGFIL.

One can imagine a design episode based on the above architecture. Here the designers would identify the key roles in their system-to-be, and identify the trust relationships among the (agents playing these) roles. Such trust relationships would describe the system in architectural terms. Upon further refinement, the designers could identify the commitments among the roles that would help realize the trust interactions. These could arise partly by (1) engendering trust (John Doe might trust a mechanic to complete a task after the mechanic commits to doing so) and (2) partly by yielding trust by fiat (Doe would not trust any arbitrary mechanic but a commitment from AGFIL or Lee CS to get Doe's car repaired would produce trust in an approved mechanic or limit Doe's liability and thus reduce the need for such trust). Trust as dependence can thus conceptually precede commitments. In other words, we would first identify the necessary trust relationships and then induce commitments that would support such trust. Trust is thus complementary to goal-based approaches such as Tropos, which capture dependencies between goals. Further, it can help address some of the challenges of high variability that recent work on Tropos has identified [16].

As Singh and Chopra [19] observe, recent agent-oriented software engineering approaches either follow mentalist models based on beliefs and intentions (and are thus ill-suited for multiagent architecture, since they inevitably describe an agent's internal state), or adopt low-level ideas from traditional software engineering (and are thus ill-suited for multiagent systems). Trust, as we have formalized it here, can help provide a systematic basis for including the mentalist concepts by showing how they may relate to the high-level architecture of a multiagent system.

6. DISCUSSION

The above approach considers trust in propositional terms. Most practical settings need parameters, which we can accommodate in a fairly straightforward manner. Similarly, an expansion to graded or measured notions of trust would be valuable. We can potentially develop such a notion by adopting some ideas of Demolombe [5]. Indeed, there is a conceptually straightforward mapping of our models to the above, which would arise by assigning relative weights to the sets of runs that our model-theoretic standard of trust $\mathbb{T}$ identifies. When such sets of runs can be assigned likelihoods of occurrence, they can additionally be used as a basis for a probabilistic definition of trust.

Trust is inherently contextual. As a result, in some uses the preconditions that apply on a claim of trust may not be explicit. Such implicit preconditions can be mapped to

antecedents in an explicit representation. Organizational context is particularly relevant from our architectural perspective: an agent may depend upon another when they are both part of the same team or organization.

Following a similar distinction for commitments [18], we can distinguish two main kinds of trust: (1) *dialectical*, i.e., about assertions or arguments relating to reports [4, 12]; or (2) *practical*, i.e., about actions, as in the present paper. We can relate the above dichotomy to trust in an agent viewed as a service provider and an agent viewed as a referrer. Examples are "if the interest rate has fallen, (I trust) my banker to grant my mortgage application (practical) or (I trust) my banker's assertion of my new loan payment (dialectical).

Following the spirit of correspondence theory as proposed by van Benthem [22], the above postulates can be given a model-theoretic basis wherein for each postulate we state a corresponding semantic constraint (in essence, a closure property) on the model. For reasons of space, we defer such constraints and theorems to a longer version of this paper.

Directions

Some important directions of future work fall out naturally from the above formal, architectural development of trust.

In *conceptual terms*, a deeper study of the reasoning postulates would be beneficial in a wide range of multiagent applications. In particular, it would be important to determine additional architectural styles. We considered simplistic multiagent systems above. This is an important start in formalizing trust, but it would be valuable to expand on this theme to specify richer systems and postulates about them. Specifically, above we treated agents as either individuals or sets of agents. In general, multiagent systems would demonstrate rich structures and consist of roles that feature in a variety of operational and institutional relationships with each other. Such relationships would naturally bear a significant impact on trust understood architecturally.

In *theoretical terms*, a rich formal language for expressing constraints and reasoning about them to determine if a particular architecture style or instance will satisfy desirable properties such as a guarantee of progress under appropriate assumptions on the behaviors of the participants. Makinson and van der Torre [13] introduced the idea of input-output logics as a general way to treat conditionalization. Our approach can be thought of as specializing their ideas for the setting of trust with inferences for completion, commitments, and teamwork that do not arise with conditionals in general, but are important for an understanding of trust. It would be interesting to explore what insights we can adopt from input-output logics.

