Relating Goal and Commitment Semantics

Pankaj R. Telang[1,2], Munindar P. Singh[2], and Neil Yorke-Smith[3,4]

[1] Cisco Systems Inc., Research Triangle Park, NC 27709, USA. prtelang@ncsu.edu
[2] North Carolina State University, Raleigh, NC 27695-8206, USA. singh@ncsu.edu
[3] Olayan School of Business, American University of Beirut, Lebanon.
[4] SRI International, Menlo Park, CA 94025, USA. nysmith@aub.edu.lb

Abstract. Whereas commitments capture how an agent relates with another agent, (private) goals describe states of the world that an agent is motivated to bring about. Researchers have observed that goals and commitments are complementary, but have not yet developed a combined operational semantics for them. This paper makes steps towards such a semantics by relating the respective lifecycles of goals and commitments. We study how the the concepts cohere for one agent and how they engender cooperation between agents. We illustrate our approach via a real-world scenario in the domain of aerospace aftermarket services. We state how our semantics yields important desirable properties, including convergence of the configurations of cooperating agents, thereby delineating some theoretically well-founded yet practical modes of cooperation in a multiagent system.

1 Introduction and Motivation

Whereas the study of goals is a long-standing theme in AI, the last several years have seen the motivation and elaboration of a theory of (social) commitments. The concepts of goals and commitments are intuitively complementary. A commitment describes how an agent relates with another agent, while a goal describes a state of the world that an agent is motivated to bring about. A commitment carries deontic force in terms of what an agent would bring about for another agent, while a goal describes an agent's proattitude toward some condition.

Researchers have begun tying these two concepts together. We go beyond existing works by developing a formal, modular approach that accomplishes the following. First, it characterizes the lifecycles and more generally the operational semantics of the two concepts. Second, it characterizes the interplay between goals and commitments. Third, this approach distinguishes the purely semantic aspects of their lifecycles from the pragmatic aspects of how a cooperative agent may reason. Fourth, it shows that certain desirable properties can be guaranteed for agents who respect selected rules of cooperation. These properties include *convergence*: the agents achieve a level of consistency internally (between the states of their goals and commitments) and externally (between the states of their commitments relevant to each other).

We begin in Sect. 2 by introducing the concepts of commitment and goals, and for each presenting their lifecycle as a state transition diagram. Sect. 3

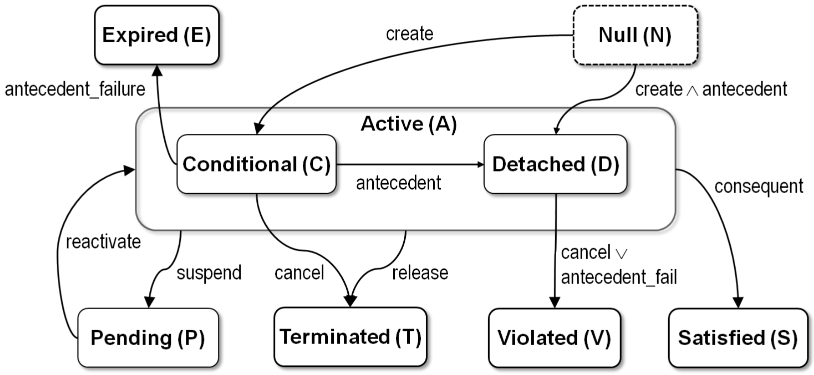

Fig. 1. Commitment lifecycle as a state transition diagram.

presents our combined operational semantics, which is based on guarded rules. We distinguish between two types of rules: mandatory structural rules which reflect the lifecycle of goals and commitments, and practical rules that an agent may choose to follow in order to achieve certain desirable properties. In Sect. 4 we state convergence properties for agents that adopt both types of rules. Sect. 5 illustrates on a real-world scenario, and Sect. 6 places our work in context.

2 Background: Commitments and Goals

Commitments. A *commitment* expresses a social relationship between two agents. Specifically, a commitment C(DEBTOR, CREDITOR, antecedent, consequent) denotes that the DEBTOR commits to the CREDITOR to bringing about the consequent if the antecedent begins to hold [10]. Fig. 1 shows the lifecycle of a commitment simplified from Telang and Singh [12] (below, we disregard timeouts, and commitment delegation or assignment). A labeled rectangle represents a commitment state, and a directed edge represents a transition, labeled with the corresponding action or event.

A commitment can be in one of the following states: Null (before it is created), Conditional (when it is initially created), Expired (when its antecedent remains forever false, while it was still Conditional), Satisfied (when its consequent is brought about while it was Active regardless of its antecedent), Violated (when its antecedent has been true but its consequent will forever be false, or it is canceled when Detached), Terminated (when canceled while Conditional or released while Active), or Pending (when suspended while Active). Active has two substates: Conditional (when its antecedent is false) and Detached (when its antecedent has held) A debtor may create, cancel, suspend, or reactivate a commitment; a creditor may release a debtor from a commitment.

