Maintenance Commitments and Goals

Pankaj R. Telang
North Carolina State University
Raleigh, NC, USA
prtelang@ncsu.edu

Munindar P. Singh
North Carolina State University
Raleigh, NC, USA
singh@ncsu.edu

Neil Yorke-Smith
American University of Beirut, Lebanon, and
University of Cambridge, UK
nysmith@aub.edu.lb

Technical Report TR-2014-9
NCSU Department of Computer Science

July 2014

Abstract

Goals and social commitments, respectively, capture agents' behavior and interactions. In this technical report we propose a theoretical framework that unifies these constructs with regard to maintenance and facilitates effective cooperation between agents. Our contribution is threefold. (1) We formulate maintenance commitments as a construct in their own right, distinct from achievement commitments to complex temporal formulas. (2) We show how achievement and maintenance commitments and goals can be composed, and give pragmatic patterns of reasoning. (3) We develop an operational semantics that describes the interplay between maintenance commitments and goals. We illustrate our approach on a real-world scenario drawn from the aerospace aftermarket business service domain.

1 Introduction

Let us consider an agent that wishes to maintain a condition, such as an aircraft operator who wishes to always have the aircraft properly serviced by a mechanic. Figure 1 shows a high-level process flow of aerospace aftermarket services [20]. Participants are an airline operator (OPER), an aircraft engine manufacturer (MFR), and a engine parts manufacturer (PMFR).

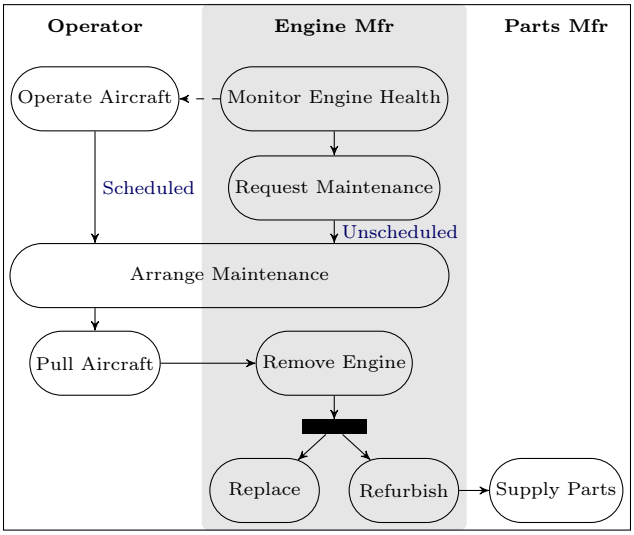

Figure 1: The aerospace aftermarket process [20].

Such situations highlight the need for understanding *maintenance*. Specifically, we address how maintenance arises in connection with goals and commitments, as needed for multiagent systems. In this manner, our work contrasts with previous work on maintenance, which emphasizes single-agent settings and primarily addresses maintenance goals.

We understand cooperation among agents via the interplay between their social state and individual mental states. We consider *social* commitments [3, 16], not to be confused with psychological commitments as in intentions [1, 11]. Consider the case of an aircraft operator, Amy, with a maintenance goal of aircraft reliability. Amy may adopt a series of achievement goals to service the aircraft engine herself. She may act on those goals herself or hire an engine manufacturer, Bob, for each monthly servicing. Or, she may form a maintenance commitment to Bob to pay him a monthly service fee if he would commit to servicing the engine every month. Bob may adopt a maintenance goal to service the engine each month, and act on the goal appropriately.

These simple patterns of cooperation illustrate how the social states (commitments) mediate between the mental states (goals) of the agents. Although commitments and goals are individually well-studied, few works (discussed below) consider them together and no previous work treats maintenance commitments and goals. Moreover, current approaches adequately formalize neither these concepts nor their function in an operational semantics of cooperation.

Contributions and Organization. This technical report introduces maintenance commitments as a construct in their own right and presents a novel operational semantics linking maintenance and achievement goals and commitments. It provides patterns of reasoning that support cooperation and which, under suitable assumptions, guarantee useful properties such as the convergence of the mental and social states of the agents involved.

The document is structured as follows. Section 2 discusses the key background. Section 3 presents our technical framework and Section 4 the operational semantics. Section 5 applies our approach on a well-known case study.

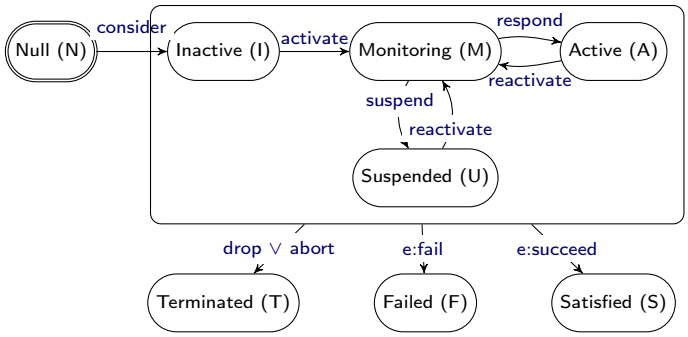

Figure 2: Lifecycle of a maintenance goal.

2 Background

In an achievement commitment, a *debtor* agent commits to a *creditor* agent to bring about a *consequent* condition if some *antecedent* condition holds [3, 16, 17]. We adopt lifecycles of achievement goals and commitments from Telang et al. [18].

Our approach to maintenance recognizes the inherent fallibility of agents in maintaining appropriate conditions in their environment. A condition to be maintained is not ensured to hold continuously, but, every time it fails to hold, the agent *reactively* attempts to resurrect it (as in [2, 9]). Further, the agent may act *proactively* and attempt to preempt failure if it can anticipate the falsehood of the maintenance condition. The above intuition applies equally to maintenance goals and commitments.

