Università degli Studi di Bologna
DEIS

Diagnosing commitments: Delegation revisited

Özgür Kafalı Paolo Torroni

April 21, 2011

DEIS Technical Report no. DEIS-LIA-001-11 *LIA Series no. 99*

Diagnosing commitments: Delegation revisited

Özgür Kafalı [1] Paolo Torroni [2]

[1] *Department of Computer Engineering - Boğaziçi University*
34342, Bebek, İstanbul - Turkey
ozgurkafali@gmail.com
[2] *DEIS - University of Bologna*
V.le Risorgimento, 2, 40136, Bologna - Italy
paolo.torroni@unibo.it

April 21, 2011

Abstract. The success of contract-based multiagent systems relies on agents complying with their commitments. When something goes wrong, it is important to understand what are the commitments' mutual relations as well as their individual states. Accordingly, we explore how commitments are related through the three-agent commitment delegation operation. We then propose exception monitoring based on such relations, and demonstrate it via a case study.*

Keywords: *Agent-based and Multi-agent Systems, Argumentation, Negotiation, Contract-Based Systems, Diagnosis*

*This paper extends the AAMAS '11 poster paper [8].

Contents

1 Introduction

A social commitment describes a contract between two agents: the debtor commits to satisfy a property for the creditor [9]. In a contract-based multiagent system, several such commitments are in effect, e.g., the merchant is committed to deliver the goods when the customer pays, the bank is committed to confirm the customer's payment in three days. The former is represented by a conditional commitment CC(merchant, customer, paid, delivered), while the latter is represented by a base-level temporal commitment C(bank, customer, paid(3)), e.g., with a deadline. Often, agents delegate their commitments to others for several reasons, e.g., they are not capable of satisfying the properties. C(courier, merchant, delivered) is a delegation of CC(merchant, customer, paid, delivered) in which the merchant delegates the task of delivery to the courier.

Usually, commitments formed between different agents are connected to each other; either explicitly (by delegation), or implicitly (other dependencies). The delegation operation is important in the sense that it extends the set of agents that a commitment is involved with. The merchant-courier example demonstrates a typical case of delegation, where the commitment of delivery between the merchant and customer agents is extended with the courier agent. The customer agent may not be aware of this extension until the delivery is completed, or something goes wrong (e.g., the deadline passes). In the case of a problem, this connection should be revealed so that if the problem is related to the courier, it can be identified. However, not all connections are explicitly identified as in the delegation case. The bank-customer example is a typical case of implicit causal dependency; the bank's confirmation of the customer's payment affects the merchant's commitment of delivery towards the customer. Again, if something goes wrong, the commitments should be traced in a chain to identify the problem.

We say that an exception has occurred regarding a property of the system if there is something wrong in the chain of commitments that are formed to satisfy that property. Possible causes of such exceptions are:

- *Violation:* One of the commitments related to the subject property is violated by its debtor, e.g., the bank does not confirm the customer's payment in time. Often, an exception is considered identical to a commitment violation.

- *Improper delegation:* One of the commitments related to the subject property is delegated without respecting its previous deadline, e.g., the merchant delays the courier's delivery by handing over the goods late.

- *Misalignment:* Two or more commitments related to the subject property are not aligned with each other, possibly due to different observations of the participating agents, e.g., the bank confirms the customer's payment but fails to notify the merchant on time.

When there are many commitments in the system at hand, in order to identify an exception we need effective ways to explore the space of commitments. In particular, we need to identify links between commitments and exclude from our search the irrelevant instances. The process of tracking down individual commitment states is called commitment monitoring [1, 10]. We extend monitoring to enable run-time tracking of exceptions via the links between agents' commitments. To this end, we propose similarity relations to connect commitments based on their components: the agents, the properties, and the temporal aspects. The properties of commitments have been studied before in [3]. Moreover, temporal constraints are used to compare commitments in [7]. However, both approaches mainly focus on two-agent interactions, or compare commitments one by one. In [4], delegation is also taken into account from the perspective of commitment alignment, but no temporal aspects are considered. In [1], commitments are tracked and reasoned upon centrally. Those work should be able to capture exceptions caused by improper delegation or misalignment. Still, they fail to reveal implicit commitment dependencies as we propose to identify here. Note that a relatively insignificant commitment violation may be the cause of an exception.

Accordingly, we propose a monitoring framework where the relations between agents' commitments are extracted to identify problems related to a commitment violation. The framework identifies all improper delegations that have occurred in the system.

The rest of the paper is structured as follows. Section 2 reviews commitments and introduces our formal model. Section 3, 4 and 5 describe commitment delegation and the similarity relations based on delegation. Section 6 describes how monitoring is performed, proposing an implementation. Section 7 demonstrates the case study. Finally, Section 8 concludes the paper with further discussion.

2 Formal Model

A commitment [9] is formed between two agents; the debtor commits to the creditor for satisfying a property. There are two types of commitments:

- *Conditional commitment:* CC(X, Y, Q, P(T)) represents a conditional commitment where X is the debtor, Y is the creditor, Q is the antecedent, and P(T) is the consequent. Here, T is the *limit* of the consequent. This is an existential temporal constraint [1, 10], i.e., the consequent has to be satisfied within T time units, as of the moment the antecedent is satisfied.