Goals. An agent's desires represent a proattitudes on part of the agent; an agent may concurrently hold mutually inconsistent desires. By contrast, goals

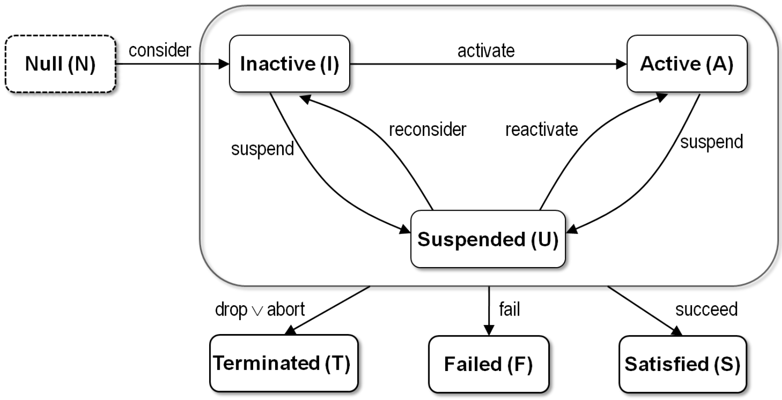

Fig. 2. Simplified lifecycle of an achievement goal as a state transition diagram.

are at least consistent desires: we take a rational agent to believe that its goals are mutually consistent. An agent's intentions are adopted or activated goals.

A *goal* $G = \mathsf{G}(x, p, r, q, s, f)$ of an agent x has a *precondition* (or context) p that must be true before G can become Active and an intention can be adopted to achieve it, a *in-condition* r that is true once G is Active until its achievement, and a *post-condition* (or effect) q that becomes true if G is successfully achieved [17]. The *success condition* s defines the success of G, and the *failure condition* f defines its failure. A goal G is successful iff s becomes true prior to f: that is, the truth of s entails the satisfaction of G only if f does not intervene. Often, the post-condition q and the success condition s coincide, but they need not. As for commitments, the success or failure of a goal depends only on the truth or falsity of the various conditions, not on which agent brings them about.

Fig. 2 simplifies Thangarajah et al.'s [14] lifecycle of an achievement goal (we do not consider maintenance goals). A goal can be in one of the following states: Null, Inactive (renamed from Pending to avoid conflict with commitments), Active, Suspended, Satisfied, Terminated, or Failed. The last three collectively are *terminal states*: once a goal enters any of them, it stays there forever. The semantic rules will link the the definition of a goal G and its states.

Before its creation, a candidate goal is in state Null; once considered by an agent (its "goal holder"), it commences as Inactive, in contrast to commitments which are created in state Active. Upon activation, the goal becomes Active; the agent may pursue its satisfaction by attempting to achieve s. If s is achieved, the goal moves to Satisfied. At any point, if the failure condition of the goal becomes true, the goal moves to Failed. At any point, the goal may enter Suspended, from which it may eventually return to an Inactive or Active state. Lastly, at any point the agent may drop or abort the goal, thereby moving it to the Terminated state.

3 Proposed Operational Semantics

Whereas a goal is specific to an agent (but see Sect. 6), a commitment involves a pair of agents. On the one hand, an agent may create commitments towards other agents in order to achieve its goals. On the other hand, an agent may consider goals in order to fulfill its commitments to other agents.

Chopra et al. [3] formalize a semantic relationship between commitments and goals. They write goals in either or both of the antecedent or consequent of a commitment, i.e., $\mathsf{C}(x, y, g_1, g_2)$, where antecedent ($g_1$) and consequent ($g_2$) are objective conditions (success conditions of one or more goals), not goals. For example, a car insurer may commit to a repair garage to paying if the latter performs a repair: C(INSURER, REPAIRER, car_repaired, payment_made). Here, car_repaired is the success condition of the insurer's goal. The insurer may consider a goal with success condition of payment_made to satisfy the commitment.

3.1 Formal Semantics

We consider the *configuration* of an agent x as the tuple $S_x = \langle \mathcal{B}, \mathcal{G}, \mathcal{C} \rangle$ where $\mathcal{B}$ is its set of beliefs, $\mathcal{G}$ its set of goals, and $\mathcal{C}$ its set of commitments. Conceptually, an agent's configuration relates to both its cognitive and its social state: it incorporates its beliefs and goals as well as its commitments. Where necessary, we index sets or states by agent; for brevity, we omit the parts of the configuration that are clear. We adopt a standard propositional logic.

- $\mathcal{B}$ is the set of x's beliefs about the current snapshot of the world, and include beliefs about itself and other agents. Each snapshot is itself atemporal.
- $\mathcal{C}$ is a set of commitments, of the form $\mathsf{C}(x, y, s, u)$, where x and y are agents and s and u are logical conditions. We use a superscript from Fig. 1 to denote the state of a commitment
- $\mathcal{G}$ is a set of goals adopted by x, of the form $\mathsf{G}(x, p, r, q, s, f)$. $\mathcal{G}$ includes goals that are Inactive. Since the goals in G are adopted, we take it that they are mutually consistent [17]. Superscripts from Fig. 2 denote goal states.