We define a maintenance goal and its lifecycle based on Duff et al. [9, 10]. Let $M = \mathsf{M}(x, m, s, f)$ be agent x's *maintenance goal* for condition m. M persists until either its success condition s or its failure condition f become true. $\mathcal{B}_x \models \neg m$ means that x believes that m is false: thus, x adopts a recovery achievement goal. Let π_x capture x's lookahead mechanism (provided by the agent designer), independent of M [9]. Then, $\mathcal{B}_x \models m \wedge \pi_x(\neg m)$ means that x believes that m will become false unless it acts appropriately: thus, x adopts a preventive achievement goal.

Figure 2 depicts a state-based lifecycle for a maintenance goal M [9]. Null means that the goal has not (yet) been created; we explicitly represent only non-Null entities in agents' configurations. The state Monitoring distinguishes maintenance from achievement goals. Once the agent considers it, M starts in the Inactive state; upon its activation, M transitions to the Monitoring state. Here, the agent monitors the (predicted) truth status of condition m. Should x believe that m is not true or will not remain true in the future, it transitions the maintenance goal M to Active, where x creates and adopts a recovery goal R or a preventive goal P, as explained above. M remains Active until m is restored or is no longer predicted to become false, whereupon M returns to Monitoring. Should P or R fail, x retries them. (We lack the space to discuss other potentially useful behaviors.) If x believes that it *cannot* prevent $\neg m$ or restore m, it may fail M. The failure condition f could incorporate this requirement or include the failure of P or R, i.e., x tries once to prevent $\neg m$ or restore m, and fails M if it cannot. We note that an agent so equipped may plan to find other ways to achieve the success condition of P or R [7]. When x transitions M to the Suspended state, it no longer monitors m and aborts any active achievement goals P or R generated by M [19]. Note the labels denote events (e:fail, e:succeed) and actions (all others). The actions are performed by the agent whereas the events are observed in the environment.

Only a few works address maintenance commitments [6, 13]. They treat maintenance commitments as achievement commitments for temporal formulae corresponding to the maintenance of a condition. However, such a representation is in adequate. For example, "always m" does not allow m to fail and be re-established. The lifecycle of maintenance commitments has not been previously studied in these works.

Chesani et al. [4] define commitments with universally quantified properties during a time interval. Their formulation is similar to Mallya et al.'s notion [13], but with greater temporal expressivity and integration. They employ a Reactive Event Calculus framework which allows greater temporal expressiveness than ours.

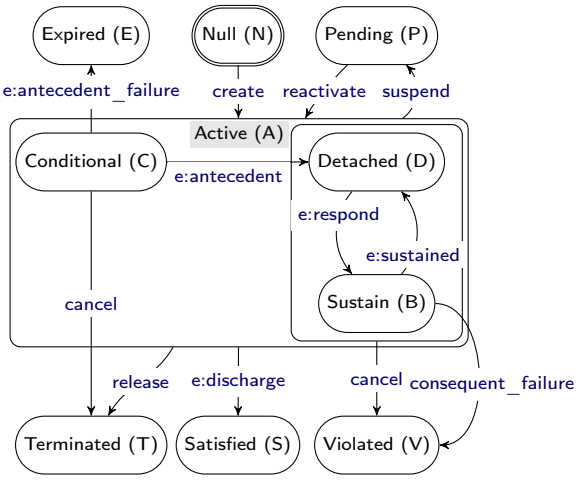

Figure 3: Lifecycle of a maintenance commitment.

However, their work does not have an explicit notion of maintenance commitment, having only a limited maintenance lifecycle.

3 Technical Framework

We treat maintenance commitments as a first-class construct, accommodating both reactive and proactive interpretations, and incorporating enactments wherein the condition may be negated and resurrected. Hence, they cohere well with our treatment of maintenance goals.

A *maintenance commitment* $S = \mathsf{S}(x, y, l, m, d)$ means that debtor x commits to creditor y that if the *antecedent* l is brought about, x will sustain the *maintenance condition* m. (S stands for *sustains*.) Our semantics is agnostic to whether l is brought about by an agent or the environment [17]. S discharges when its *discharge condition* d becomes true; d serves as condition after which the debtor need not maintain m. We sometimes replace irrelevant terms with the placeholder symbol '$\cdot$'.

Figure 3 shows our proposed lifecycle of a maintenance commitment. The substate Sustain distinguishes maintenance from achievement commitments. A maintenance commitment becomes Active upon creation. When the antecedent becomes true, the commitment moves to state Detached. Should the maintenance condition m become false (or be predicted to become false, i.e., $\mathcal{B}_x \models m \wedge \pi_x(\neg m)$), the commitment transitions via the respond transition to state Sustain (label 'B' stands for 'broken'). Here, the agent acts to restore the truth of (or prevent the falsehood of) m, whereupon the commitment returns to Detached. This behavior is akin to the Active–Monitoring states of a maintenance goal; however, Detached–Sustain are more closely linked and form a substate because of their common transitions.

Unlike an achievement commitment, which is Satisfied when its consequent becomes true, a maintenance commitment persists (unless it expires, terminates, or is violated) until its discharge condition d becomes true. The two cancel actions correspond to different source and target states. Here, e:antecedent, e:respond, e:sustained, e:antecedent_failure, and e:discharge are events; all other transitions are actions.

Unlike a maintenance goal, a maintenance commitment has no explicit failure condition because e:consequent_failure resembles a goal's failure condition. A maintenance commitment is violated if e:consequent_failure holds when in Sustain state, or if the debtor cancels the commitment after it has been detached.

Like any commitment, S is part of the mental state of both agents, debtor and creditor. Like any commitment, however, the roles of x and y in S are not identical. Maintaining the condition m is, in particular, the 'responsibility' of the debtor agent x. Hence the commitment does not automatically change

state if y believes m has become false (or y predicts it will become false, i.e., $\mathcal{B}_y \models m \wedge \pi_y(\neg m)$); it only transitions to Sustain if x believes this to be the case. In a situation where y and x have differing beliefs about the state of the world concerning m, agent y might for instance communicate with x to say that y believes x should act about y's perceived falsehood of the maintenance condition.