- *Base-level commitment:* When the antecedent Q is satisfied, a base-level commitment C(X, Y, P(T)) is formed. Here, T is defined as the *deadline* of the consequent, which is slightly different from the limit of a conditional commitment. The temporal constraint T of a conditional commitment represents the amount of time allowed to satisfy P (relative deadline), while T in the base-level commitment represents an absolute deadline. For example, the commitment CC(merchant, customer, paid, delivered(7))

tells that the delivery should take place at most seven time units after the payment. If the payment is done at time 3, then the base-level commitment C(merchant, customer, delivered(10)) tells that the merchant has to deliver latest at time 10.

The commitment properties (i.e., antecedent and consequent) are described by conjunctions of atomic propositions together with temporal constraints to express limits or deadlines. Currently, we do not support negation or disjunction of properties. A commitment is a live object and changes state through its life-cycle [12]. We use the following four states for commitments: (1) *conditional*, when Q is not yet satisfied, (2) *active*, when Q is satisfied and P is not yet satisfied, (3) *fulfilled*, when P is satisfied within T, and (4) *violated* when P is not satisfied within T.

We are now ready to define the monitoring framework.

Definition 1 Monitoring framework $\mathcal{F}$ is a five-tuple $\langle \mathcal{P}, \mathcal{R}, \mathcal{A}, \mathcal{T}, \mathcal{M} \rangle$, where

- $\mathcal{P}$ is a set of conditional commitments, representing a contract or a protocol [12, 2],
- $\mathcal{R}$ is a set of roles, each consisting of a subset of $\mathcal{P}$'s commitments and a set of actions,
- $\mathcal{A}$ is a set of agents, each enacting a role in $\mathcal{R}$,
- $\mathcal{T}$ is a trace of events, consisting of a set of actions performed at specific time points, and
- $\mathcal{M}$ is the monitoring process.

∎

Commitments in $\mathcal{P}$ are abstract entities, i.e., *templates* which include roles from $\mathcal{R}$ in place of agents. When the agents in $\mathcal{A}$ are bound to the roles in the protocol [5], the commitments become real. The trace of events $\mathcal{T}$ describes a specific protocol execution, by which commitments change state accordingly. The monitoring process $\mathcal{M}$ consistently checks for improper delegations during the protocol's execution. Similarly to diagnosis, which looks for assumptions over executions of activities that classifies these executions either as correct (an activity behaves as intended) or faulty (an activity does not behave as intended) [6], monitoring seeks to detect faulty activity executions. It is important that monitoring is carried out at runtime, in reaction to events that bring about properties characterizing a faulty state. We propose to identify exceptions through the monitoring of agents' commitments. We describe $\mathcal{M}$ in more detail in Section 6.

3 Delegation

Previous work has looked at commitments and their relations from different angles. Chopra and Singh [4, 3] compare commitments via a *strength* relation using the commitments' properties, Kafalı *et al.* [7] focus on the temporal aspects of commitments and provide similarity relations based on the commitments' deadlines. In particular, we combine both approaches, and propose similarity relations based on the three-agent commitment delegation operation. First, we describe a delegation.

Definition 2 A *delegation* of a commitment *CC (X, Y, Q, P)*, called *primary*, is a new commitment where either X or Y plays the role of the creditor or debtor, and a new agent Z is responsible for bringing about the antecedent Q or the consequent P. ▮

Six types of delegation are particularly meaningful. Only some of them have been considered previously in the literature. Let us define and illustrate them one by one, considering *CC(merchant, customer, paid, delivered)* as our *primary*, like in Figure 1.