We capture the operational semantics of reasoning about goals and commitments via guarded rules in which S_i are configurations:

$$\frac{guard}{S_1 \longrightarrow S_2} \tag{1}$$

$S_i.\mathcal{B}$, $S_i.\mathcal{G}$, and $S_i.\mathcal{C}$ are the appropriate components of S_i. $S_i \longrightarrow S_j$ is a transition. In most settings, we can specify a family of transitions as an action. For example, for a commitment C, $\mathsf{suspend}(C)$ refers to the set of transitions $S_i \longrightarrow S_j$ where $C \in S_i.\mathcal{C}$ and $\mathsf{suspend}(C) \in S_j.\mathcal{C}$. For actions a and b, $a \wedge b$ indicates that both must be performed.

The same guard may enable multiple transitions $S_i \longrightarrow S_j$ with the same S_i, indicating choice (of the agent involved). For example, intuitively, if a commitment corresponding to a goal expires, an agent could either (i) establish an alternative commitment or (ii) drop the goal. The resulting rules have the same guards, but specify different transitions.

We assume that rational agents seek to achieve their Active goals. That is, an agent at least believes that it has some means to achieve the success condition s of a goal it intends. Either the agent can adopt a plan whose success will achieve s, or it can seek to persuade another agent to bring about the condition s.

3.2 Structural Rules

We distinguish between two types of rules. *Structural* rules specify the progression of a commitment or a goal per their respective lifecycles. Each action that an agent can perform on a goal or a commitment derives a rule of this form. The guard of such a rule is an objective fact. For example, if f holds, a goal whose failure condition is f would be considered as having Failed. Rules such as these capture the hard integrity requirements represented by the lifecycles of goals and commitments. In our particular setting, such rules are both complete and deterministic, in that there is exactly one target state for each potential transition. The state diagrams in Fig. 1 and 2 correspond to the structural rules. The rules are straightforward to derive; we write one rule out in full below, and omit the remainder for reasons of space. We also do not write the standard lifting rule that relates transitions on single commitment/goal to transitions on sets.

A conceptual relationship is established between a goal and a commitment when they reference each other's objective conditions. Even when related in such a manner, however, the goal and the commitment independently progress in accordance with their respective lifecycles. For example, consider a goal $G = \mathsf{G}(x, p, r, q, s, f)$ of agent x. To satisfy this goal, x may create a commitment $\mathsf{C}(x, y, s, u)$. That is, agent x may commit to agent y to bring about u if y brings about s. When y brings about s, C detaches, and G is satisfied. We describe the progression of x's configuration as a structural rule:

$$\frac{\mathcal{B}_x \models s}{\langle G^A, C^A \rangle \longrightarrow \langle G^S, C^D \rangle} \tag{2}$$

where the superscripts denote commitment and goal states from Figs. 1 and 2. Some rules apply in multiple states, indicated via superscripts such as $C^{E \vee T}$.

3.3 Practical Rules

Practical (reasoning) rules capture not necessary integrity requirements, but rather patterns of pragmatic reasoning that agents may or may not adopt under different circumstances. The guard of such a rule is usually the antecedent or consequent of a commitment or the success or failure condition of a goal. The outcome of such a rule can be expressed as an action or an event from the applicable lifecycle diagram, which effectively summarizes a family of transitions from configurations to configurations. For example, an agent having an Active goal may decide to create a commitment as an offer to another agent, in order to persuade the second agent to help achieve its goal. Or, an agent may decide to create a goal to service a commitment.

Such practical rules may be neither complete nor deterministic, in that an agent may find itself at a loss as to how to proceed or may find itself with multiple options. Such nondeterminism corresponds naturally to a future-branching temporal model: each agent's multiplicity of options leads to many possible progressions of its configuration and of the configurations of its peers. The convergence results we show below indicate that our formulated set of rules are complete (i.e., sufficient) in a useful technical sense.

Note that the practical rules are merely options that an agent has available when it adopts these rules as patterns of reasoning—as illustrated in our earlier example of two possible agent actions when a commitment expires. An agent may refine on these rules to always select from among a narrower set of the available options, for example, 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.

It is helpful to group the practical rules into two cases.

Case I: From Goals to Commitments. Here, an agent creates a commitment to satisfy its goal. Consider an agent x having a goal $G = \mathsf{G}(x, p, r, q, s, f)$, and a commitment from x to y: $C = \mathsf{C}(x, y, s, u)$. Notice that s occurs as the success condition of G and the antecedent of C. This case presumes that x lacks (or prefers not to exercise) the capability to bring about s, but can bring about u, and that y can bring about s. Thus x uses C as a means to achieve G (x's *end goal*). Agent x's (goal holder) practical reasoning rules are as follows.

Note that we do not assume that commitments are symmetric. That is, in general, an agent may have a commitment to another agent without the latter having a converse commitment to the former agent.