3.1 Architecture and Assumptions

We consider cognitive agents with a simple architecture. Each agent maintains a set of beliefs, goals, and commitments, and iteratively selects and executes *practical rules*. Practical rules (formalized in Section 4) are reasoning rules that guide the agent's decision making. The beliefs, goals, and commitments are ground clauses [5]: e.g., an achievement commitment to pay \$1 for a book could be represented as $\mathsf{C}(\mathsf{seller}, \mathsf{buyer}, \mathsf{pay\text{-}id_0\text{-}\$1}, \mathsf{deliver\text{-}id_0\text{-}book})$, where the pay term is the antecedent and the deliver term is the consequent. Here, $\mathsf{id_0}$ is a transaction identifier that uniquifies the commitment. Further, for simplicity, we assume the ground clauses are temporally qualified [15]: that is, we do not say "door open"—since the door can become closed and open repeatedly—but "door open at t_0", which is fixed. Actions are initiated by an agent, and events by the environment.

On each iteration, an agent processes any environmental percepts, and updates its beliefs, which may cause state changes of some goals or commitments per the corresponding lifecycle. For example, a goal becomes failed upon its failure condition occurring (being perceived by the agent); a commitment becomes satisfied upon its consequent. Next, the agent executes zero or more practical rules that apply according to the state of a commitment, a goal, or a commitment–goal pair under the constraint that, for each commitment and each goal, the agent selects at most one of the applicable practical rules. The agent executes the chosen rules, which update the state of the affected goals and commitments as per the rule actions. Finally, the agent acts on the environment to bring about conditions that feature in its commitments or performing actions on the commitments. Below, we show how to encode patterns of reasoning about maintenance commitments via practical rules.

The beliefs and goals of each agent are private. As in most previous works on commitments (except [5]), here the agents communicate synchronously, which simplifies alignment [5]: each commitment is represented in the same state by both its debtor and creditor. There is no central store for the social state. The agents *do not* agree upon any actions. Each agent affects its goals; a commitment may be created only by its debtor. Further, the agents may adopt different sets of practical rules. An agent reasons about its goals to select and instantiate zero or more of the applicable practical rules. For example, the ENTICE rule (see later) describes that an agent makes an offer but leaves open what exactly the agent offers. And, of course, the other agent may not be enticed by the offer.

3.2 Grammar of Goals and Commitments

An agent's *configuration* captures both its cognitive state (goals and beliefs) and its social state (commitments of which it is creditor or debtor). Formally, agent z's configuration is the tuple $\mathcal{E}_z = \langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S}, \mathcal{A} \rangle$.

- $\mathcal{B}$ is the set of agent z's beliefs, of the form $\mathsf{B}(z, p)$, about the world, and may include beliefs about itself and others. p is a proposition; we do not allow nested beliefs.
- $\mathcal{G}$ is the set of agent z's achievement goals, i.e., $\mathsf{G}(z, s, f)$, where s and f are propositions (s is the success condition, sought to be achieved, and f the failure condition).
- $\mathcal{M}$ is the set of agent z's maintenance goals, of the form $\mathsf{M}(z, m, s, f)$, where m, s, and f are propositions.
- $\mathcal{C}$ is the set of achievement commitments in which z features as debtor or creditor, i.e., $\mathsf{C}(x, y, u, v)$, where x and y are distinct agents, u and v are propositions (which may include commitments) for the antecedent and consequent respectively, and $z = x$ or $z = y$.
- $\mathcal{S}$ is the set of maintenance commitments in which z features as debtor or creditor, i.e., $\mathsf{S}(x, y, l, m, d)$, where x and y are distinct agents, l, m, and d are propositions (which may include commitments), and $z = x$ or $z = y$.

- $\mathcal{A}$ is a set of pairs; each pair contains a commitment (achievement or maintenance) or a goal (achievement or maintenance), and an action on that commitment or goal: $(C, \mathit{CAction})$, $(S, \mathit{SAction})$, $(G, \mathit{GAction})$, or $(M, \mathit{MAction})$. An agent employs practical rules to set the action for a goal or commitment. If an action is set for a commitment or goal, the commitment or goal transitions to a new state as per the commitment's or goal's life cycle.

Table 1 gives in BNF the formal syntax for configuration components. *Trans* yields a lifecycle transition (e.g., release or e:fail) from the earlier figures. The conditions of goals and commitments are propositional expressions. Since goals are private, it is nonsensical for an agent to form a goal about a goal of another agent. Since goals and commitments (as debtor) are autonomously created, a goal to adopt a goal is immediately satisfied. A goal or commitment to get another party to commit can be satisfied via the agent's capabilities only by infringing upon the other's autonomy. Of course, enticing or otherwise persuading another party to commit is appropriate—indeed, this is the main subject of this technical report and is tackled as explained below. Thus, for simplicity, we eliminate goals that nest the goals or commitments of others. A commitment in a goal condition has the meaning of creating the commitment. For example, G(OPER, C(OPER, MFR, engine_provided, engine_paid)) corresponds to agent OPER having a goal to create the commitment. The goal is satisfied once the nested commitment is created.