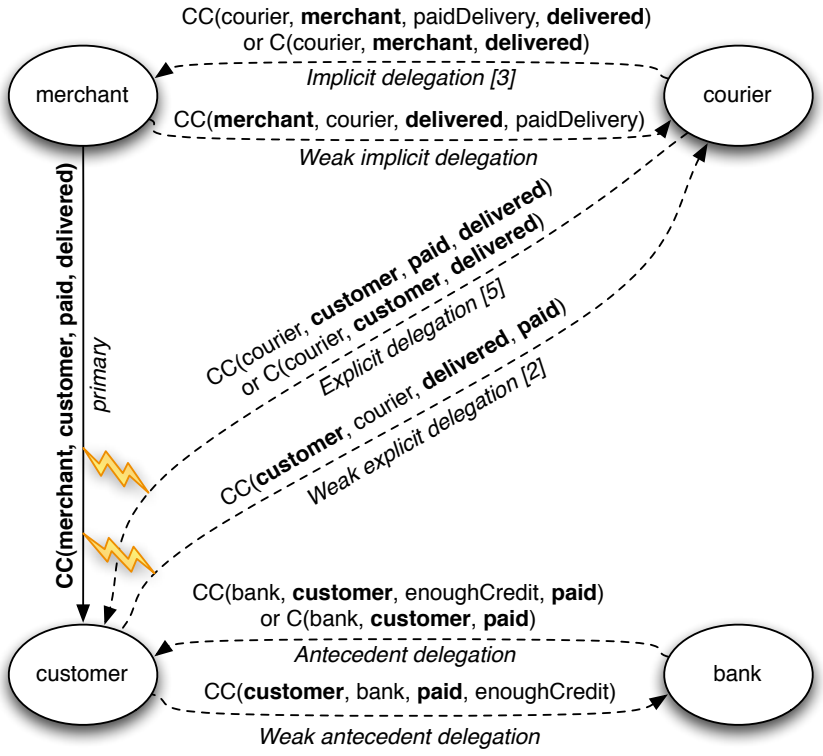

Figure 1. *Sample Delegations*

Definition 3 (Explicit delegation) The primary is canceled and a new commitment *CC (Z, Y, Q, P)* is created. That is, a new debtor is committed to the same creditor. ▮

This delegation operation was proposed by Yolum and Singh [12]. A possible explicit delegation of the primary is *CC(courier, customer, paid, delivered)*. The new debtor *courier* replace the old debtor *merchant*, and the primary is canceled. Note that the antecedent (*paid*) is unchanged, but that may not necessarily be the case in an extended version of explicit delegation (see below).

Definition 4 (Weak explicit delegation) The primary is canceled and a new commitment *CC (Y, Z, P, Q)* is created. That is, the creditor *Y* of the primary is now the debtor of the new commitment, and *Y* wishes to achieve *P* via a new creditor *Z*. This is a weak delegation to achieve *P* since there is no obligation for *Z* to satisfy *P* unless *Z* needs *Q* satisfied. ▮

The concept of weak delegation is inspired by Chopra *et al.* 's work [2]. A possible weak explicit delegation of the primary is *CC(customer, courier, delivered, paid)*. Note that the roles of creditor and debtor are reversed, and accordingly also antecedent and consequent are reversed.

Definition 5 (Implicit delegation) While the primary is still active, a new commitment *CC (Z, X, R, P)* is created. That is, the debtor *X* of the primary is now the creditor of a new commitment for the same consequent *P*. ▮

This type of delegation chain (e.g., two dependent commitments) was proposed by Kafalı *et al.* [7]. A possible implicit delegation of the primary is *CC(courier, merchant, paidDelivery, delivered)*. Note that the creditor is the *merchant*, which is the primary's debtor. For that reason, to maintain a commitment to the initial creditor (the *customer*), the primary is not canceled, but it remains.

Definition 6 (Weak implicit delegation) While the primary is still active, a new commitment *CC (X, Z, P, R)* is created. That is, the debtor *X* of the primary also becomes the debtor of a new commitment where the antecedent *P* is the primary's consequent. ▮

A possible weak implicit delegation of the primary is *CC(merchant, courier, delivered, paidDelivery)*.

Definition 7 (Antecedent delegation) While the primary is still active, a new commitment *CC (Z, Y, R, Q)* is created. That is, the creditor *Y* of the primary also becomes the creditor of a new commitment for the antecedent *Q* of the primary. ▮

A possible antecedent delegation of the primary is *CC(bank, customer, enoughCredit, paid)*. Note that the initial consequent (*delivered*) does not appear in the antecedent delegation. For that reason, to maintain a commitment about the initial consequent, the primary is not canceled, but it remains.

Definition 8 (Weak antecedent delegation) While the primary is still active, a new commitment *CC (Y, Z, Q, R)* is created. That is, the creditor *Y* of the primary is now the debtor of a new commitment which has the same antecedent *Q* as the primary. ▮

A possible weak antecedent delegation of the primary is *CC(customer, bank, paid, enoughCredit)*.

Most of the above definitions can be extended to base-level commitments. In addition, (weak) explicit delegation can be extended to have an antecedent R different from Q. Also note that a special case of (weak) implicit delegation is where R equals Q. Figure 1 summarizes the examples of commitment delegation given above.

We gave an exhaustive account of how a commitment can be *rationally* delegated, i.e., by preserving the responsibilities of roles in relation with the primary's properties. To see this, let us enumerate all the rational delegation possibilities for a commitment, and then show that they are covered by Definitions 3-8.

Let us first consider rational delegations that include the consequent *P* of a primary *CC(X, Y, Q, P)*. The secondary will have *P* either as the consequent or the antecedent. Moreover, by Definition 2, a new agent *Z* should replace either *X* or *Y*.

Case 1: *P* is the secondary's consequent. In the primary, the debtor is responsible for *P*. Therefore we need a new debtor *Z* responsible for *P*. There are three alternatives: (1.1) *CC(Z, Y, R, P)*, which is an (extended) explicit delegation; (1.2) *CC(Z, X, R, P)*, which is an implicit delegation; or (1.3) *CC(Z, W, R, P)*, with *W* also different from *X* and *Y*. However, the latter case is just a different commitment about *P*, but it cannot be considered a delegation, since there is no common agent between primary and secondary.