Recall that superscripts indicate the state of a goal or commitment; for a goal G, the Suspended state is indicated by G^{U}. The guard is a pattern-matching expression. For example, $\langle G^A \rangle$ matches all configurations in which G is Active, regardless of other goals and commitments.

– ENTICE: If G is active and C is null, x creates an offer (C) to another agent.

$$\frac{\langle G^A, C^N \rangle}{\mathsf{create}(C)} \text{ ENTICE} \tag{3}$$

Motivation: (Only) by creating the commitment can the agent satisfy its goal.

– SUSPEND OFFER: If G is suspended, then x suspends C.

$$\frac{\langle G^U, C^A \rangle}{\mathsf{suspend}(C)} \text{ SUSPEND OFFER} \tag{4}$$

Motivation: The agent may employ its resources in other tasks instead of working on the commitment.

– REVIVE: If G is active, and C is pending, then x reactivates C.

$$\frac{\langle G^A, C^P \rangle}{\mathsf{reactivate}(C)} \text{ REVIVE} \tag{5}$$

Motivation: An active commitment is needed by the agent to satisfy its goal.

– WITHDRAW OFFER: If G fails or is terminated, then x cancels C.

$$\frac{\langle G^{T \vee F}, C^A \rangle}{\mathsf{cancel}(C)} \text{ WITHDRAW OFFER} \tag{6}$$

Motivation: The commitment is of no utility once the end goal for which it is created no longer exists.

– REVIVE TO WITHDRAW: If G fails or is terminated and C is pending, then x reactivates C.

$$\frac{\langle G^{T \vee F}, C^P \rangle}{\mathsf{reactivate}(C)} \text{ REVIVE TO WITHDRAW} \tag{7}$$

Motivation: If the goal fails or is terminated, and the commitment is pending, then the agent reactivates the commitment, and later cancels the commitment by the virtue of WITHDRAW OFFER. As per the commitment lifecycle from Fig. 1, an agent needs to reactivate a commitment before cancelling it.

– NEGOTIATE: If C terminates or expires, and G is active or suspended, then x creates another commitment C' to satisfy its goal.

$$\frac{\langle G^{A \vee U}, C^{E \vee T} \rangle}{\mathsf{create}(C')} \text{ NEGOTIATE} \tag{8}$$

Motivation: The agent persists with its goal by trying alternative ways to induce other agents to cooperate.

– ABANDON END GOAL: If C terminates or expires, then x gives up on G.

$$\frac{\langle G^{A \vee U}, C^{E \vee T} \rangle}{\mathsf{drop}(G)} \text{ ABANDON END GOAL} \tag{9}$$

Motivation: The agent may decide no longer to persist with its end goal. Note that an agent may also employ a structural rule to drop a goal without any condition.

It is necessary only that the rules cover *possible* combinations of goal and commitment states. For example, the $\langle G^A, C^V \rangle$ state is not possible since C can violate only after G satisfies; hence no rule is required.

Case II: From Commitments to Goals Here, an agent creates a goal to bring about its part (consequent if debtor, antecedent if creditor) in a commitment.

Consider commitment $C = \mathsf{C}(x, y, s, u)$ and goals $G_1 = \mathsf{G}(x, p, r, q, u, f)$ and $G_2 = \mathsf{G}(y, p', r', q', s, f')$. The practical reasoning rules for agent x are as follows.

– DELIVER: If G_1 is null and C is detached, then x considers and activates goal G_1 to bring about C's consequent.

$$\frac{\langle G_1^N, C^D \rangle}{\mathsf{consider}(G_1) \wedge\ \mathsf{activate}(G_1)} \text{ DELIVER} \tag{10}$$

DELIVER′: If G_1 is inactive and C is detached, then x activates goal G_1 to bring about C's consequent.

$$\frac{\langle G_1^I, C^D \rangle}{\mathsf{activate}(G_1)} \text{ DELIVER'} \tag{11}$$

Motivation: The agent is honest in that it activates a goal that would lead to discharging its commitment.

– BACK BURNER: If G_1 is active and C is pending, then x suspends G_1.

$$\frac{\langle G_1^A, C^P \rangle}{\mathsf{suspend}(G_1)} \text{ BACK BURNER} \tag{12}$$

Motivation: By suspending the goal, the agent may employ its resources to work on other goals.

– FRONT BURNER: If G_1 is suspended and C is detached, then x reactivates G_1.

$$\frac{\langle G_1^U, C^D \rangle}{\mathsf{reactivate}(G_1)} \text{ FRONT BURNER} \tag{13}$$

Motivation: An active goal is necessary for the agent to bring about its part in the commitment.

– ABANDON MEANS GOAL: If G_1 is active and C terminates (y releases x from C) or violates (x cancels C), then x drops G_1.

$$\frac{\langle G_1^A, C^{T \vee V} \rangle}{\mathsf{drop}(G_1)} \text{ ABANDON MEANS GOAL} \tag{14}$$

Motivation: The goal is not needed since the commitment for which it is created no longer exists.