$\mathcal{E}$: *AgentState*	$\longrightarrow$	$\langle \mathit{Beliefs}, \mathit{Goals}, \mathit{MGoals}, \mathit{Comms}, \mathit{SComms}, \mathit{As} \rangle$
$\mathcal{B}$: *Beliefs*	$\longrightarrow$	$\mathit{Belief} \langle , \mathit{Belief} \rangle^*$
$\mathcal{G}$: *Goals*	$\longrightarrow$	$\mathit{Goal} \langle , \mathit{Goal} \rangle^*$
$\mathcal{M}$: *MGoals*	$\longrightarrow$	$\mathit{MGoal} \langle , \mathit{MGoal} \rangle^*$
$\mathcal{C}$: *Comms*	$\longrightarrow$	$\mathit{Comm} \langle , \mathit{Comm} \rangle^*$
$\mathcal{S}$: *SComms*	$\longrightarrow$	$\mathit{SComm} \langle , \mathit{SComm} \rangle^*$
$\mathcal{A}$: *As*	$\longrightarrow$	$A \langle , A \rangle^*$
A	$\longrightarrow$	$\langle \mathit{Goal}, \mathit{GAction} \rangle \mid \langle \mathit{MGoal}, \mathit{MAction} \rangle \mid \langle \mathit{Comm}, \mathit{CAction} \rangle \mid \langle \mathit{SComm}, \mathit{SCAction} \rangle$
Belief	$\longrightarrow$	$\mathsf{B}(\mathit{Agent}, \mathbf{atom})$
Goal	$\longrightarrow$	$\mathsf{G}^{\mathit{GState}}(\mathit{Agent}, \mathit{PC}, P)$
GAction	$\longrightarrow$	**consider** | **activate** | **suspend** | **reconsider** | **reactivate** | **terminate**
GState	$\longrightarrow$	**N** | **I** | **A** | **U** | **T** | **F** | **S**
MGoal	$\longrightarrow$	$\mathsf{M}^{\mathit{MState}}(\mathit{Agent}, \mathit{PC}, P, P)$
MAction	$\longrightarrow$	**consider** | **activate** | **suspend** | **reconsider** | **reactivate** | **respond** | **terminate**
MState	$\longrightarrow$	**N** | **I** | **M** | **A** | **U** | **T** | **F** | **S**
Comm	$\longrightarrow$	$\mathsf{C}^{\mathit{CState}}(\mathit{Agent}, \mathit{Agent}, \mathit{PC}, \mathit{PC})$
CAction	$\longrightarrow$	**create** | **suspend** | **reactivate** | **cancel** | **release**
CState	$\longrightarrow$	**N** | **C** | **E** | **D** | **P** | **T** | **V** | **S**
SComm	$\longrightarrow$	$\mathsf{S}^{\mathit{SState}}(\mathit{Agent}, \mathit{Agent}, \mathit{PC}, \mathit{PC}, P)$
SAction	$\longrightarrow$	**create** | **suspend** | **reactivate** | **cancel** | **release** | **respond** | **sustained**
SState	$\longrightarrow$	**N** | **C** | **E** | **D** | **B** | **P** | **T** | **V** | **S**
P	$\longrightarrow$	**atom** $\mid P \wedge P \mid P \vee P \mid \neg P$
PC	$\longrightarrow$	$P \mid \mathit{PC} \wedge \mathit{PC} \mid \mathit{PC} \vee \mathit{PC} \mid \neg \mathit{PC} \mid \mathit{Comm} \mid \mathit{SComm}$

Table 1: Syntax for beliefs, commitments, and goals.

4 Patterns of Reasoning

Our approach is able to formalize important patterns of reasoning with maintenance goals and commitments. We specify a set of practical rules to encode each pattern.

Figure 4 summarizes of the patterns relating a maintenance goal to a maintenance and an achievement commitment. Note that Figure 4 includes an achievement pattern (going from C to a pair G and G [18]), even though it is applied here on top of S. Figure 5 summarizes of the patterns relating an achievement goal to maintenance commitments. We next specify the operational semantics of the practical rules, and then discuss each pattern in turn.

4.1 Operational Semantics

The lifecycle of a commitment or a goal specifies its progression in light of significant events, such as creation and satisfaction. For example, if f holds, a goal whose failure condition is f is necessarily Failed. In other words, the lifecycle captures the hard integrity requirements on our formulation of goals and commitments. The agent may choose as it pleases but if it chooses to create a commitment, the commitment becomes created—there are no two things about it. In contrast, *practical rules* capture patterns of pragmatic reasoning that agents may or may not adopt under different circumstances: they are the rules of an agent program.

A practical rule involves an agent deciding to perform an action. Such a rule is characterized by introducing the action into the agent's $\mathcal{A}$ component. For example, consider a rule that creates a commitment $C = \mathsf{C}(x, y, \mathsf{S}(y, x, l, m, d), q)$ if a maintenance goal $M = \mathsf{M}(x, m, s, f)$ is in state Monitoring. Agent x may choose this rule if it wishes to outsource maintenance of m to some agent y. The following rule shows how a create is inserted into x's configuration. Let $\mathcal{A}' = \mathcal{A} \cup \{\langle C^N, \mathsf{create}\rangle\}$ be the next action component. C^N means that C is in state Null, i.e., the commitment has not yet been created. create is an action that creates the commitment, i.e., moves from Null to Active.

$$\frac{M^M \in \mathcal{M} \quad C \notin \mathcal{C}}{\langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S}, \mathcal{A}\rangle \rightarrow \langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S}, \mathcal{A}'\rangle} \tag{1}$$

The guard condition (above the line) means that maintenance goal M is in state Monitoring (indicated by superscript W) and is in x's set of maintenance goals $\mathcal{M}$, and commitment C is not in x's set of commitments $\mathcal{C}$, i.e., C is Null.

If the guard condition is true, then the agent acts, causing the transition (below the line) to a state in which the create action applied to C is added in the set of commitment actions/events in $\mathcal{A}$. For convenience, we abbreviate Eq. 1 as the guard and the action: $\langle M^M, C^N\rangle$ and $\mathsf{create}(C)$.

Superscripts of M and C are states from the maintenance goal and the maintenance commitment lifecycle, respectively. Further, we employ conjunction or disjunction in the superscript. For example, $M^{A \vee M}$ means $\mathsf{M}(x, m, s, f) \in \mathcal{M}$, and M is either Active or Monitoring. Recall that actions are produced by practical rules and consumed by making transitions in the lifecycles as described in Section 3.1.

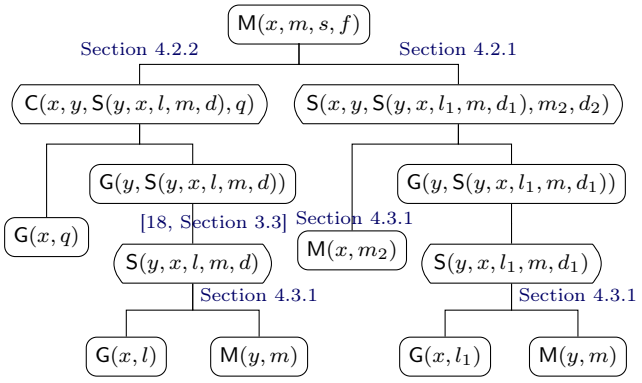

Figure 4: Goal and commitment patterns applied, starting from a maintenance goal.