Case 2: *P* is the secondary's antecedent. In that case, *Z* will be new creditor expecting *P*. There are again three alternatives: (2.1) *CC(Y, Z, P, R)*, which is a weak explicit delegation, (2.2) *CC(X, Z, P, R)*, which is a weak implicit delegation; and (2.3) *CC(W, Z, P, R)*, which cannot be considered a delegation for the same reasons as in Case 1.

Let us now consider rational delegations that include the antecedent *Q*. There are again two possibilities:

Case 3: *Q* is the secondary's consequent. Rationally, the antecedent should be different from *Q* and *P*, and the creditor should be the same *Y*, since *Y* is the agent who expects *Q* satisfied. The only option for a delegation is to have a new debtor *Z*, other than *X* or *Y*. That is the definition of antecedent delegation, *CC(Z, Y, R, Q)*.

Case 4: *Q* is the antecedent. Rationally, the secondary's debtor should be *Y*, and the secondary should have a new creditor *Z*, other than *X* or *Y*. This defines a weak antecedent delegation, *CC(Y, Z, Q, R)*.

4 Similarity

We say that a commitment is *delegation-similar* to another commitment if one is a delegation of the other according to Definitions 3-8. We showed that our account of commitment

delegation is exhaustive (if we restrict ourselves to rational delegations). Now, we shift the focus to commitments that are similar *via* other commitments. We will therefore need to define ternary similarity relations, which include two commitments similar to each other (we call them "secondary"), and a "primary" commitment which connects them. For now, we omit the temporal constraints from the commitments.

Definition 9 Commitment $CC_1(X_1, Y_1, Q_1, P_1)$ is consequent-delegation similar to commitment $CC_2(X_2, Y_2, Q_2, P_2)$ via commitment $CC_3(X_3, Y_3, Q_3, P_3)$ iff

1. $P_3 \models P_1 \wedge P_2$ (conjunction), and
2. $Y_1 = Y_2 = X_3$ (delegation).

We call CC_3 the *primary*, and CC_1, CC_2 *secondary*. ▮

In Definition 9, the debtor X_3 of a primary about a complex commitment (in which the consequent P_3 is a conjunction of several parts, $P_3 = P_1 \wedge P_2 \wedge \ldots$) negotiates two implicit delegations of two such parts, with two new agents. As a result, we have two delegations, and therefore two secondaries, connected by the same primary. Note that the conditional commitments in the definition can also be base-level commitments since the antecedents are not part of the similarity relation.

Consider for instance the commitments in $\mathcal{C}_{consequent}$:

$$\mathcal{C}_{consequent} = \begin{cases} C_3(\text{merchant, customer, delivered} \wedge \text{invoiced}) \\ C_1(\text{courier, merchant, delivered}) \\ C_2(\text{accountant, merchant, invoiced}) \end{cases}$$

According to C_3, the merchant is committed to deliver the item and provide the invoice. Now, assume that the merchant delegates the delivery of the item to the courier, and the preparation of the invoice to his accountant. Thus, we have the commitments (C_1 and C_2). Accordingly, C_1 and C_2 are consequent-delegation similar via C_3. C_3 is primary, and C_1, C_2 are secondary. The converse case (weak delegation) is also possible:

Definition 10 Commitment $CC_1(X_1, Y_1, Q_1, P_1)$ is weak consequent-delegation similar to commitment $CC_2(X_2, Y_2, Q_2, P_2)$ via commitment $CC_3(X_3, Y_3, Q_3, P_3)$ iff

1. $P_3 \models Q_1 \wedge Q_2$ (conjunction), and
2. $X_1 = X_2 = X_3$ (delegation).

▮

Let us now focus on the antecedent. Consider the following example:

$$\mathcal{C}_{antecedent} = \begin{cases} CC_3(\text{merchant, customer, paid} \wedge \text{confirmed, delivered}) \\ C_1(\text{accountant, customer, paid}) \\ C_2(\text{bank, customer, confirmed}) \end{cases}$$

According to CC_3, once the customer pays and her card is confirmed, the merchant will deliver the item. Now, the customer delegates the payment to her accountant, and the confirmation to the bank. This brings us two new commitments, C_1 and C_2, respectively. The following definitions capture this type of similarity between C_1 and C_2.

Definition 11 Commitment $CC_1(X_1, Y_1, Q_1, P_1)$ is antecedent-delegation similar to commitment $CC_2(X_2, Y_2, Q_2, P_2)$ via commitment $CC_3(X_3, Y_3, Q_3, P_3)$ iff

1. $Q_3 \models P_1 \wedge P_2$ (conjunction), and
2. $Y_1 = Y_2 = Y_3$ (delegation).

As usual, CC_3 is the *primary*, and CC_1, CC_2 *secondary*. ▮

Definition 11 revises Definition 9 for conjunction of antecedents. This also represents a causal relation between the commitments, since CC_3 cannot be discharged until CC_1 and CC_2 are both fulfilled. The secondary commitments in the definition can also be base-level commitments. However, CC_3 cannot be a base-level commitment anymore since Q_3 is part of the relation. Antecedent delegation is not actively used in monitoring, but acts as a connective in a chain of delegations. In the previus example, C_1 and C_2 are antecedent-delegation similar via CC_3. A weaker version is possible:

Definition 12 Commitment $CC_1(X_1, Y_1, Q_1, P_1)$ is weak antecedent-delegation similar to commitment $CC_2(X_2, Y_2, Q_2, P_2)$ via commitment $CC_3(X_3, Y_3, Q_3, P_3)$ iff

1. $Q_3 \models Q_1 \wedge Q_2$ (conjunction), and
2. $X_1 = X_2 = Y_3$ (delegation).