– PERSIST: If G_1 fails or terminates and C is detached, then x activates goal G_1' identical to G_1.

$$\frac{\langle G_1^{T \vee F}, C^D \rangle}{\mathsf{consider}(G_1') \wedge \mathsf{activate}(G_1')} \text{ PERSIST} \tag{15}$$

Motivation: The agent persists in pursuing its part in the commitment.

– GIVE UP: If G_1 fails or terminates and C is detached, then x cancels C.

$$\frac{\langle G_1^{T \vee F}, C^D \rangle}{\mathsf{cancel}(C)} \text{ GIVE UP} \tag{16}$$

Motivation: x gives up pursuing its commitment by cancelling or releasing it.

Many of the practical reasoning rules for agent y are similar to x's.

– DETACH: If G_2 is null and C is conditional, then y considers and activates goal G_2 to bring about C's antecedent.

$$\frac{\langle G_2^N, C^C\rangle}{\mathsf{consider}(G_2)\wedge\ \mathsf{activate}(G_2)}\ \text{DETACH} \tag{17}$$

DETACH′: If G_2 is inactive and C is conditional, then y activates goal G_2 to bring about C's antecedent.

$$\frac{\langle G_2^I, C^C\rangle}{\mathsf{activate}(G_2)}\ \text{DETACH'} \tag{18}$$

Motivation: The creditor brings about the antecedent hoping to influence the debtor to discharge the commitment.

– BACK BURNER: If G_2 is active and C is pending, then y suspends G_2.

$$\frac{\langle G_2^A, C^P\rangle}{\mathsf{suspend}(G_2)}\ \text{BACK BURNER} \tag{19}$$

Motivation: By suspending the goal, the agent may employ its resources to work on other goals.

– FRONT BURNER: If G_2 is suspended and C is conditional, y reactivates G_2.

$$\frac{\langle G_2^U, C^C\rangle}{\mathsf{reactivate}(G_2)}\ \text{FRONT BURNER} \tag{20}$$

Motivation: An active goal is necessary for the agent to bring about its part in the commitment.

– ABANDON MEANS GOAL: If G_2 is active and C expires or terminates (either x cancels, or y releases x from C), then y drops G_2.

$$\frac{\langle G_2^A, C^{E\vee T}\rangle}{\mathsf{drop}(G_2)}\ \text{ABANDON MEANS GOAL} \tag{21}$$

Motivation: The goal is not needed since the commitment for which it is created no longer exists.

– PERSIST: If G_2 fails or terminates and C is conditional, then y activates goal G_2' identical to G_2.

$$\frac{\langle G_2^{T\vee F}, C^C\rangle}{\mathsf{consider}(G_2')\wedge\mathsf{activate}(G_2')}\ \text{PERSIST} \tag{22}$$

Motivation: The agent persists in pursuing its part (either antecedent or consequent) in the commitment.

– GIVE UP: If G_2 fails or terminates and C is conditional, y releases x from C.

$$\frac{\langle G_2^{T\vee F}, C^C\rangle}{\mathsf{release}(C)}\ \text{GIVE UP} \tag{23}$$

Motivation: y gives up pursuing its commitment by cancelling or releasing it.

4 Convergence Properties

We would like to be assured that a coherent world state will be reached, no matter how the agents decide to act, provided that they act according to the rules we have given. We conjuecture that the practical rules are sufficient for an agent to reach a coherent state, as stated in the following set of propositions.

Informally, in a *coherent* state, corresponding goals and commitments align.

Definition 1 *Let* $G = \mathsf{G}(x, p, r, q, s, f)$ *be a goal and* $C = \mathsf{C}(x, y, s, u)$ *a commitment. Then we say that any configuration that satisfies* $\langle G^A, C^A \rangle$, $\langle G^U, C^P \rangle$, $\langle G^{T \vee F}, C^{E \vee T \vee V} \rangle$, *or* $\langle G^S, C^S \rangle$ *is a* coherent state *of* G *and* C.

We have rules that can recreate goals and commitments (namely, PERSIST and NEGOTIATE). These rules could cause endless cycles; therefore we introduce:

Definition 2 *A* progressive rule *is any practical rule other than* PERSIST *and* NEGOTIATE. *The latter two rules we call* nonprogressive.

Propositions 1 and 2 capture the intuition of coherence of a single agent's configuration. All possible agent executions eventually lead to one of the coherent states if the agent obeys our proposed practical rules. They relate to the situations of Case I and Case II respectively.

Proposition 1. *Suppose* $G = \mathsf{G}(x, p, r, q, s, f)$ *and* $C = \mathsf{C}(x, y, s, u)$. *Then there is a finite sequence of progressive rules interleaving finitely many occurrences of nonprogressive rules that leads to a coherent state of* G *and* C. □

Proposition 2. *Suppose* $C = \mathsf{C}(x, y, s, u)$ *and* $G = \mathsf{G}(x, p, r, q, u, f)$. *Then there is a finite sequence of progressive rules interleaving finitely many occurrences of nonprogressive rules that leads to a coherent state of* G *and* C. □

Proposition 3 applies to the configurations of two agents related by a commitment. If the agents obey our proposed practical rules, then the state of the debtor's means goal follows the state of the creditor's end goal.

