Inferring Commitment Semantics in Multi-Agent Interactions

Paula Chocron
IIIA,CSIC and Universitat Autònoma de Barcelona
Barcelona, Spain
pchocron@iiia.csic.es

Marco Schorlemmer
IIIA,CSIC
Barcelona, Spain
marco@iiia.csic.es

ABSTRACT

Commitments are a useful abstraction to specify the social semantics of multi-agent communication languages. To use them in open and heterogeneous systems, it is necessary to develop solutions to the problem of interoperability, an effort that has already provided methods to, for example, align commitments between interlocutors. In this paper we consider the problem of commitment semantics inference, which can be summarized as follows: how can an agent that arrives to a community with an established language discover its social semantics, only by observing interactions? We introduce a method based on simple learning techniques that tackles this problem. We show that the basic commitment semantics is not possible to infer, and discuss different ways of enriching it that make inference feasible. We show experimentally how our technique performs for each of these extensions.

KEYWORDS

Commitments; Semantic Inference; Multi-Agent Communication

ACM Reference Format:
Paula Chocron and Marco Schorlemmer. 2018. Inferring Commitment Semantics in Multi-Agent Interactions. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 9 pages.

1 INTRODUCTION

Multi-agent systems make possible the interaction between participants that have different backgrounds, knowledge, and abilities, allowing for rich collaboration situations that harness this diversity. In such open communities, ensuring that all participants understand language in the same way is practically impossible. Establishing one central, fixed language can work for small and closed communities in which participants have no exchanges with foreign agents. In open situations, however, external agents who do not know the language may join the community at any time. In order to integrate these new agents in the system, it is necessary to provide them with techniques to learn the language. This can be challenging, in particular since language in open communities is likely to change and evolve, making fixed specifications obsolete. The situation therefore requires the development of techniques to learn a new language in a dynamic way.

Which techniques can be used to learn a language depends on how this language is specified. The question of what should be expressed by an interaction specification has been extensively discussed in the multi-agent systems community. Traditional specification techniques such as Finite-State Machines [2] or logic-based ones [9] determine what can be said and when, describing the flow of interactions. As a complement (or alternative) to these techniques, Singh proposed in [14] the notion of *commitment semantics*, which focuses on the social semantics of utterances. A commitment is a relation between a debtor and a creditor, where the first commits to do something if the creditor does something else. The meaning of expressions is, then, defined in terms of operations over commitments. For example, a specification could determine that when an agent says *Offer*, it is creating a commitment to make an item available if the receiver of the message pays for it.

Commitments are useful because they capture social semantics without over-specifying other aspects such as the order in which messages are uttered. Naturally, a commitment specification is only useful if it is shared by all the interlocutors. If one of the agents has the specification above for *Offer*, while another one thinks it is enough to acknowledge the offer to obtain the item, it is unlikely that they will have a successful transaction. The community working on commitment semantics has already considered different aspects of the interoperability problem. For example, Chopra and Singh propose an adequate definition of interoperability, based on the commitments perceived by each agent [5]. Other work considers commitment misalignment, which can occur in an asynchronous environment where messages can be lost or delayed [6].

In this work we tackle a problem that, to the best of our knowledge, remains unexplored. We consider a foreign agent that arrives to a community whose commitment specification it does not know. There is no shared meta-language, so the agent cannot ask directly what words mean. However, it can observe interactions between other agents, who do know the community's vocabulary and its social semantics. In this paper we analyze under which conditions the agent can infer the commitment specification with only this information. We use a simple probabilistic technique, in which the agent analyses possible meanings in the interactions it observes. This method is similar to the ones in [3, 4], where agents infer alignments between vocabularies from the experience of interacting, although they do not use commitment semantics. We show how when using a basic commitment semantics agents can learn a specification that is useful to interact, but which is not necessarily the correct one. Incorporating simple semantic extensions is enough to infer the actual specification. We evaluate all the methods with simulations using randomly generated protocols.

In the following section we describe possible scenarios in which the techniques in this paper would be useful. In Section 3 we describe formally the language to specify commitments. Section 4 presents a probabilistic technique to discover social meanings from

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

observing interactions between two agents. The problem is particularly challenging because the possibility of operating with commitments adds significant uncertainty. Section 5 shows experimentally that the basic commitment semantics does not allow to infer meaning correctly. Section 6 discusses different extensions, all of which were proposed in the literature on commitments for multi-agent systems, that enable agents to learn the specification.

2 SCENARIOS

The general situation that we consider in this paper can be described as follows. We assume there exists an established community with a shared commitment specification that guides interactions between their members. This specification defines which utterances trigger which operations over commitments. To interact successfully, agents must know this specification. Otherwise they may unintentionally engage in commitments without complying with them later. We consider a foreign agent that wants to interact with agents in the community, but which does not have access to the specification. This can happen for different reasons. First, the agent may be unable to ask for it. If it knows nothing about the foreign language, it may not even be able to ask somebody else for the specification. Second, even if the specification is available, it may be impossible to understand for the new agent. This is because the words or symbols used to describe meanings can be different from the ones the agent would use. To make an analogy, this would be like giving a dictionary with definitions in English to someone who does not speak any English. Finally, the specification may not exist as something that can be shared. Social conventions are dynamic and fast-evolving, and they are not necessarily always explicitly written down. The fact that agents can follow a convention does not necessarily mean that they can make it explicit.

The technique that we propose allows the foreign agent to infer the commitment semantics used by the community only by observing interactions between interlocutors who already know it. In this way, the agent can start interacting only when it is sure that it knows the semantics (because all the interactions it sees adjust well to the specification it inferred). We do require that the agent can observe many interactions, particularly when vocabularies are large. We identify three types of scenarios where this is possible.

Scenarios with many available interactions. This includes any kind of open environment with many participants and public interactions. An example are the auction or market systems described in [8]. An agent who wants to start participating in an auction community has, in addition to the reasons mentioned before, an extra incentive to avoid asking for definitions: in such a competitive environment, it could be deceived. Observing public interactions between other agents is a way of learning the semantics discretely and efficiently.