▮

Let us finally consider the following situation:

$$\mathcal{C}_{causal} = \begin{cases} CC_1(\text{bank, client, requested, delivered}) \\ CC_3(\text{courier, bank, printed, delivered}) \\ C_2(\text{office, bank, printed}) \end{cases}$$

According to CC_1, once the client requests a credit card, the bank will deliver the card. Now, the bank delegates the delivery to the courier via CC_3. However, in order to deliver,

the courier needs the card printed. Thus, the bank makes another delegation with the office via C_2. There is a causal relation between commitments here: in order to satisfy CC_1's secondary CC_3, a new commitment C_2 is in place. In particular, CC_3 is the link between two otherwise seemingly unrelated commitments.

We will say that CC_1 and C_2 are causal-delegation similar via CC_3. Put formally:

Definition 13 Commitment $CC_1(X_1, Y_1, Q_1, P_1)$ is causal-delegation similar to commitment $CC_2(X_2, Y_2, Q_2, P_2)$ via commitment $CC_3(X_3, Y_3, Q_3, P_3)$ iff

1. $P_1 = P_3$ and $X_1 = Y_3$ (implicit-delegation), and
2. $P_2 = Q_3$ and $Y_2 = Y_3$ (antecedent-delegation).

Here, we call CC_1 *outcome*, CC_2 *cause*, and CC_3 *connective*. ▮

Definition 13 connects two commitments through two delegations; one consequent (implicit) and one antecedent delegation. Here, CC_1 and CC_2 can also be base-level commitments since Q_1 and Q_2 are not part of the relation. However, CC_3 cannot be a base-level commitment since Q_3 is part of the relation. We do not consider the weak delegation case for causal delegation. Because, if the *consequent delegation* is weak, then the causal structure between the *connective* and *cause* no longer exists.

The similarity relations described in Definitions 9-13 allow us to trace commitments in a chain-like structure, e.g., nested delegations. They are also exhaustive, in the sense that they describe all possible, rational cases of commitment delegation. Let us informally see why.

The primary commitment in (weak) explicit delegation no longer exists after delegation, thus we do not need to consider it in the similarity relations. That is, there is no longer a primary commitment to compare with. Implicit delegation is covered in Definition 13, between commitments *outcome* and *connective*. In addition, Definition 9 covers implicit delegation where conjunction of consequents is considered. Weak implicit delegation with conjunction of consequents is covered in Definition 10, the single consequents case is trivial (i.e., each consequent is identical).

Antecedent delegation is covered in Definition 13, between commitments *connective* and *cause*. In addition, Definition 11 covers antecedent delegation where conjunction of antecedents is considered. Weak antecedent delegation with conjunction of antecedents is covered in Definition 12. Again, the single antecedents case is trivial.

5 Limits And Deadlines

We will now extend the ternary relations by taking into account the temporal constraints. Since we are interested in exceptional situations and in the possible reasons behind them, we will identify and define cases of delegation in which the deadline of the primary is not

properly propagated onto the secondary. In particular, we will call *improper* a delegation that exceeds the deadline of the primary commitment.

Definition 14 An improper consequent delegation between a primary and a secondary commitment (as by Definition 9) occurs if either of the following holds:

1. All conditional: If the limit of a secondary is greater than the limit of the primary.
2. All base-level: If the deadline of a secondary is greater than the deadline of the primary.
3. Only primary conditional: If the deadline of a secondary is greater than the limit of the primary added to the current time point.
4. Only primary base-level: If the limit of a secondary added to the current time point is greater than the deadline of the primary.

∎

Consider for instance the commitments in $\mathcal{C}_{improper-consequent}$:

$$\mathcal{C}_{improper-consequent} = \begin{cases} C_3(\text{merchant, customer, delivered}(12) \wedge \text{invoiced}(12)) \\ C_1(\text{courier, merchant, delivered}(12)) \\ C_2(\text{accountant, merchant, invoiced}(14)) \end{cases}$$

These are the same commitments in $\mathcal{C}_{consequent}$, modified with temporal constraints. Assume that the current time is 8. Now, this is an improper delegation. Because, the deadline of C_2 is greater than that of C_3. Note that the occurrence of an exception, although likely, is not inevitable since the accountant may still satisfy $invoiced$ at time 12.

Note that we cannot have an improper antecedent delegation, since we do not consider time limits for the antecedent of a commitment in this work.

Definition 15 An improper causal delegation between a cause and an outcome commitment (as by Definition 13) occurs if either of the following holds:

1. All conditional: If the limit of the cause added to the limit of the connective is greater than the limit of the outcome.
2. Only connective conditional: If the deadline of the cause added to the limit of the connective is greater than the deadline of the outcome.