Proposition 3. (Goal convergence across agents) *Suppose* $G_1 = \mathsf{G}(x, p_1, r_1, q_1, s, f_1)$ *and* $G_2 = \mathsf{G}(y, p_2, r_2, q_2, s, f_2)$ *are goals, and* $C = \mathsf{C}(x, y, s, u)$ *is a commitment. Then there is a finite sequence of rules drawn from the practical rules that leads to* G_2*'s state equaling* G_1*'s state.* □

The formal proof of these propositions is part of our current work [13].

5 Illustrative Application

We illustrate the value of integrated reasoning over commitments and goals with a real-world scenario, drawn from European Union CONTRACT project [15] in the domain of aerospace aftermarket services.

Fig. 3 shows a high-level process flow of aerospace aftermarket services. The participants are an airline operator, an aircraft engine manufacturer, and a parts

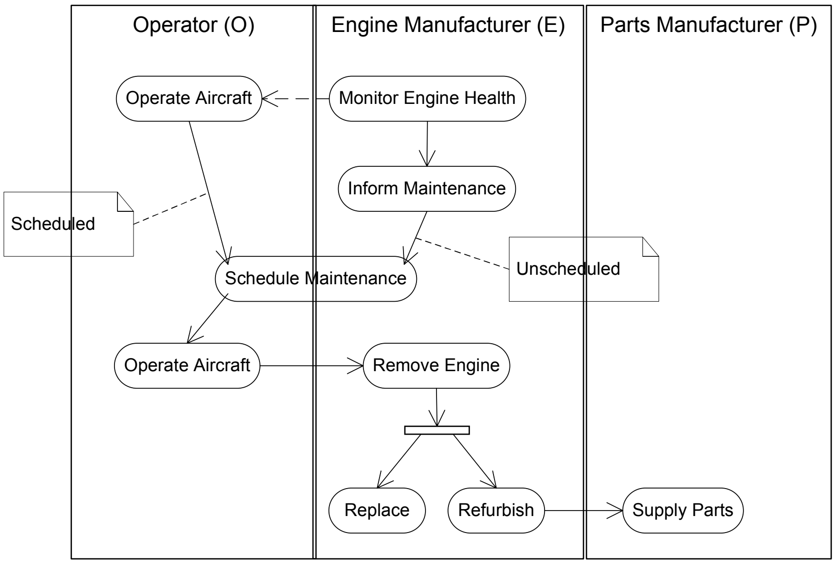

Fig. 3. A high-level model of the aerospace aftermarket process (verbatim from the Amoeba [5] paper, originally from CONTRACT project [15])

manufacturer. 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. If a plane waits on the ground for an engine to be serviced, the manufacturer pays a penalty to the operator. As part of the agreement, the operator regularly provides engine health data to the manufacturer, and may proactively request the manufacturer to perform schedule engine maintenance. The manufacturer analyzes the health data and informs the operator of any required unscheduled engine maintenance. As part of servicing the engine, the manufacturer can either refurbish or replace it. The manufacturer maintains a supply of engines by procuring parts from a parts manufacturer.

Table 1 describes the goals and commitments that model this scenario. For reasons of space, we exclude the airline manufacturer purchasing parts from the parts manufacturer. In the table, service_promised proposition represents creation of C_3 and C_4, and health_reporting_promised represents creation of C_5.

Table 2 describes a possible progression of the aerospace scenario. Each step shows the structural or practical reasoning rule that the airline manufacturer (MFG) or the operator (OPER) employ, and how their configurations progress. For readability, we place new or modified state elements in bold, and omit satisfied commitments and goals in steps subsequent to their being satisfied.

In Steps 1 and 2, the airline manufacturer and the operator consider and activate goals G_1 and G_2. In Step 3, the manufacturer entices (ENTICE rule) the operator to create C_1, which would enable the manufacturer to satisfy G_1.

Table 1. Goals and commitments from the aerospace scenario.