Scenarios in which a log is available. In this case an agent tries to enter a smaller, closed community with well-defined internal semantics. An example of such a community are the groups of parents described in [11], where members interact with each other to collaborate in everyday tasks. If the community gives the agent access to a log of interactions, it could learn the semantics before starting to participate.

Scenarios with emergent semantics. An agent that already knows the language of a community can use the techniques that we propose to understand new meaning that is created dynamically by other interlocutors. This kind of meaning is known as *emergent semantics* [1] and it is difficult to capture since it is usually not explicit. Using our techniques an artificial agent could infer, for example, the commitment meaning of expressions that appear spontaneously in social media communities only by analyzing their use. Since the agent only needs to learn the meaning of a small vocabulary, it does not require that many interactions.

3 COMMITMENT SPECIFICATIONS

A semantics based on commitments describes the social effect that utterances have without imposing any restriction on how the flow of the interaction will be. A commitment is an abstract concept relating two agents (debtor and creditor) and two propositions (antecedent and consequent). The intuitive meaning of a commitment is that, when it holds, the debtor is obliged to bring about the consequent if the creditor enforces the antecedent. As an example, the commitment created by *Offer* that we mentioned in the introduction could have agent a_1 as a debtor, a_2 as a creditor, and the actions of paying and delivering an item as antecedent and consequent, respectively. Agents can operate with commitments, creating, deleting, or even assigning them to other agents; they do this by sending to each other messages that are associated with the operations.

The syntax and semantics of (operations over) commitments have been formalized in multiple ways. Originally, commitments were defined over a propositional language, and the antecedent and consequent were propositions on this language. A separated vocabulary was used to communicate the operations. We will, instead, follow the approach proposed in [12]. In this work, commitments are defined over the same events that trigger them. That means that an agent can, by performing an action u, commit to perform v if the creditor performs w. We will use this idea considering events to be messages that are exchanged between agents. In this way, agents operate over commitments by sending messages, and at the same time commitments are defined over the utterance of words. That is, a commitment with v as antecedent and u as consequent means that the creditor must say u if the debtor says v. In the example above, the creation of the commitment to buy an item is triggered by uttering the word *Offer*, and it says that, if the debtor says *Pay*, the creditor must say *Item* later. In this way, we provide a general approach by keeping all the specification at the level of utterances.

In this section we describe the syntax and the semantics of our language to specify commitments. While the idea is heavily inspired by the work in [12], we present it in a different way that will ease the exposition of the inference techniques. We also made some simplifications that make the inference more approachable. Concretely, we allow words to have only one meaning, and the antecedent and consequent of a commitment are single messages. We also consider interactions between two agents only.

3.1 Syntax

A *commitment specification* relates words in a vocabulary with their social meaning, that is, with operators over commitments. We consider three of the basic commitment operators: *Create*, *Cancel*, and

Release [19]. Intuitively, *Create* initiates a commitment contract in which a debtor commits to say something (the consequent) if the creditor says something else (the antecedent). Both *Cancel* and *Release* are ways of finishing the commitment without necessarily uttering the consequent. The difference between them is that *Cancel* is uttered by the debtor, while only the creditor can *Release* a commitment. Since we only consider interactions between two agents, we do not include the *Assign* and *Delegate* operators that only make sense in larger communities. From now on, we use a vocabulary $\mathcal{V}$ and a set of agent IDs $\mathcal{A} = \{a_1, a_2\}$. We will refer with a and a' (assuming that $a \neq a'$) to any agent in $\mathcal{A}$.

Definition 3.1. Let $v, v' \in \mathcal{V}$ be two words. A *commitment operation* is a term $op(v, v')$ where $op \in \{Create, Release, Cancel\}$. We call $O_{\mathcal{V}}$ the possible commitment operations over words in vocabulary $\mathcal{V}$.

A specification is a function between $\mathcal{V}$ and the set of commitment operations. Not all words in the vocabulary need to have a commitment meaning, so we include a *None* term to the co-domain of the function.

Definition 3.2. A *specification* over $\mathcal{V}$ is a total function **means** : $\mathcal{V} \to O_{\mathcal{V}} \cup \{None\}$

The **means** function relates words with their social meaning only, letting us focus on the inference of that kind of semantics in particular. Of course, words in the vocabulary could have additional meanings. For example, there could exist a physical dimension of semantics that relates words with events, in which a word *Pay* corresponds to a transfer to a bank account. The problem of learning correspondences between words and an observable physical meaning has been extensively studied (see [16], for example, for a cognitive approach) and we do not consider it here.

Example. We will use the example we already mentioned to illustrate concepts throughout this paper. From now on, we will refer to it as the *transaction* example. Consider a vocabulary $\mathcal{V} = \{Offer, Withdraw, Complain, Pay, Receipt, Item, ReturnMoney, Reject\}$. The following one is a possible commitment specification:

$$\begin{aligned}
\mathbf{means}(Offer) &= Create(Pay, Item)\\
\mathbf{means}(Item) &= Create(Complain, ReturnMoney)\\
\mathbf{means}(Withdraw) &= Cancel(Pay, Item)\\
\mathbf{means}(Reject) &= Release(Pay, Item)
\end{aligned}$$

And for the rest of $v \in \mathcal{V}$, $\mathbf{means}(v) = None$.

Let us remark two aspects of the proposed commitment specification language. First, it allows for embedded commitments. For example, the definition of *Offer* uses *Item*, which creates a commitment itself (to return the money if the agent who paid complains). Second, operations over commitments do not have a debtor and a creditor. Instead, this information is implicit and depends on who utters the word and to whom, as we will see below. Intuitively, if a_1 sends *Offer* to a_2, the commitment will be created with a_2 as creditor and a_1 as debtor.

3.2 Semantics

The semantics of the specification language describes how agents can operate with commitments by sending messages to each other. Our ultimate goal is to define when an interaction between two agents *complies* with a specification, that is, when the agents have satisfied all the commitments they made. Before presenting the operational semantics, let us introduce the notion of messages and interactions.

Definition 3.3. Let $a, a' \in \mathcal{A}$ ($a \neq a'$) and $v \in \mathcal{V}$. A *message* is a tuple $m = \langle a, a', v\rangle$, where a and a' are the sender and receiver respectively.[1] An *interaction* is a sequence of messages $I = [m_1, \ldots, m_n]$.