Table 2: From m-goal to m-commitment.

Rules for goal holder	Guard	Action
ENTICE	$\langle M^M, S^N \rangle$	$\mathsf{create}(S)$
SUSPEND OFFER	$\langle M^U, S^C \rangle$	$\mathsf{suspend}(S)$
REVIVE	$\langle M^M, S^P \rangle$	$\mathsf{reactivate}(S)$
WITHDRAW OFFER	$\langle M^{T \vee F}, S^C \rangle$	$\mathsf{cancel}(S)$
NEGOTIATE	$\langle M^{M \vee U}, S^{E \vee T} \rangle$	$\mathsf{create}(S')$
ABANDON END GOAL	$\langle M^{M \vee U}, S^{E \vee T} \rangle$	$\mathsf{drop}(M)$
SUSPEND END GOAL	$\langle M^M, S_1^D \rangle$	$\mathsf{suspend}(M)$
REACTIVATE END GOAL	$\langle M^U, S_1^{T \vee V} \rangle$	$\mathsf{reactivate}(M)$
RELEASE MAINTENANCE	$\langle M^{T \vee F}, S_1^D \rangle$	$\mathsf{release}(S_1)$

4.2 Patterns Starting with a Maintenance Goal

4.2.1 From M-Goal to M-Commitment

The right branch of Figure 4 shows this pattern. Agent x uses a nested S to make the offer to y. Table 2 lists the practical rules for this pattern. For example, the ENTICE rule has guard $\langle M^M, S^N \rangle$, and action $\mathsf{create}(S)$. The first six rules apply to S, the outer maintenance commitment; the remaining three rules apply to $S_1 = \mathsf{S}(y, x, l_1, m, d_1)$, the nested maintenance commitment.

- ENTICE: x entices y by making an offer S. If y creates S_1, then x commits to maintaining m_2. *Motivation:* x can entice y to maintain M for it by offering to maintain m_2; y may be so enticed if y has a maintenance goal for m_2.
- SUSPEND OFFER: x suspends S if M is suspended. *Motivation:* By suspending S, x informs y that it can work on other tasks instead of sustaining S.
- REVIVE: x reactivates S if M is waiting and S is pending. *Motivate:* x needs S to be active to get y to maintain M.
- WITHDRAW OFFER: x cancels S if M fails or is terminated, and S is conditional. *Motivation:* S is of no utility if M, for which it was created, no longer exists.
- NEGOTIATE: If M is waiting or suspended, and S is expired or terminated, then x creates a new offer S'. *Motivation:* x persists with its goal M by trying alternative ways to induce other agents to cooperate.
- ABANDON END GOAL: If M is waiting or suspended, and S is expired or terminated, x drops M. *Motivation:* Instead of persisting with its goal M, x decides to drop it.
- SUSPEND END GOAL: After detaching S_1, x suspends M. *Motivation:* Since S_1 is detached, x expects y to maintain M, and therefore suspends M.
- REACTIVATE END GOAL: In case y violates S_1, x reactivates M. *Motivation:* By violating the maintenance commitment, y stops maintaining the condition m. Hence, x reactivates M to ensure maintenance of m.
- RELEASE MAINTENANCE: In case M fails or is terminated, and S_1 is detached, then x releases y from S_1. *Motivation:* Since M is failed or terminated, x does not require m to be maintained any longer.

4.2.2 From M-Goal to A-Commitment

There is a similar pattern from M-Goal to A-Commitment, seen in the left branch of Figure 4. Eq. 1 is the ENTICE rule of this pattern. Since the rules of this pattern are similar to those just given, we omit them here for brevity.

4.3 Patterns Starting with Commitments

4.3.1 From M-Commitment to M-Goal and A-Goal

Figure 4 shows three instances of this pattern that starts from a maintenance commitment. For example, starting from $S = \mathsf{S}(y, x, l, m, d)$, x adopts an achievement goal for l, $G = \mathsf{G}(x, l)$, and y adopts a maintenance goal for m, $M = \mathsf{M}(y, m)$. We refer to G and M as *means goals*, since they are the means by which x and y perform their part in S. In this case, M should be scoped such as to maintain m until S discharges, i.e., the success condition s of the maintenance goal M should be the same as the discharge condition d of the maintenance commitment S. Table 3 shows the practical rules for this pattern. We first describe the rules for the creditor x of the maintenance commitment S.

- DETACH and DETACH$'$: If G is null and S is conditional, then x considers and activates G to detach S. *Motivation:* By detaching S, x hopes to influence y to maintain m.
- BACK BURNER: If G is active and S is pending, then x suspends G. *Motivation:* x need not work on G when S is suspended.
- FRONT BURNER: If G is suspended and S is conditional, then x reactivates G. *Motivation:* An active G is necessary to detach S.
- ABANDON MEANS GOAL: If G is active and S is expired or terminated, then x drops G. *Motivation:* G is of no utility if S for which it is created no longer exists.
- PERSIST and PERSIST$'$: If G fails or is terminated and S is conditional, then x considers and activates G' (another instance of G). *Motivation:* x still wants S detached.
- GIVE UP: If G fails or is terminated and S is conditional, x releases y from S. *Motivation:* x gives up detaching S.

Next, we describe the rules for the debtor y of S:

- MAINTAIN and MAINTAIN$'$: If M is null and S is detached, then y considers and activates M to prevent violating its commitment S. *Motivation:* y is cooperative and tries to prevent violating its commitments.
- BACK BURNER: If M is waiting and S is suspended, then y suspends M. *Motivation:* y need not maintain M if S is suspended, and can employ its resources on other tasks.
- FRONT BURNER: If M is suspended and S is detached, then y reactivates M. *Motivation:* y must maintain M to prevent violating S.
- ABANDON MEANS GOAL: If M is waiting and S is terminated or violated, then y drops M. *Motivation:* M is of no utility if S for which it is created ceases to exist.
- PERSIST and PERSIST$'$: If M fails or is terminated and S is detached, then y considers and activates M' (another instance of M). *Motivation:* y persists in preventing violation of its commitment.
- GIVE UP: If M fails or is terminated and S is detached, then y cancels S. *Motivation:* y gives up trying to prevent violation of its commitment.