ID	Goal, Commitment, or Event	Description
G_1	G(MFG, $\top$, $\top$, payment_made $\wedge$ health_reporting_promised, payment_made $\wedge$ health_reporting_promised, insufficient_money)	Airline manufacturer's (MFG's) goal to receive the payment and the promise to provide the health report
G_2	G(OPER, $\top$, $\top$, engine_provided $\wedge$ service_promised, engine_provided $\wedge$ service_promised, engine_not_provided)	Operator's (OPER's) goal to receive the engine and the promise to provide the service
G_3	G(OPER, $\top$, $\top$, payment_made $\wedge$ health_reporting_promised, payment_made $\wedge$ health_reporting_promised, insufficient_money)	Operator's goal to make the payment and the promise to provide the health report
G_4	G(MFG, $\top$, $\top$, engine_provided $\wedge$ service_promised, engine_provided $\wedge$ service_promised, engine_not_provided)	Airline Manufacturer's goal to provide the engine and the promise to provide the service
$G_5[i]$	G(OPER, service_needed[i], $\top$, service_requested[i], service_requested[i], service_not_requested[i])	Operator's goal to request the service; there is an instance of this goal for each occurrence of service needed
$G_6[i]$	G(MFG, service_requested[i], $\top$, service_provided[i] , service_provided[i], service_not_provided[i])	Manufacturer's goal to provide the service; there is an instance of this goal for each service request
$G_7[i]$	G(MFG, engine_down[i], $\top$, penalty_paid[i], penalty_paid[i], penalty_not_paid[i])	Manufacturer's goal to pay the penalty if the engine is down; there is an instance of this goal for each engine down occurrence
C_1	C(MFG, OPER, payment_made $\wedge$ health_reporting_promised, engine_provided $\wedge$ service_promised)	Mfr's commitment to operator to provide the engine and service if operator pays and promises to provide the health report
C_2	C(OPER, MFG, engine_provided $\wedge$ service_promised, payment_made $\wedge$ health_reporting_promised)	Operator's commitment to the mfr to pay and to provide the health report if the mfr provides the engine and service
$C_3[i]$	C(MFG, OPER, service_requested[i] $\wedge$ $\neg$expired, service_provided[i])	Mfr's commitment to the operator to provide the service if the operator requests service prior to the contract expiration
$C_4[i]$	C(MFG, OPER, engine_down[i] $\wedge$ $\neg$expired, penalty_paid[i])	Manufacturer's commitment to the operator to pay penalty if the engine is down prior to the contract expiration; there is an instance of this commitment for each occurrence of the engine downtime
$C_5[i]$	C(OPER, MFG, health_report_requested[i] $\wedge$ $\neg$expired, health_report_provided[i])	Operator's commitment to the manufacturer to provide the health report if the manufacturer requests the report; there is an instance of this commitment for each health report request

Notice how ENTICE causes manufacturer's configuration to reach the coherent state $\langle\{G_1^A\},\{C_1^A\}\rangle$. Similarly in Step 4, operator creates C_2.

In Step 5, the manufacturer considers and activates G_4 to detach (DETACH rule) C_2. Observe how DETACH activates manufacturer's (debtor's) means goal G_4, which corresponds to the operator's (creditor's) end goal G_2. In Step 6, the operator considers and activates G_3 to detach C_1.

In Step 7, due to other priorities, the operator decides to suspend G_2. The operator suspends C_2 (SUSPEND OFFER rule) in Step 8, which transitions its configuration to the coherent state $\langle\{G_2^U\},\{C_2^P\}\rangle$. In Step 9, the manufacturer suspends G_4 (BACK BURNER rule), which transitions its configuration to the coherent state $\langle\{G_4^U\},\{C_2^P\}\rangle$. Observe how the practical reasoning rules cause the manufacturer (debtor) to suspend its means goal G_4 in response to the operator (creditor) suspending its end goal G_2. In Step 10–11, the operator reactivates G_2, and reactivates (REVIVE rule) C_2. In Step 12, the manufacturer reactivates (REVIVE rule) G_4.

In Steps 13–15, the manufacturer provides engine (engine_provided) to the operator and creates C_3 and C_4. Recall that service_promised means creation of C_3 and C_4, and satisfaction condition of G_2 and G_4 is engine_provided $\wedge$ service_promised. Therefore, in Step 15, G_2 and G_4 are satisfied. Further since engine_provided $\wedge$ service_promised is consequent of C_1 and antecedent of C_2, in Step 15, C_1 is satisfied and C_2 is detached. In Steps 16–17, operator pays the manufacturer (payment_made), and creates C_5 (health_reporting_promised). This satisfies G_1, G_3, and C_2. Observe how, in Step 17, the practical reasoning rules cause the manufacturer's and the operator's configuration to reach the coherent states $\langle\{G_1^S\},\{C_1^S\}\rangle$ and $\langle\{G_2^S\},\{C_2^S\}\rangle$.

A *service_needed* event occurs at Step 18; it instantiates the parameter i with the value 1. In response, the operator activates $G_5[1]$, an instance of G_5, to request the service in Step 19. By its requesting the service, in Step 20 the operator satisfies $G_5[1]$ and detaches $C_3[1]$, an instance of C_3. To deliver upon its commitment, the manufacturer activates $G_6[1]$ in Step 21, and provides the service in Step 22. This satisfies $G_6[1]$, and $C_3[1]$. Finally, in Step 23, only the recurring commitments C_3, C_4, and C_5 remain in the agent configurations.

6 Related Work

Chopra et al. [3] formalize semantic relationship between agents and protocols encoded as goals and commitments, respectively to verify at design time if a protocol specification (commitments) supports achieving goals in an agent specification, and vice versa. In contrast, our semantics applies at runtime, and we propose practical reasoning rules that agents may follow to achieve coherence between related goals and commitments. Dalpiaz et al. [4] propose a model of agent reasoning based on pursuit of *variants*—abstract agent strategies for pursuing a goal. We conjecture that their approach can be expressed as sets of practical reasoning rules, such as those we described above.