Interactions are sequences, and as such any operation over sequences can be applied to them. We will use the following ones: append ($I \,.\, m$, for a message m), length ($len(I)$), indexing ($I[i]$ for some $1 \leq i \leq len(I)$), and subsequence ($I[i : j]$, for $1 \leq i < j \leq len(I)$).

The operational semantics uses the notion of *commitment*. This notion is not explicit in the syntax of the specification language, but only used to define its semantics.

Definition 3.4. A *commitment* is a tuple in the set $C = \mathcal{V} \times \mathcal{V} \times \mathcal{A} \times \mathcal{A}$. The agent IDs represents the debtor and the creditor respectively, and we require them to be different. Two words are the antecedent and the consequence of the commitment.

We will define the semantics of the language by describing the state of each possible commitment for an interaction I and specification **means**. A commitment is *inactive* when the debtor has not created it, *active* when the debtor created it, but its antecedent has not been uttered and *detached* when the debtor created it and its antecedent was uttered by the creditor.

Definition 3.5. Let **means** be a commitment specification over vocabulary $\mathcal{V}$. A *state function* is the function $\sigma : C \times I \to \{active, inactive, detached\}$ that is defined as follows. Let $u, v \in \mathcal{V}$, and c be a commitment (u, v, a_1, a_2). The case when a_2 is the debtor is analogous.

$$\sigma(c, []) = inactive$$

$$\sigma(c, I'.\langle a_1, a_2, w\rangle) = \begin{cases} active & \text{if } \sigma(c, I') = inactive \text{ and}\\ & means(w) = Create(u, v)\\ inactive & \text{if } \sigma(c, I') = detached \text{ and}\\ & means(w) = Cancel(u, v)\\ & \text{or } w = v\\ \sigma(c, I') & \text{otherwise} \end{cases}$$

$$\sigma(c, I'.\langle a_2, a_1, w\rangle) = \begin{cases} inactive & \text{if } \sigma(c, I') = detached \text{ and}\\ & means(w) = Release(u, v)\\ detached & \text{if } \sigma(c, I') = active\\ & \text{and } w = u\\ \sigma(c, I') & \text{otherwise} \end{cases}$$

Our semantics is sequential: if a commitment is detached at some point, it has to be turned inactive, independently of what happened before. This allows for the specification of interactions in which the same commitment is made twice. The state function σ always returns the last state of a commitment.

[1] Since we only consider interactions between two agents, it would be enough to have one agent in the definition, but we keep two for clarity and compatibility reasons.

The states *active, detached* and *inactive* are enough to define the operational semantics of our specification language. We choose to use only these ones to maintain the definition short and clear. However, other notions that appear in the literature will be important for the inference process. Concretely, we will need to distinguish between different ways of turning a commitment from detached to inactive. When the debtor cancels it, we will say it is *canceled*, when the creditor releases it will be *released*, and when the debtor utters the consequent it will be *discharged*.

Finally, we define the notion of *compliance* of an interaction. Intuitively, an interaction complies with a specification if it has no detached commitments, that is, every commitment that was created and detached finished in one of the three possible ways.

Definition 3.6. An interaction I *complies* with a specification **means** if there are no detached commitments, that is, if $\sigma(c, I) = inactive$ or $\sigma(c, I) = active$ for every $c \in C$.

Commitments that are detached at the end of an interaction are normally called *violated*. As we will explain later, we assume all interactions agents see are compliant, so we will not need this distinction.

To illustrate these ideas, consider the buying example, given by the specification above. Consider the following interactions:

$$I_1 = [\langle a_1, a_2, \mathit{Offer}\rangle, \langle a_2, a_1, \mathit{Pay}\rangle]$$
$$I_2 = [\langle a_1, a_2, \mathit{Offer}\rangle, \langle a_2, a_1, \mathit{Pay}\rangle, \langle a_2, a_1, \mathit{Withdraw}\rangle]$$
$$I_3 = [\langle a_1, a_2, \mathit{Offer}\rangle, \langle a_2, a_1, \mathit{Pay}\rangle, \langle a_1, a_2, \mathit{Item}\rangle]$$

The first interaction does not comply with the specification, since the commitment $(\mathit{Pay}, \mathit{Offer}, a_1, a_2)$ is detached. I_2 and I_3 do comply, since the commitment is canceled and discharged, respectively. Note that another commitment is created in I_3 by saying *Item*, but it is never detached.

4 INFERRING SOCIAL SEMANTICS

The problem of inferring the commitment semantics of a community can be formulated as follows. Agents in a community interact following a commitment specification **means** over vocabulary $\mathcal{V}$. Each interaction is between two agents a_1 and a_2, is finite, and complies with the specification. An external agent, called *student*, observes the complete interactions sequentially, one at a time. The student does not know neither the vocabulary $\mathcal{V}$ used by the agents nor the function **means**, but it assumes they are behaving according to some specification like the ones that we described in the previous section. Moreover, the student does share with the rest of the agents the general operational semantics that we described in the previous section. That is, the student may not know that **means**(*Offer*) = $Create(\mathit{Pay}, \mathit{Item})$, but it knows that if it does, then $\sigma(\langle \mathit{Pay}, \mathit{Item}, a_1, a_2\rangle, [\langle a_1, a_2, \mathit{Offer}\rangle]) = active$. This does not imply that the student uses the same words in the specification. It could perfectly well use any other word instead of *Create*; the important part is that they work in the same way.

In this section we present a technique that the student can use to infer the mappings in **means** only by observing interactions between agents that already know the specification.

A Technique to Infer Social Meaning

We propose a probabilistic approach, in which the student maintains a confidence value for each mapping between a word and a commitment operation. We represent these values as a function $\omega : \mathcal{V} \times O_{\mathcal{V}} \rightarrow \mathbb{R}^+$. This function is partial, since the student does not know $\mathcal{V}$ a priori, which also implies it cannot compute all the possible meanings in $O_{\mathcal{V}}$. The values in ω are updated with each observation of a new complete interaction, to reflect new evidence for possible mappings. We call $\mathcal{V}^{\omega}$ the set of $v \in \mathcal{V}$ such that $\omega(v, o)$ is defined for some $o \in O_{\mathcal{V}}$. Similarly, for each $v \in \mathcal{V}$, $O^{\omega}(v)$ are the commitment operations $o \in O_{\mathcal{V}}$ for which $\omega(v, o)$ is defined. These are the meanings considered possible for v.