∎

Consider the commitments in $\mathcal{C}_{improper-causal}$:

$$\mathcal{C}_{improper-causal} = \begin{cases} CC_1(\text{bank, client, requested, delivered(7)}) \\ CC_3(\text{courier, bank, printed, delivered(5)}) \\ CC_2(\text{office, bank, confirmed, printed(3)}) \end{cases}$$

These are the same commitments as in $\mathcal{C}_{causal}$, modified with temporal constraints. Obviously, there is a problem with this choice of conditional commitments. Because, in order for the card to be delivered, it has to be printed first, and the time requirements for those two processes exceed the time limit that the bank has towards the client. The bank should have gotten into other commitments that would have lead the fulfillment of its primary commitment towards the client [2]. However, note that CC_2 and CC_3 may still fulfill CC_1 since the debtors of those commitments may satisfy the consequents before the deadlines.

Definition 16 An improper delegation is an improper consequent delegation or an improper causal delegation, or a combination of them. ▮

Combinations of delegations can occur in a chain-like structure. In such cases, the deadlines or limits should be propagated for correct monitoring. Consider for instance the commitments in $\mathcal{C}_{joint}$:

$$\mathcal{C}_{joint} = \begin{cases} C_1(\text{bank, client, delivered(10)}) \\ CC_2(\text{courier, bank, printed} \wedge \text{tested, delivered(5)}) \\ C_3(\text{office, bank, printed(5)} \wedge \text{tested(5)}) \\ C_4(\text{operator, office, printed(5)}) \\ C_5(\text{tester, office, tested(7)}) \end{cases}$$

There is no problem when we consider C_1, CC_2 and C_3 only; the deadlines are consistent. However, the deadline of C_5 is greater than the deadline of C_3 which creates a problem for the C_3, C_4 and C_5 consequent-delegation group. This should be propagated up to C_1. That is, the expected deadline of C_3 is now extended, which further extends the expected deadline of C_1.

6 Monitoring

Given a monitoring framework $\mathcal{F} = \langle \mathcal{P}, \mathcal{R}, \mathcal{A}, \mathcal{T}, \mathcal{M} \rangle$, $\mathcal{M}$ identifies all the improper delegations that occurred up to the current time point T. It makes use of the current states of the commitments $\mathcal{C}_T$ in the system to produce $\mathcal{M}_T$; the set of improper delegations. Now, let us formally describe $\mathcal{M}$.

Definition 17 Given a monitoring framework $\mathcal{F}$, the current time point T, and a set of commitments $\mathcal{C}_T = \{C_1, ..., C_i, ..., C_n\}$ that describe the current system at T, the monitoring process $\mathcal{M}$ produces the monitoring outcome $\mathcal{M}_T = \{(C_i, C_j) \mid C_i, C_j \in \mathcal{C}_T$ and C_i is an improper delegation of $C_j\}$ that contains the improper delegations among C_T. ▮

The purpose of monitoring is to identify the faults (e.g., improper delegations) among agents' commitments. Once we have identified the improper delegations, we can signal an exception based on $\mathcal{M}_T$. The following two rules describe when an exception occurs:

$$\frac{monitor(C_m) \wedge \exists C_i : (C_i, C_m) \in \mathcal{M}_T}{exception(C_i, C_m)}$$

The first rule states that if we are interested in the monitoring of commitment C_m, and there is a commitment C_i that is an improper delegation of C_m, then there is an exception.

$$\frac{monitor(C_m) \wedge \exists C_j : delegation(C_j, C_m) \wedge \exists C_i : (C_i, C_j) \in \mathcal{M}_T}{exception(C_i, C_m)}$$

The second rule extends the first rule by taking into account nested delegations. In particular, if a delegation (not necessarily improper) of the subject commitment C_m has an improper delegation, then there is an exception. Note that the level of nested delegations is indefinite. That is, C_m may have been delegated several times (e.g., C_m to C_x and C_x to C_y), and then the occurrence of an improper delegation triggers the exception. Note that we do not provide an algorithm for exception monitoring in this paper. Rather, we present the rules that describe how a certain case will be considered an exception in a declarative way. We plan to build a distributed procedure where agents will identify exceptions via an exchange of their local knowledge.

We implemented a proof-of-concept monitoring framework prototype using $\mathcal{REC}$ [1, 10, 7], a reactive Event Calculus tool used for run-time monitoring of commitments. The input to a $\mathcal{REC}$ reasoner is the following:

- a *commitments model* that contains the rules for manipulation of commitments,
- a *domain model* that contains the protocol rules that describe the agents' domain,
- an *event trace* that contains the actions of the agents throughout time.

Listing 1 shows part of the commitments model. First, the states of the commitments are described. Then, the rules that describe the state transitions are defined[1]. In $\mathcal{REC}$, we can express that an event *initiates* (or *terminates*) a temporal *fluent*, by way of *initiates(Event, Fluent, Time)* relations. A commitment with its state is considered a temporal fluent.