4.3.2 From Nested A-Commitment to A-Goals

This pattern, not seen in Figure 4, has an offer that nests two maintenance commitments, $C = \mathsf{C}(x, y, \mathsf{S}(y, x, \top, m, d), \mathsf{S}(x, y, \top, l, d))$. Such a commitment can capture a style of joint activity such as: "If you continue to hold up your end of the table, I will continue to hold up my end of the table, and we will both stop when d comes to hold." This pattern combines two patterns: achievement commitment to achievement goals [18], and maintenance commitment to maintenance and achievement goal. The practical rules of these two patterns (Table 3 and [18, Section 3.3]) apply to this pattern. For example, y considers and activates (DETACH and DETACH') an achievement goal for $S_y = \mathsf{S}(y, x, \top, m, d)$ to detach C, and x considers and activates (DELIVER and DELIVER' [18]) an achievement goal for $S_x = \mathsf{S}(x, y, \top, l, d)$. To satisfy the achievement goals, y and x create S_y and S_x, respectively. Then to prevent violating S_y and S_x, y and x consider and activate (MAINTAIN and MAINTAIN$'$) $M_y = \mathsf{M}(y, m)$ and $M_x = \mathsf{M}(x, l)$, respectively.

Table 3: From m-commitment to m-and a-goals.

Rules for creditor	**Guard**	**Action**
DETACH	$\langle G^N, S^C \rangle$	$\mathsf{consider}(G)$
DETACH′	$\langle G^I, S^C \rangle$	$\mathsf{activate}(G)$
BACK BURNER	$\langle G^A, S^P \rangle$	$\mathsf{suspend}(G)$
FRONT BURNER	$\langle G^U, S^C \rangle$	$\mathsf{reactivate}(G)$
ABANDON MEANS GOAL	$\langle G^A, S^{E \vee T} \rangle$	$\mathsf{drop}(G)$
PERSIST	$\langle G^{T \vee F}, S^C \rangle$	$\mathsf{consider}(G')$
PERSIST′	$\langle G^{T \vee F}, G'^I, S^C \rangle$	$\mathsf{activate}(G)$
GIVE UP	$\langle G^{T \vee F}, S^C \rangle$	$\mathsf{release}(S)$
Rules for debtor	**Guard**	**Action**
MAINTAIN	$\langle M^N, S^D \rangle$	$\mathsf{consider}(M)$
MAINTAIN′	$\langle M^I, S^D \rangle$	$\mathsf{activate}(M)$
BACK BURNER	$\langle M^M, S^P \rangle$	$\mathsf{suspend}(M)$
FRONT BURNER	$\langle M^U, S^D \rangle$	$\mathsf{reactivate}(M)$
ABANDON MEANS GOAL	$\langle M^M, S^{T \vee V} \rangle$	$\mathsf{drop}(M)$
PERSIST	$\langle M^{T \vee F}, S^D \rangle$	$\mathsf{consider}(M')$
PERSIST′	$\langle M^{T \vee F}, M'^I, S^D \rangle$	$\mathsf{activate}(M')$
GIVE UP	$\langle M^{T \vee F}, S^D \rangle$	$\mathsf{cancel}(S)$

4.4 Patterns Starting with an Achievement Goal

4.4.1 From A-Goal to M-Commitment

The right branch of Figure 5 shows this pattern. Agent x outsources an achievement goal $G = \mathsf{G}(x, s)$ to y by creating a maintenance commitment $S = \mathsf{S}(x, y, s, m, d)$. That is, x commits to y to maintaining m if y brings about s. The practical rules for this pattern are similar to the first six rules of maintenance goal to achievement commitment pattern from Table 2. Instead of M and S, the rules apply to G and S. For example, the ENTICE rule has guard $\langle G^A, S^N \rangle$, and action $\mathsf{create}(S)$. Note that the last three rules from Table 2 are irrelevant to this pattern since unlike M, G satisfies when S detaches.

4.4.2 From A-Goal to Nested M-Commitment

The left branch of Figure 5 shows this pattern. It is similar to the achievement goal to maintenance commitment pattern except x creates a nested commitment (achievement within maintenance): $S = \mathsf{S}(x, y, \mathsf{C}(y, x, \top, s), m, d)$. That is, x commits to y to maintaining m if y commits to bringing about s. Observe the difference between S from this pattern, and $S' = \mathsf{S}(x, y, s, m, d)$ from achievement goal to maintenance commitment pattern. To detach S, y only needs to *commit* to bringing about s, whereas to detach S', y needs to bring about s. As compared to S', S has greater safety for y [14]. This pattern has the same practical rules as Section 4.4.1.

5 Applying the Theory

We illustrate the value of integrated reasoning over maintenance commitments and goals with the aerospace aftermarket scenario of Figure 1. The operator has an achievement goal to procure an engine, $G_1 = \mathsf{G}(\text{OPER}, \mathsf{engine\_provided})$, and a maintenance goal to keep the engine operational, $M_1 = \mathsf{M}(\text{OPER}, \mathsf{engine\_running}, \mathsf{expiry}, \mathsf{engine\_dead} \vee \mathsf{engine\_sold})$. The engine manufacturer provides engines to the airline operator, and additionally services the engines to keep them operational; in return, the operator pays the manufacturer. That is, the operator commits as follows to the manufacturer: $C_1 = \mathsf{C}(\text{OPER}, \text{MFR}, \mathsf{engine\_provided}, \text{en-}$

Table 4: Progression of configurations in an aerospace scenario.