The update works as follows. For each interaction, the student first computes the possible meanings for a word, and then it updates the value of those mappings in ω. Finding all possible meanings is challenging due to the existence of operations to finish commitments (*Cancel* and *Release*), combined with the fact that the student does not know the complete vocabulary a priori. As an example, consider again the *transaction* specification. Suppose the student observes the interaction $[\langle a_1, a_2, \mathit{Offer}\rangle), \langle a_2, a_1, \mathit{Pay}\rangle, \langle a_1, a_2, \mathit{Withdraw}\rangle]$. Since it does not know a priori that *Item* exists, the student is unable to infer the correct semantics, and it may think that **means**(*Offer*) = $Create(\mathit{Pay}, \mathit{Withdraw})$. Our solution to this problem consists on inferring possible *Create* operations separately, by looking at the possible fulfilled commitments. With this information, the student can learn the rest of the operations.

From now on, we will consider a student that observes an interaction I with at least one message. For simplicity, we will refer to the word uttered in position i (with $1 \leq i \leq len(I)$) as v_i. From now on we will only work with commitments where a_1 is always the debtor. The case when a_2 is the debtor is analogous. First of all, the student computes possible *Create* meanings for each word in I. For an index i, these are the set of all commitment that can have been discharged after it, that we will call $Disch(I, i)$.

Definition 4.1. Let I be an interaction and i an index such that $1 \leq i \leq len(I)$ and $I[i] = \langle a_1, a_2, v_i\rangle$ (everything is analogous for a_2). The set $Disch(I, i)$ contains all the commitments that may have been created by v_i and later discharged. Concretely, a commitment (v_j, v_h, a_1, a_2) is in $Disch(I, i)$ if there exist indexes j and h after i ($i < j < h < len(I)$ and $v_j \neq v_h \neq v_i$) such that $I[j] = \langle a_2, a_1, v_j\rangle$ and $I[h] = \langle a_1, a_2, v_h\rangle$.

As an example, consider the interaction $I = [\langle a_1, a_2, \mathit{Offer}\rangle, \langle a_2, a_1, \mathit{Pay}\rangle, \langle a_1, a_2, \mathit{Receipt}\rangle, \langle a_1, a_2, \mathit{Item}\rangle]$. The set of possible discharged commitments for index 1 is $Disch(I, 1) = \{(\mathit{Pay}, \mathit{Receipt}, a_1, a_2), (\mathit{Pay}, \mathit{Item}, a_1, a_2)\}$. The set of possible discharged commitments for index 2 is empty, because since a_2 spoke only once, it cannot have discharged any commitment.

To update possible *Create* meanings, the student rewards the mappings between each message v_i and the creation of all the commitments in $Disch(I, i)$. Let ρ_1 be a reward parameter. For all $1 \leq i < len(I)$, and for all $(v_j, v_h, a_1, a_2) \in Disch(I, i)$, let $o = Create(v_j, v_h)$:

$$\omega(v_i, o) \leftarrow \begin{cases} \omega(v_i, o) + \rho_1 & \text{if } o \in O^{\omega}(v_i) \\ \rho_1 & \text{otherwise} \end{cases}$$

The process to update *Release* and *Cancel* operations is similar. First, the student generates possible meanings. Words that are candidates to mean *Release* are those that are uttered by the creditor of a commitment after the commitment was created and detached. *Cancel* candidates are the analogous ones that were uttered by the debtor of the commitment. Of course, the student ignores a priori which *Create* operations were actually uttered. For this reason we consider all *Create* meanings that are possible for a word according to ω. To compute possible *Cancel* and *Release* meanings, we first need to define the set $Det(I, i)$, that contains all possible detached commitments that were created by v_i.

Definition 4.2. Let I be an interaction and i an index such that $1 \leq i \leq len(I)$ and $I[i] = \langle a_1, a_2, v_i \rangle$. The set $Det(I, i)$ are all commitments that can have been created by v_i according to the student's ω and that are detached after i in I. Concretely, $(v_j, v, a_1, a_2) \in Det(I, i)$ if $Create(v_j, v) \in O^{\omega}(v_i)$, $I[j] = \langle a_2, a_1, v_j \rangle$ for some $i < j \leq len(I)$, and $\langle a_1, a_2, v \rangle \notin I[j : len(I)]$.

As an example, consider the interaction $I = [\langle a_1, a_2, \mathit{Offer} \rangle, \langle a_2, a_1, \mathit{Pay} \rangle, \langle a_1, a_2, \mathit{Item} \rangle]$, and suppose that $Create(\mathit{Pay}, \mathit{Receipt}) \in O^{\omega}(\mathit{Offer})$. Then $(\mathit{Pay}, \mathit{Receipt}, a_1, a_2) \in Det(I, 1)$.

If a commitment is in $Det(I, i)$, and since all interactions comply with the specification, one of two possibilities is true. Either the *Create* meaning for that commitment is wrong for v_i, or it was canceled or released after it was detached. A commitment (v_j, v, a_1, a_2) in $Det(I, i)$ can be canceled by any word uttered by a_1 after j, or released by those uttered by a_2 after j. These are the possible meanings to update.

Of course, since agents consider all possible *Create* meanings, some of them will be wrong. To take this into account, when updating the values for the possible *Cancel* and *Release* meanings in $Det(I, i)$ the student uses the confidence on the mapping between the corresponding *Create* operation and v_i. To this end, we use the values in a normalized version of ω. This function, called $\hat{\omega}$, is obtained by scaling the function ω in such a way that for each $v \in \mathcal{V}^{\omega}$, $\sum_{o \in O^{\omega}} \hat{\omega}(v, o) = 1$, and for all $o \in O^{\omega}$, $0 \leq \hat{\omega}(v, o) \leq 1$.