[1]Note that lines starting with % are comments.

```
% commitment states
conditional(C, T):- holds_at(status(C, conditional), T).
...

% create as conditional
initiates(E, status(C, conditional), T):- ccreate(E, C, T).

% create as active
initiates(E, status(C, active), T):- create(E, C, T).

% conditional to active
terminates(E, status(C1, conditional), T):- detach(E, C1, C2, T).

initiates(E, status(C1, detached), T):- detach(E, C1, _, T).

initiates(E, status(C2, active), T):- detach(E, _, C2, T).

detach(E, cc(X, Y, Q, P, t(T1)), c(X, Y, P, t(T2)), T):-
 conditional(cc(X, Y, Q, P, t(T1)), T),
 initiates(E, Q, T), T2 is T + T1.
...
```

Listing 1. *Commitments model*

Listing 2 shows part of the rules that describe the example domain. Note that an offer from the bank to client creates a conditional commitment between the two agents. The event trace will be given for a sample execution when we present the case study.

```
% fluent manipulation
initiates(exec(pay(Client, Bank, Card)), paid(Card), _):-
 isClient(Client), isBank(Bank), isCard(Card).
...

% commitment manipulation
ccreate(exec(offer(Bank, Client, Card)),
 cc(Bank, Client, paid(Card), delivered(Card), t(7)), _):-
 isBank(Bank), isClient(Client), isCard(Card).
...
```

Listing 2. *Domain model*

Given these inputs, $\mathcal{REC}$ produces an outcome that demonstrates the agents' fluents through time. This is used to monitor the individual states of the commitments at run-time [1, 10]. However, we are not limited to this. We provide exception monitoring via the relations among those commitments. Thus, we extend the commitments model with a similarity model and an exception model.

Listing 3 shows part of the rules that describe delegation-based similarity, and how improper delegations occur (Section 4). Moreover, Listing 4 describes the rules for exceptions.

```
% delegation
explicitDelegation(c(Z, Y, P, _), c(X, Y, P, _)):- X \= Z.
...

delegation(C1, C2):- explicitDelegation(C1, C2).
...

% improper delegation
initiates(_, improperDelegation(
  c(X3, Y3, P3, t(T3)), c(X1, Y1, P1, t(T1))), T):-
 active(c(X1, Y1, P1, t(T1)), T),
 conditional(cc(X2, Y2, Q2, P2, t(T2)), T),
 active(c(X3, Y3, P3, t(T3)), T),
 implicitDelegation(
  cc(X2, Y2, Q2, P2, t(T2)), c(X1, Y1, P1, t(T1))),
 antecedentDelegation(
  c(X3, Y3, P3, t(T3)), cc(X2, Y2, Q2, P2, t(T2))),
 (T2 + T3) > T1.
...
```

Listing 3. *Similarity model*

```
% Monitoring rule 1
initiates(_, exception(C1, C2), T):-
 holds_at(improperDelegation(C1, C2), T).

% Monitoring rule 2
initiates(E, exception(C1, C2), T):-
 holds_at(improperDelegation(C1, C), T),
 active(C2, T), delegation(C, C2).
```

Listing 4. *Exception model*

7 Case Study

Let us consider the protocol in Table 1. It is represented by three commitments. The bank must deliver the credit card within 7 days of the customer's request (CC_1). When the card is requested, the bank notifies the office for printing the card (CC_3). Then, the courier delivers the card to the client (CC_2). Notice the client's role; it only includes the commitment CC_1 and two actions, for requesting and getting the card delivered. The last row of Table 1 shows which agents enact the corresponding roles in the protocol. Consider now the following trace:

$$\mathcal{P}_{card} = \begin{cases} CC_1\text{(bank, client, requested, delivered(7))} \\ CC_2\text{(courier, bank, printed, delivered(3))} \\ CC_3\text{(office, bank, confirmed, printed(3))} \end{cases}$$

$$\mathcal{R}_{client} = \begin{cases} CC_1 \\ \text{request(client, bank)} \rightarrow \text{requested} \\ \text{deliver(X, client)} \rightarrow \text{delivered} \end{cases}$$

$$\mathcal{A} = \left\{ \text{ bank(}\textit{hsbc}\text{), client(}\textit{federico}\text{), courier(}\textit{ups}\text{), office(}\textit{office}\text{) } \right\}$$

Table 1. *Acquire credit card ($\mathcal{P}_{card}$)*

$$\mathcal{T} = \begin{cases} 4 & \text{request(}\textit{federico}\text{, }\textit{hsbc}\text{)} & \text{(the client requests the credit} \\ & & \text{card from the bank on day 4)} \\ 7 & \text{confirm(}\textit{hsbc}\text{, }\textit{office}\text{)} & \text{(the bank confirms the request)} \\ 10 & \text{print(}\textit{office}\text{, }\textit{ups}\text{)} & \text{(the office produces the card and} \\ & & \text{passes it to the courier)} \end{cases}$$

The following commitments are in place at time 11:

$$\mathcal{C}_{11} = \begin{cases} C_1\text{(hsbc, federico, delivered(11))} \\ CC_2\text{(ups, hsbc, printed, delivered(3))} \\ C_3\text{(office, hsbc, printed(10))} \end{cases}$$

Notice the pattern among these three commitments; CC_2 is an implicit delegation of C_1 (Definition 5), and C_3 is an antecedent delegation of CC_2 (Definition 7). Then C_3 is delegation-similar to C_1 via CC_2.