Step	**Rule**	**OPER's Action**	**OPER's State**	**MFR's Action**	**MFR's State**
1	(structural)—Figure 2	activate(M_1)	$\langle M_1^M \rangle$		$\langle\ \rangle$
2	ENTICE—Figure 4	create(C_2)	$\langle M_1^M, C_2^A \rangle$		$\langle C_2^A \rangle$
3	DETACH—Figure 4		$\langle M_1^M, C_2^D, S^D \rangle$	create(S)	$\langle C_2^D, S^D \rangle$
4	MAINTAIN—Figure 4		$\langle M_1^M, C_2^D, S^D \rangle$	activate(M_2)	$\langle M_2^M, C_2^D, S^D \rangle$
5	SUSPEND END GOAL—Figure 4		$\langle M_1^U, C_2^D, S^D \rangle$		$\langle M_2^M, C_2^D, S^D \rangle$
6	DELIVER—[18]	activate(G_2)	$\langle G_2^A, M_1^U, C_2^D, S^D \rangle$		$\langle M_2^M, C_2^D, S^D \rangle$
7	(structural)—[18]	maint_paid	$\langle G_2^S, M_1^U, C_2^S, S^D \rangle$		$\langle M_2^M, C_2^S, S^D \rangle$
8	(structural)—Figure 2		$\langle \neg$engine_running$, M_1^U, S^B \rangle$		$\langle \neg$engine_running$, M_2^A, S^B \rangle$
9	(structural)—Figure 2		$\langle \neg$engine_running$, M_1^U, S^B \rangle$	activate(R)	$\langle \neg$engine_running$, R^A, M_2^A, S^B \rangle$
10	(structural)—Figure 2		$\langle$engine_running$, M_1^U, S^D \rangle$	engine_running	$\langle$engine_running$, R^S, M_2^M, S^D \rangle$

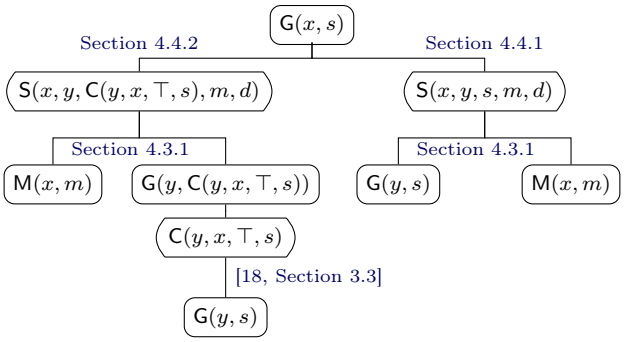

Figure 5: Goal and commitment patterns applied, starting from an achievement goal.

gine_paid), and (following the left branch of Figure 4) C_2 = C(OPER, MFR, S, maint_paid) where S = S(MFR, OPER, engine_paid, engine_running, expiry). The engine_paid and maint_paid conditions model the payments that the operator makes to the manufacturer for the engine and its maintenance. The maintenance commitment S discharges when the maintenance goal M_1 satisfies, that is, the discharge condition (expiry) of S is the success condition of M_1. Let G_2 = G(OPER, maint_paid) be the operator's goal to pay the manufacturer for maintaining the engine. Finally, let M_2 = M(MFR, engine_running, R, P, expiry, engine_dead) be the manufacturer's maintenance goal for the engine, where R and P are goals that the manufacturer believes will (if successful) repair the engine and prevent its failure, respectively.

For space reasons, we apply our approach to a portion of the scenario. We compact a few steps by combining goal consideration (move from initial state to Inactive) and activation (move from Inactive to Active). Suppose that the operator has purchased the engine, i.e., G_1 and C_1 have both succeeded. Table 4 shows a possible progression of the operator and the manufacturer's configurations. The Rule column shows the practical rule along with the origin of that rule. In Step 1, the operator activates the maintenance goal M_1, and in Step 2 employs ENTICE rule to create commitment C_2. In Step 3, the manufacturer employs DETACH rule to create maintenance commitment S. Note that upon creation, S is detached since its antecedent is true (OPER has already purchased the engine). In Step 4, the manufacturer employs MAINTAIN rule to activate M_2; in Step 5, the operator suspends M_1 since S is now detached. In Steps 6–7, the operator activates and satisfies G_2, the goal to pay the manufacturer for engine maintenance. In Step 8, we suppose that the engine fails and stops running, which causes M_2 to activate, and S to sustain. The manufacturer employs RECOVER to activate the recovery goal R to fix the engine in Step 9. In Step 10, MFR fixes the engine, which satisfies R and causes M_2 to transition to Monitoring, and S to Sustain.

6 Discussion and Directions

Summary. We treat maintenance commitments as a first-class construct, enabling our main contribution of a cohesive account of maintenance goals and commitments. In contrast, current approaches (mostly on goals) treat commitments in an ad hoc fashion; and those involving commitments focus on achievement, leaving the treatment of maintenance to implicit procedures rather than logical models. We support important usage patterns.

The net benefit is that our approach facilitates cooperation in a wider variety of realistic settings, especially long-lived interactions where achievement-based approaches simply do not apply. For instance, observe how, in the aerospace scenario, the state of commitments shared between OPER and MFR remains consistent between the two agents, showing how our approach supports cooperation extended over time. We further respect the autonomy of the agents. Agents can choose whether or not to adopt the patterns of reasoning expressed in our semantics as the practical rules.

An agent may refine these rules to always select from among a narrower set of the available options, e.g., through other reasoning about its preferences and utilities. Our approach supports such metareasoning

capability in principle, but we defer a careful investigation of it to future research.

Properties. Our syntax, reductions, and semantics support important properties, discussed here informally. Our proofs, not included in this technical report, employ temporal model checking. Exploring fully formal properties from our semantics is future work.