For each $(v_j, v, a_1, a_2) \in Det(I, i)$ and for each message $m \in I[j : len(I)]$, let $o = Release(v_j, v)$ if $m = \langle a_2, a_1, v_h \rangle$, or $o = Cancel(v_j, v)$, if $m = \langle a_1, a_2, v_h \rangle$. With ρ_2 being another reward parameter, the update is as follows:

$$\omega(v_j, o) \leftarrow \begin{cases} \omega(v_j, o) + \rho_2 \cdot \hat{\omega}(v, o) & \text{if } o \in O^{\omega}(v_h) \\ \rho_2 & \text{otherwise} \end{cases}$$

Information about possible *Cancel* and *Release* operations can also be helpful to update the *Create* meanings. There can be commitments in $Det(I, v_i)$ that cannot be canceled or released, because there are no utterances that can cancel them. This happens when the word that detaches the commitment is the last one in the interaction. For each commitment $(v_j, v, a_1, a_2) \in Det(I, i)$, if $j = len(I)$, the commitment cannot be created in the first place, and the student punishes the possible create with a parameter ρ_3:

$$\omega(v_i, Create(v_j, v)) \leftarrow \omega(v_i, Create(v_j, v)) - \rho_3$$

5 PERFORMANCE ANALYSIS

We evaluated our inference technique using two different approaches.[2] First, we analyzed how the student behaves when it uses the specifications it infers to interact with other agents. The second approach consists in measuring the correctness of the learned specifications with respect to the real ones.

Before describing the experiments, we need to establish which is the semantics that a student infers from the observations. This specification is obtained from the confidence distribution ω, and we will call it $\mathbf{means}^{\omega}$. It assigns to each word the possible meaning with maximal confidence value if there is enough evidence for it, that is, if its confidence is higher than the confidence on any other possible meaning by a difference larger than a given threshold. If there is not enough evidence, it assigns $None$:

Definition 5.1. Consider a student with confidence function ω and an evidence parameter ϵ_1. The specification $\mathbf{means}^{\omega}$ maps each $v \in \mathcal{V}\omega$ with a meaning in $O^{\omega}(v)$ as follows:

$$\mathbf{means}^{\omega}(v) = \begin{cases} \underset{o \in O^{\omega}(v)}{\text{argmax}} \;\; \omega(v, o) & \text{if for all } o' \in O^{\omega}(v) \\ & \omega(v, o) - \epsilon_1 > \omega(v, o') \\ None & \text{otherwise} \end{cases}$$

We used randomly generated specifications and interactions in the experimentation. The only restriction we imposed to the specifications was that the *Cancel* and *Release* meanings were over commitments that could be created: to include $Cancel(v, w)$ (or $Release(v, w)$) as a meaning, we required that there existed a word with meaning $Create(v, w)$. For the interactions, we gave higher probability to the ones that had active and detached commitments. We performed experiments with vocabularies of 16 words.

5.1 Experiment 1

The first experiment analyses how often a student violates a commitment according to the original specification when it uses $\mathbf{means}^{\omega}$ to interact. We simulated interactions between agents that send messages to each other alternating turns. Each agent in the interaction follows its own specification, and it always complies with it. Concretely, agents always send messages that make it possible to comply with all the commitments in a predefined number of messages. We analyze how students that use a learned $\mathbf{means}^{\omega}$ interact with other agents.

In each experiment a student goes through a training phase in which it observes a given number of interactions that follow a specification **means**. We performed the experiment for training phases of different length. The observed interactions have random lengths of between 4 and 10 messages, to avoid choosing a length arbitrarily. After the training phase, we let the student interact with an agent 50 times, with each interaction having 6 messages. The student used $\mathbf{means}^{\omega}$ to interact, while the other agent used **means**. For each interaction, we checked if the student had complied with **means**. We performed the experiment for numbers of observed interactions between 50 and 350, repeating it 50 times for each value. Table 1 shows the average proportion of successes for each number of training observations. These results show that the student learns

[2] The code is available in https://github.com/paulachocron/commitment-semantics-learning

50	100	150	200	250	300	350
55%	68.5%	76,4%	86%	88.5%	95%	96.5%

Table 1: Proportion of compliant interactions for different numbers of observed interactions

relatively fast specifications that allow it to interact with a marginal number of commitment violations.

5.2 Experiment 2

In the second experiment we compare the semantics in $\mathbf{means}^{\omega}$ with the original ones in **means**, using the standard *precision* measure.[3] Let $\mathcal{V}^c$ be all the $v \in \mathcal{V}^{\omega}$ such that $\mathbf{means}^{\omega}(v) = \mathbf{means}(v)$. The precision of $\mathbf{means}^{\omega}$ with respect to **means** is $precision(\mathbf{means}^{\omega}, \mathbf{means}) = \frac{|\mathcal{V}^c|}{|\mathcal{V}^{\omega}|}$.

For each experiment, we generated a specification **means** and let a student observe, sequentially, interactions that complied with it. These interactions had variable length of between 4 and 10 messages. For each new interaction, we measured the precision of $\mathbf{means}^{\omega}$ with respect to **means**. The experiment ended when the student found the correct meaning for all words (that means that $\text{precision}(\mathbf{means}^{\omega}, \mathbf{means}) = 1$, and all the words in $\mathcal{V}$ are known), or when it had seen a limit of 1000 interactions. We used the values for parameters that had best performance in a preliminary test: the same value for ρ_1 and ρ_2, and a much higher one for ρ_3.

Figure 1 shows the mean of the obtained precisions as a function of the number of seen interactions for different types of specifications: (a) without *Cancel* or *Release* meanings, (b) with *Release* meanings, (c) with *Cancel* or *Release* meanings. In the figure it is clear how the inclusion of cancel meanings affects significantly the convergence to the correct specification. The results show that the commitment semantics that we presented, without any external restriction, is not possible to learn completely. This is because it is impossible to distinguish when an agent canceled a commitment from when it discharged it by uttering its consequent. For example, consider the following two specifications:

(1) $\mathbf{means}(Item) = Create(Complaint, ReturnMoney)$, $\mathbf{means}(ReturnMoney) = None$
(2) $\mathbf{means}(Item) = Create(Complaint, Sorry)$, $\mathbf{means}(Sorry) = None$, $\mathbf{means}(ReturnMoney) = Cancel(Complaint, Sorry)$

Suppose the real semantics is the first one. Since agents only receive compliant interactions, $[\langle a_1, a_2, Item\rangle, \langle a_2, a_1, Complaint\rangle]$ will always be followed by $\langle a_1, a_2, ReturnMoney\rangle$, but agents have no way of deciding if *ReturnMoney* discharges the commitment or cancels it. This situation is aggravated by the fact that the student never increases the confidence for *None* meanings. Although the semantics the student learns allows to interact correctly with others, they are actually using different meanings.