Now assume that no delivery has occurred until time 12. Figure 2 shows the output of j-$\mathcal{REC}$, the Java-based $\mathcal{REC}$ reasoner. The horizontal axis shows the timeline of events. The fluents are positioned vertically, and their truth values (e.g., states for commitments) are computed according to the events. C_1 is indeed violated since its deadline has passed. Because of the similarity relation, CC_2 and C_3's deadlines together affect C_1. Even though the printing of the card is completed at day 10, the courier has 3 more days for delivery, which will eventually exceed C_1's deadline. Indeed, notice in the figure that delivery is completed at time 13, which fulfills the commitment of UPS to HSBC (C_2). However, the commitment of HSBC to Federico (C_1) is violated. We have the fluents *improperDelegation* and *exception*, corresponding to each improper delegation of commitments (both

Figure 2. $\mathcal{REC}$ *output*

conditional and base-level) in the system. Here, the bank should have confirmed the client's request earlier, and notified the office accordingly.

8 Discussion

Work presented in this paper advances the state of the art in several directions. First, we identify the ways that a commitment can be extended with a third party (e.g., a delegatee agent), giving an exhaustive account. We use motivating examples inspired from an e-commerce scenario, to show that delegation can follow meaningful patterns, other than the usual one or two considered in literature. To the best of our knowledge, no systematic classification of commitment delegation types has ever been done before.

Moreover, we provide similarity relations to connect commitments with one another. The relations we propose are again exhaustive, in the sense that they are capture all possible chains of rational delegation. The similarity relations are ternary relations, which connect two (otherwise unrelated) commitments via another existing commitment called primary. Nested delegations are captured (either explicit or implicit) through similarity. Similarity is fundamental to guide the monitoring process.

Finally, we apply such notions to the problem of handling exceptions in contract-regulated systems. We identify possible reasons of exceptions by considering time-related commitments and ways of delegating such commitments that may bring about inconsistent states. We call such delegations improper. We describe a framework that enables commitment monitoring, at run-time, given a specification of the current system state. The framework is able to identify the improper delegations that cause the exception. We implemented the framework in $\mathcal{REC}$ [1, 10, 7].

For future work, we plan to extend the language of commitment properties with negation and disjunction. Commitments with negated propositions are interesting in the sense that an agent commits to ensure something will not happen. This can be related with a maintenance goal [11, 1], where a certain property should hold at all times during a specified interval. Moreover, we plan to build a distributed procedure where agents will collaboratively monitor improper delegations by exchanging their local knowledge.

Acknowledgement

The first author is supported by Boğaziçi University Research Fund under grant BAP5694, and the Turkish State Planning Organization (DPT) under the TAM Project 2007K120610. We thank Marco Montali for providing us with a working implementation of j-$\mathcal{REC}$, which enabled us to run experiments.

References

[1] F. Chesani, P. Mello, M. Montali, and P. Torroni. Commitment tracking via the reactive event calculus. In *IJCAI '09: Proceedings of the 21st International Joint Conference on Artifical Intelligence*, pages 91–96, 2009.

[2] A. K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Reasoning about agents and protocols via goals and commitments. In *AAMAS '10: Proceedings of The 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 457–464, 2010.

[3] A. K. Chopra and M. P. Singh. Constitutive interoperability. In *AAMAS '08: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 797–804, 2008.

[4] A. K. Chopra and M. P. Singh. Multiagent commitment alignment. In *AAMAS '09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 937–944, 2009.

[5] M. Dastani, V. Dignum, and F. Dignum. Role-assignment in open agent societies. In *AAMAS '03: Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems*, pages 489–496, 2003.

[6] G. Friedrich. Repair of service-based processes – an application area for logic programming. The ALP Newsletter, December 2010. http://www.cs.nmsu.edu/ALP/2010/12/.

[7] Ö. Kafalı, F. Chesani, and P. Torroni. What happened to my commitment? Exception diagnosis among misalignment and misbehavior. In *Computational Logic in Multi-Agent Systems*, volume 6245 of *Lecture Notes in Computer Science*, pages 82–98, 2010.

[8] Ö. Kafalı and P. Torroni. Diagnosing commitments: Delegation revisited (extended abstract). In *AAMAS '11: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011.

[9] M. P. Singh. An ontology for commitments in multiagent systems: Toward a unification of normative concepts. *Artificial Intelligence and Law*, 7:97–113, 1999.

[10] P. Torroni, F. Chesani, P. Mello, and M. Montali. Social commitments in time: Satisfied or compensated. In *DALT*, volume 5948 of *Lecture Notes in Computer Science*, pages 228–243. Springer, 2009.

[11] M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: a unifying framework. In *AAMAS '08: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 713–720, 2008.

[12] P. Yolum and M. P. Singh. Flexible protocol specification and execution: applying event calculus planning using commitments. In *AAMAS '02: Proceedings of the 1st International Conference on Autonomous Agents and Multiagent Systems*, pages 527–534, 2002.