Section 4's patterns and their rules show how we employ commitments. We illustrate one instance of multiagent progression of our semantics in Table 4. The key assumption behind this work is that the agents are cooperative. Because in general agents are autonomous, they can act as they please, including violating their commitments. Therefore we assume the agents are *realistic*—their commitments gain the interest of the creditor (e.g., in Figure 4's left branch, y has the goal $\mathsf{G}(y, q)$)—and *cooperative*—each debtor is willing to discharge its commitment—and that the agents choose to follow the practical rules of our semantics. Excluding infinite cycles between Monitoring–Suspended or Detached–Sustain states, we can state:

Proposition 1 (Multiagent progress) *Let x and y be a pair of agents that follow Section 4.2.1. Suppose $M_x = \mathsf{M}(x, m, d)$, $M_y = \mathsf{M}(y, m', d')$, $S_{xy} = \mathsf{S}(x, y, S_{yx}, m', d')$, and $S_{yx} = \mathsf{S}(y, x, \top, m, d)$. Then eventually agent x's configuration $\mathcal{E}_x$ contains either $\langle M_x^S, S_{yx}^S \rangle$ or $\langle M_x^{T \vee F}, S_{yx}^{E \vee T \vee V} \rangle$, and $\mathcal{E}_y$ contains either $\langle M_y^S, S_{xy}^S \rangle$ or $\langle M_y^{T \vee F}, S_{xy}^{E \vee T \vee V} \rangle$.*

Directions. Dignum et al. [8] show how to incorporate social attitudes (obligations and norms) within an agent's mental state. They do not consider how agents may choose to adopt the social attitudes because of the mental attitudes, nor vice versa. Dignum et al. do not provide the kinds of cooperation patterns we study here. However, they raise the pertinent topic of agent preferences, incorporating which in our framework is an important future direction.

Günay et al. [12] study dynamic protocol generation wherein agents generate commitments to other agents at runtime, taking into account the (beliefs about the) goals and capabilities of the agents. Their work goes from achievement goals to achievement commitments, whereas our practical rules describe the ways in which achievement and maintenance goals and commitments relate to one another.

For application in a given context, our patterns provide building blocks from which more elaborate protocols can be founded, such as those involving three or more agents. A tool to verify the properties identified above can help to evaluate different sets of practical rules, as a way to assessing their suitability for a particular setting, such as a particular business contract.

Acknowledgments

We thank Amit Chopra for discussions. MPS thanks the National Science Foundation for partial support under grant 0910868. NYS thanks the Operations group at the Judge Business School and the fellowship at St Edmund's College, Cambridge.

References

[1] M. E. Bratman. *Intention, Plans and Practical Reason.* Harvard University Press, Cambridge, MA, 1987.

[2] L. Braubach, A. Pokahr, D. Moldt, and W. Lamersdorf. Goal representation for BDI agent systems. In *Proc. ProMAS*, pages 44–65, 2004.

[3] C. Castelfranchi. Commitments: From individual intentions to groups and organizations. In *Proc. of the International Conference on Multiagent Systems*, pages 41–48, 1995.

[4] F. Chesani, P. Mello, M. Montali, and P. Torroni. Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems*, 27:85–130, 2013.

[5] A. K. Chopra and M. P. Singh. Multiagent commitment alignment. In *Proc. of 8th Intl. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 937–944, 2009.

[6] M. Dastani, L. van der Torre, and N. Yorke-Smith. Monitoring interaction in organisations. In *Proc. COIN'12*, LNCS 7756, pages 17–34. Springer, 2013.

[7] L. de Silva, S. Sardina, and L. Padgham. First principles planning in BDI systems. In *Proc. AAMAS*, pages 1105–1112, 2009.

[8] F. Dignum, D. Morley, E. A. Sonenberg, and L. Cavedon. Towards socially sophisticated BDI agents. In *Proc. Intl. Conf. on Multiagent Systems*, pages 111–118, 2000.

[9] S. Duff, J. Harland, and J. Thangarajah. On proactivity and maintenance goals. In *Proc. AAMAS*, pages 1033–1040, 2006.

[10] S. Duff, J. Thangarajah, and J. Harland. Maintenance goals in intelligent agents. *Computational Intelligence*, 30(1):71–114, 2014.

[11] B. Dunin-Keplicz and R. Verbrugge. *Teamwork in Multi-Agent Systems: A Formal Approach*. Wiley, Chichester, UK, 2010.

[12] A. Günay, M. Winikoff, and P. Yolum. Commitment protocol generation. In *Proc. of DALT'12*, pages 67–82, 2012.

[13] A. U. Mallya, P. Yolum, and M. P. Singh. Resolving commitments among autonomous agents. In *Proc. Workshop on Agent Communication*, pages 166–182, 2003.

[14] E. Marengo, M. Baldoni, C. Baroglio, A. K. Chopra, V. Patti, and M. P. Singh. Commitments with regulations: Reasoning about safety and control in REGULA. In *Proc. AAMAS*, pages 467–474, 2011.

[15] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6(2):101–155, 1982.

[16] M. P. Singh. Social and psychological commitments in multiagent systems. In *Proc. of AAAI Fall Symposium on Knowledge and Action at Social and Organizational Levels*, pages 104–106, 1991.

[17] M. P. Singh. Commitments in multiagent systems: Some history, some confusions, some controversies, some prospects. In F. Paglieri, L. Tummolini, R. Falcone, and M. Miceli, editors, *The Goals of Cognition: Essays in Honor of Cristiano Castelfranchi*, chapter 31, pages 591–616. College Publications, London, 2012.

[18] P. R. Telang, N. Yorke-Smith, and M. P. Singh. A coupled operational semantics for goals and commitments. In *Proc. 9th ProMAS Workshop*, pages 3:1–3:16, 2011.

[19] J. Thangarajah, J. Harland, D. N. Morley, and N. Yorke-Smith. Suspending and resuming tasks in BDI agents. In *Proc. AAMAS*, pages 405–412, Estoril, Portugal, May 2008.

[20] C. J. van Aart, J. Chábera, M. Dehn, M. Jakob, K. Nast-Kolb, J. L. C. F. Smulders, P. P. A. Storms, C. Holt, and M. Smith. Use case outline and requirements. Deliverable D6.1, IST CONTRACT Project, 2007. http://tinyurl.com/6adejz.