6 SEMANTIC EXTENSIONS

The situation we just explained does not really affect the usability of the learning technique. For the language to make sense, there must

[3] In this case it is not necessary to consider recall, since as soon as the student has seen all the words in the vocabulary, $|\mathcal{V}^{\omega}| = |\mathcal{V}|$.

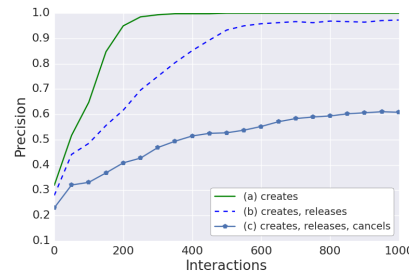

Figure 1: Convergence for different specification types

be a difference between canceling a commitment and discharging it by uttering its consequent. Otherwise, there would be no need to have two operations. Indeed, the community working on commitments has proposed diverse ways of distinguishing between both operations. These differences are not in the operational semantics of the language, but are external factors, related to how the language is used. In the rest of this section we explain some of them and we investigate how our technique can take them into account, obtaining better results in the second experiment.

6.1 Frequency

The simplest way of distinguishing between discharges and cancelations is by how frequently they occur. In [15] it is argued that canceling a commitment should be an exceptional behavior, reserved for when for some reason agents cannot discharge it. For the notion of commitments to make sense, it is necessary that agents respect them most of the times. Considering a case in which the discharges are more common than cancelations is actually enough to improve the performance of our technique.

In a situation in which *Create* operations are more frequent than *Cancel* and *Release* ones, it makes sense to spend the first interactions trying to learn the first, making the division explicit. To implement this idea, we make the student update only *Create* possible meanings for some time, and then start including *Cancel* and *Release* operations. In the experimentation, we found that updating only *Create* meanings for the first 10% of the total interactions yielded a good balance.

To test this idea, we built interactions in which agents are more likely to discharge than to cancel the commitments they made. Figure 2 shows the results for different ratios of cancelations to discharges. Considering discharges to be only twice as likely than cancelations already improves the performance notably. The modification updating only *Create* meanings for the first 10% interactions is actually better even for the 1 to 1 case, as it can be seen comparing this with Figure 1.

6.2 Observing Punishments

Another way of differentiating between discharges and cancelations takes into account the consequences for the agent that performs the action. This idea, developed for example in [15], considers a difference between what agents *should* do and what agents *can* do. They should not cancel the commitments they make, but they can do it because it may be necessary in some cases. However,

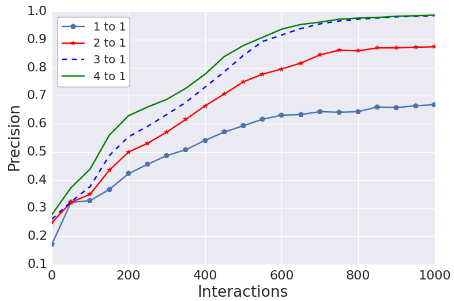

Figure 2: Frequency

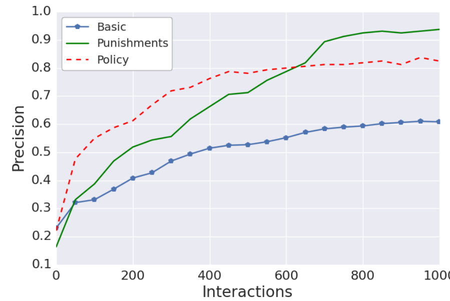

Figure 3: Punishments and Policy

agents that cancel their commitments receive a punishment for their behavior. This mechanism allows to change the punishment in a flexible way, according to the commitment that has been canceled or to other factors. For example, the punishment for canceling a commitment to give an item if the creditor pays may be low if there is an emergency, but high otherwise.

Taking into account punishments for canceling commitments can help finding the correct social meanings. We consider a simple case in which agents receive a fixed punishment if they canceled any commitment in an interaction. The student can observe this punishment, as well as which agent was punished. If the student observes an interaction and the information that agent a_1 was punished, it means that there is at least one commitment that a_1 created, a_2 detached, and a_1 canceled. Of course, the student does not know which commitment that is. However, if it observes that an agent was not punished, it knows that all its detached commitments were discharged or released, but not canceled. The student can therefore discard many possible *Cancel* meanings.

First of all, the student will only reward possible cancel meanings for those interactions when the agent was punished. For an interaction where a_i was not punished, the student subtracts a value ρ_4 from all possible *Cancel* meanings for all possible detached commitments. This is because we assume agents are punished when they cancel a commitment, not anytime they utter a word with a *Cancel* meaning. Concretely, consider indexes $1 \le i < j < h \le len(I)$, and let $I[h] = \langle a_1, a_2, v_h \rangle$ and $Cancel(u, v) \in O^{\omega}(v_h)$. If $(u, v, a_1, a_2) \in Det(I, i)$ and $I[j] = \langle a_2, a_1, v \rangle$, the student updates ω as follows:

$$\omega(v_h, Cancel(u, v)) \longleftarrow \omega(v_h, Cancel(u, v)) - \rho_4$$

Now the student is obtaining extra information about *Cancel* meanings, which can also be used to update *Create* meanings. This can be done by punishing all those *Create* meanings for which the operation is detached, and there is no possible *Cancel* or *Release* with high value later. Let $1 \le i < j \le len(I)$, and $I[i] = \langle a_1, a_2, v_i \rangle$, and suppose $Create(v_j, v) \in O^{\omega}(v_i)$ and $I[j] = \langle a_2, a_1, v_j \rangle$ but $\langle a_1, a_2, v \rangle \notin I[j : len(I)]$ (it is not discharged). Let ϵ_2 be a parameter. If for all h such that $j < h \le len(I)$, either $\hat{\omega}(v_h, Cancel(v_j, v)) \le \epsilon_2$ if $I[h] = \langle a, v_h \rangle$ or $\hat{\omega}(v_h, Release(v_j, v)) \le \epsilon_2$ if $I[h] = \langle a, v_h \rangle$ (or the meanings are not in $O^{\omega}(v_h)$), the original *Create* meaning is punished with a ρ_5 parameter:

Figure 3 shows the percentage of convergence for students that receive information about the cancel punishments. As it can be seen, the student is better at inferring commitment semantics when it can observe punishments, reaching high precision. The technique still fails sometimes because agents fail to distinguish *None* from *Cancel* or *Release* meanings, but it no longer presents the problem that we described before.