In *practical terms*, an important consideration is of a pattern language for expressing architectures. Such a language could provide a basis for a tool and methodology for specifying architectures. A greater goal is to develop an extensive approach for *service-oriented computing* in the broadest sense of the term that considers not technical (web or grid) services as emphasized today but service engagements mediated by flexible and expressive trust relations.

Acknowledgments

Thanks to the anonymous referees and to Amit Chopra for helpful comments. Thanks to the Army Research Laboratory for partial support under Cooperative Agreement Number W911NF-09-2-0053.

7. REFERENCES

[1] C. Castelfranchi, R. Falcone. Principles of trust for MAS. *ICMAS*, pp. 72–79, 1998.

[2] A. K. Chopra, M. P. Singh. Constitutive interoperability. *AAMAS*, pp. 797–804, May 2008.

[3] P. Dasgupta. Trust as a commodity. In D. Gambetta, ed., *Trust: Making and Breaking Cooperative Relations*, ch. 4, pp. 49–72. 2000.

[4] M. Dastani, A. Herzig, J. Hulstijn, L. van der Torre. Inferring trust. *AAMAS CLIMA*, *LNCS* 3487, pp. 144–160. Springer, 2004.

[5] R. Demolombe. Graded trust. *AAMAS Trust*, pp. 1–12, 2009.

[6] N. Desai, A. K. Chopra, M. P. Singh. Amoeba: A methodology for modeling and evolution of cross-organizational business processes. *ACM TOSEM*, 19(2):6:1–6:45, October 2009.

[7] R. Falcone, C. Castelfranchi. From dependence networks to trust networks. *AAMAS Trust*, 2009.

[8] R. Falcone, G. Pezzulo, C. Castelfranchi, G. Calvi. Contract nets for evaluating agent trustworthiness. *Proc. 6th & 7th Trust Workshops*, *LNCS* 3577, ch. 3, pp. 43–58. Springer, 2005.

[9] K. Fullam, K. S. Barber. Dynamically learning sources of trust information. *AAMAS*, pp. 1062–1069, 2007.

[10] M. Johnson, J. M. Bradshaw, P. Feltovich, C. Jonker, M. B. van Riemsdijk, M. Sierhuis. Coactive design. *AAMAS COIN Workshop*, pp. 49–56, 2010.

[11] A. Jøsang. A subjective metric of authentication. *ESORICS*, *LNCS* 1485, pp. 329–344, 1998. Springer.

[12] C.-J. Liau. Belief, information acquisition, and trust in multi-agent systems. *Art. Intell.*, 149(1):31–60, 2003.

[13] D. Makinson, L. van der Torre. Input-output logics. *J. Philosophical Logic*, 29:383–408, 2000.

[14] R. Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.

[15] D. Parnas. Information distribution aspects of design methodology. *Proc. IFIP*, TA-3, pp. 26–30, 1971.

[16] L. Penserini, A. Perini, A. Susi, J. Mylopoulos. High variability design for software agents: Extending Tropos. *ACM TAAS*, 2(4):16:1–16:27, November 2007.

[17] M. Shaw, D. Garlan. *Software Architecture*. Prentice-Hall, 1996.

[18] M. P. Singh. Semantical considerations on dialectical and practical commitments. *AAAI*, pp. 176–181, 2008.

[19] M. P. Singh, A. K. Chopra. Programming multiagent systems without programming agents. *ProMAS 2009 Workshop*, *LNAI* 5919, pp. 1–14. Springer, 2010.

[20] P. Sztompka. *Trust: A Sociological Theory*. Cambridge University Press, 1999.

[21] C. J. van Aart et al. Use case outline and requirements. IST CONTRACT Project, 2007.

[22] J. F. A. K. van Benthem. Correspondence theory. In D. Gabbay, F. Guenthner, eds., *Hbk Phil. Log*, vol. II, pp. 167–247. Reidel, 1984.

[23] Y. Wang, M. P. Singh. Formal trust model for multiagent systems. *IJCAI*, pp. 1551–1556, 2007.

[24] P. Yolum, M. P. Singh. Enacting protocols by commitment concession. *AAMAS*, pp. 116–123, 2007.