6.3 Cancelation policies

In the first papers about commitments [13], the authors proposed to have an extra specification with regulations that are external to the commitment semantics. These are higher-order constraints that describe the conditions under which different operations over commitments can be performed. The conditions to cancel commitments, in particular, are specified by *cancelation policies*. In this section we study how this extra information can help in the process of inferring meaning.

Following our original idea that only utterances are observable, we define policies as sets of constraints over words, that establish what has to be said before being allowed to cancel a commitment.

Definition 6.1. Let $O^{cl}_{\mathcal{V}\cup\{*\}}$ be all possible *Cancel* operations over the vocabulary V extended with a sign $*$. A *cancelation policy* is a relation $Pol : O^{cl}_{\mathcal{V}\cup\{*\}} \times A \times \mathcal{V} \times A$.

If, for example, $(Cancel(u, v), a_1, w, a_2) \in Pol$, agent a_2 has to say w before a_1 can cancel a commitment (u, v, a_1, a_2). If the *Cancel* operation has $*$ as antecedent or consequent, the rule is valid for cancelations of commitments with any word in that position. As an example, a policy could say that a_1 can only cancel a commitment to give an item if there was an emergency, which has to be communicated by that same agent. This would be expressed as a social policy as follows:

$$(Cancel(*, Item), a_1, Emergency, a_1) \in Pol$$

We can now define the notion of compliance of an interaction with a cancelation policy. The same is valid analogously if a_2 performs the cancelation.

Definition 6.2. Consider a vocabulary $\mathcal{V}$ and an interaction I, a specification **means**, and a cancelation policy Pol over $\mathcal{V} \cup \{*\}$. An index $1 \le i \le len(I)$ such that $I(i) = \langle a_1, v_i \rangle$ and $\textbf{means}(v_i) = Cancel(u, v)$ complies with Pol if, for $w \in \mathcal{V}, a \in A$,

$$(Cancel(u, v), a_1, w, a) \in Pol \implies \langle a, w \rangle \in I[1 : i]$$

and the same holds for $Cancel(*, v)$, $Cancel(u, *)$, and $Cancel(*, *)$. The interaction I complies with Pol if all its indexes comply with it.

Cancelation policies are common cultural knowledge that is universally shared, also by the student. Note that this implies also that the student needs to know (part of) the vocabulary a priori. These rules can be helpful to decide the meaning of words, since they can rule out impossible meanings. We assume that the interactions seen by the student always comply with the cancel policy, and that the student knows this.

The process to take cancelation policies into account is simple. When a student with a cancel policy *Pol* observes an interaction I, it checks each word for possible cancel meanings, and punishes those that would not comply with *Pol*. For each $1 \leq i \leq len(I)$ such that $I[i] = \langle a_1, a_2, v_i \rangle$, if $Cancel(u, v) \in O^{\omega}(v_i)$ and assigning that meaning would make the index i non-compliant, punish the mapping with a parameter ρ_6:

$$\omega(v_i, Cancel(u, v)) \leftarrow \omega(v_i, Cancel(u, v)) - \rho_6$$

The effect of cancelation policies on the inference process depends on which kind of *Cancel* operations are regulated. If the policy has rules for *Cancel* operations with $*$ for both the antecedent and the consequent, the effect is similar than in the case with punishments, since the student knows that, in some situations, the agents cannot have canceled their commitments. Figure 3 shows the results for a student that shares a policy with the community. We used policies that had one rule for each *Create* meaning in the original specification, with $*$ as consequent. These rules are the most informative ones, since in the case without policies the student confuses canceling operations with the same antecedent. As we can see, agents reach higher F-score sooner than in the case with punishments. This may be because every interaction is affected by the policy, while interactions where agents cancel commitments give no extra information in the case of cancelation policies. However, they obtain a lower value in the end, which may be because the restrictions imply that some words are uttered less often than others, and the student does not have enough information to learn their meaning.

7 RELATED WORK

As we already mentioned, there exist multiple efforts to tackle different aspects of the interoperability problem for commitments. Chopra and Singh presented the concept of *constitutive interoperability* [5], that defines two agents as aligned if they share the same commitments. They also provided techniques to align commitments, in the case, for example, of asynchronous interactions in which a message can arrive late [6]. In addition, Gunay et al. tackled the problem of generating protocols dynamically, adapting to an open situation [10].

The problem of inferring commitment semantics is closely related to the more investigated problem of *norm inference*, that studies how an agent can learn what actions are allowed and which ones are not in a community [7]. This similarity is unsurprising given the relation between the notions of *obligation* norms and commitments, discussed in [15]. In the same paper, however, the author points at critical differences. Mainly, agents cannot operate with obligations in the same way they can cancel or release commitments. The problem of inferring rules lacks therefore the complexity that, as we have seen, arises from these operations. Also related is the problem of Process Mining [18], that provides techniques to analyze business processes from the information that is stored in event logs. In this case, procedures are specified with automatas or rules. Our work can be used in a similar way when the specification has a component that can be expressed with commitments.

The problem of learning word meanings from scratch has been tackled several times, but almost never for social semantics. Instead, a large amount of work considers how agents can learn from observing a common environment [17], or by interacting with each other [3, 4]. The latter approaches use updating mechanisms that are similar to the ones we present here. In that case, confidence distributions are used by agents who have different vocabularies but similar task specifications to obtain a translation that allows them to interact.

8 CONCLUSIONS AND FUTURE WORK

In this paper we presented a method to infer the commitment semantics used by a community, that can be used by a student who observes compliant interactions between other agents. Importantly, our techniques use no external resource, and make no assumption about the student, except that it knows the operational semantics of the specification language. The actual language used in the specification, however, can be different. This provides a very flexible method that can be used in open situations.

The semantics of commitments that we proposed is not possible to learn completely by observing interactions, since it does not distinguish between the acts of canceling and of discharging a commitment. We explored different extensions to the semantics that help to differentiate between these two actions, and showed how they are useful to infer the correct semantics. Already observing agents that discharge their commitments more frequently than they cancel them results in a better inference process.

In this paper we only performed experiments with small vocabularies, that are illustrative, but not realistic use cases. Exploratory experiments with larger vocabularies that show that, while the techniques seem to scale up in terms of finding good alignments, they become very slow for large vocabularies. This is because there are many possible meanings for each words. This can be taken into account using a pruning technique on the possible mappings, or considering pragmatic restrictions, such as not allowing agents to cancel commitments that have not been created.

Until now we have only considered students that observe compliant interactions, in which agents do not violate any of their commitments. However, this is not necessarily how real interactions are. Violations can be considered meaningfully in our technique if the student can identify them as such, for example observing a punishment. Extending our method to this case is also future work.

Finally, it would be interesting to combine our approach with other kind of information that can also help the learning process, such as linguistic structure. To this end, it is first necessary to develop a meaningful integration between commitment specifications and more complex linguistic structures.

Acknowledgments. This research has been funded by the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 607062 /ESSENCE: Evolution of Shared Semantics in Computational Environments/, and by CSIC's Proyectos Intramurales Especiales DIVERSIS (no. 201750E064) and SMA (no. 201550E040).

REFERENCES

[1] Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth. 2003. The chatty web: emergent semantics through gossiping. In *Proceedings of the 12th international conference on World Wide Web*. ACM, 197–206.

[2] Daniel Brand and Pitro Zafiropulo. 1983. On communicating finite-state machines. *Journal of the ACM (JACM)* 30, 2 (1983), 323–342.

[3] Paula Chocron and Marco Schorlemmer. 2016. Attuning Ontology Alignments to Semantically Heterogeneous Multi-Agent Interactions. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. 871–879. https://doi.org/10.3233/978-1-61499-672-9-871

[4] Paula Chocron and Marco Schorlemmer. 2017. Vocabulary Alignment in Openly Specified Interactions. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1064–1072. http://dl.acm.org/citation.cfm?id=3091210.3091275

[5] Amit K. Chopra and Munindar P. Singh. 2008. Constitutive Interoperability. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS '08)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 797–804. http://dl.acm.org/citation.cfm?id=1402298.1402335

[6] Amit K. Chopra and Munindar P. Singh. 2015. Generalized Commitment Alignment. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 453–461. http://dl.acm.org/citation.cfm?id=2772879.2772938

[7] Stephen Cranefield, Felipe Meneguzzi, Nir Oren, and Bastin Tony Roy Savarimuthu. 2016. A Bayesian Approach to Norm Identification. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. 622–629. https://doi.org/10.3233/978-1-61499-672-9-622

[8] Mark dInverno, Michael Luck, Pablo Noriega, Juan A. Rodriguez-Aguilar, and Carles Sierra. 2012. Communicating Open Systems. *Artif. Intell.* 186 (July 2012), 38–94. https://doi.org/10.1016/j.artint.2012.03.004

[9] Ulrich Endriss, Nicolas Maudet, Fariba Sadri, and Francesca Toni. 2003. Logic-based agent communication protocols. In *Workshop on agent communication languages*. Springer, 91–107.

[10] Akin Günay, Michael Winikoff, and Pinar Yolum. 2015. Dynamically Generated Commitment Protocols in Open Systems. *Autonomous Agents and Multi-Agent Systems* 29, 2 (March 2015), 192–229. https://doi.org/10.1007/s10458-014-9251-7

[11] Andrew Koster, Jordi Madrenas, Nardine Osman, Marco Schorlemmer, Jordi Sabater-Mir, Carles Sierra, Angela Fabregues, Dave de Jonge, Josep Puyol-Gruart, and Pere Garcia-Calvés. 2013. u-Help: Supporting Helpful Communities with Information Technology. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems (AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1109–1110. http://dl.acm.org/citation.cfm?id=2484920.2485095

[12] Elisa Marengo, Matteo Baldoni, Cristina Baroglio, Amit K. Chopra, Viviana Patti, and Munindar P. Singh. 2011. Commitments with Regulations: Reasoning About Safety and Control in REGULA. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2 (AAMAS '11)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 467–474. http://dl.acm.org/citation.cfm?id=2031678.2031684

[13] Munindar P. Singh. 1999. An ontology for commitments in multiagent systems:. *Artificial Intelligence and Law* 7, 1 (01 Mar 1999), 97–113. https://doi.org/10.1023/A:1008319631231

[14] Munindar P. Singh. 2000. A Social Semantics for Agent Communication Languages. In *Issues in Agent Communication*. 31–45. https://doi.org/10.1007/10722777_3

[15] M. P. Singh. 2012. Commitments in multiagent systems: Some history, some confusions, some controversies, some prospects. In *The goals of cognition: Essays in honor of Cristiano Castelfranchi*, Tummolini L. Falcone R. Miceli M. Paglieri, F. (Ed.). 1–29.

[16] Jeffrey Mark Siskind. 1996. A Computational Study of Cross-Situational Techniques for Learning Word-to-Meaning Mappings. *Cognition* 61, 1-2 (1996), 39–91.

[17] Luc Steels. 1998. The Origins of Ontologies and Communication Conventions in Multi-Agent Systems. *Autonomous Agents and MultiAgent Systems* 1 (1998), 169–194. https://doi.org/10.1023/A:1010002801935

[18] Wil M. P. van der Aalst. 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes* (1st ed.). Springer Publishing Company, Incorporated.

[19] Mahadevan Venkatraman and Munindar P. Singh. 1999. Verifying Compliance with Commitment Protocols. *Autonomous Agents and Multi-Agent Systems* 2, 3 (01 Sep 1999), 217–236. https://doi.org/10.1023/A:1010056221226