Table 2. Progression of configurations in the aerospace scenario.

#	Event or Rule	MFG's Action	MFG's State	OPER's Action	OPER's State
1	(structural)	consider(G_1) $\wedge$ activate(G_1)	$\langle\{\boldsymbol{G_1^A}\}\rangle$		$\langle\ \rangle$
2	(structural)		$\langle\{G_1^A\}\rangle$	consider(G_2) $\wedge$ activate(G_2)	$\langle\{\boldsymbol{G_2^A}\}\rangle$
3	ENTICE	create(C_1)	$\langle\{G_1^A\},\{\boldsymbol{C_1^C}\}\rangle$		$\langle\{G_2^A\},\{\boldsymbol{C_1^C}\}\rangle$
4	ENTICE		$\langle\{G_1^A\},\{C_1^C,\boldsymbol{C_2^C}\}\rangle$	create(C_2)	$\langle\{G_2^A\},\{C_1^C,\boldsymbol{C_2^C}\}\rangle$
5	DETACH	consider(G_4) $\wedge$ activate(G_4)	$\langle\{G_1^A,\boldsymbol{G_4^A}\},\{C_1^C,C_2^C\}\rangle$		$\langle\{G_2^A\},\{C_1^C,C_2^C\}\rangle$
6	DETACH		$\langle\{G_1^A,G_4^A\},\{C_1^C,C_2^C\}\rangle$	consider(G_3) $\wedge$ activate(G_3)	$\langle\{G_2^A,\boldsymbol{G_3^A}\},\{C_1^C,C_2^C\}\rangle$
7	(structural)		$\langle\{G_1^A,G_4^A\},\{C_1^C,C_2^C\}\rangle$	suspend(G_2)	$\langle\{\boldsymbol{G_2^U},G_3^A\},\{C_1^C,C_2^C\}\rangle$
8	SUSPEND OFFER		$\langle\{G_1^A,G_4^A\},\{C_1^C,\boldsymbol{C_2^P}\}\rangle$	suspend(C_2)	$\langle\{G_2^U,G_3^A\},\{C_1^C,\boldsymbol{C_2^P}\}\rangle$
9	BACK BURNER	suspend(G_4)	$\langle\{G_1^A,\boldsymbol{G_4^U}\},\{C_1^C,C_2^P\}\rangle$		$\langle\{G_2^U,G_3^A\},\{C_1^C,C_2^P\}\rangle$
10	(structural)		$\langle\{G_1^A,G_4^U\},\{C_1^C,C_2^P\}\rangle$	reactivate(G_2)	$\langle\{\boldsymbol{G_2^A},G_3^A\},\{C_1^C,C_2^P\}\rangle$
11	REVIVE		$\langle\{G_1^A,G_4^U\},\{C_1^C,\boldsymbol{C_2^C}\}\rangle$	reactivate(C_2)	$\langle\{G_2^A,G_3^A\},\{C_1^C,\boldsymbol{C_2^C}\}\rangle$
12	REVIVE	reactivate(G_4)	$\langle\{G_1^A,\boldsymbol{G_4^A}\},\{C_1^C,C_2^C\}\rangle$		$\langle\{G_2^A,G_3^A\},\{C_1^C,C_2^C\}\rangle$
13	(structural)	engine_provided	$\langle\{G_1^A,G_4^A\},\{C_1^C,C_2^C\}\rangle$		$\langle\{G_2^A,G_3^A\},\{C_1^C,C_2^C\}\rangle$
14	(structural)	create(C_3)	$\langle\{G_1^A,G_4^A\},\{C_1^C,C_2^C,\boldsymbol{C_3^C}\}\rangle$		$\langle\{G_2^A,G_3^A\},\{C_1^C,C_2^C,\boldsymbol{C_3^C}\}\rangle$
15	(structural)	create(C_4)	$\langle\{G_1^A,\boldsymbol{G_4^S}\},\{\boldsymbol{C_1^S},\boldsymbol{C_2^D},C_3^C,\boldsymbol{C_4^C}\}\rangle$		$\langle\{\boldsymbol{G_2^S},G_3^A\},\{\boldsymbol{C_1^S},\boldsymbol{C_2^D},C_3^C,\boldsymbol{C_4^C}\}\rangle$
16	(structural)		$\langle\{G_1^A\},\{C_2^D,C_3^C,C_4^C\}\rangle$	payment_made	$\langle\{G_3^A\},\{C_2^D,C_3^C,C_4^C\}\rangle$
17	(structural)		$\langle\{\boldsymbol{G_1^S}\},\{\boldsymbol{C_2^S},C_3^C,C_4^C,\boldsymbol{C_5^C}\}\rangle$	create(C_5)	$\langle\{\boldsymbol{G_3^S}\},\{\boldsymbol{C_2^S},C_3^C,C_4^C,\boldsymbol{C_5^C}\}\rangle$
18	service_needed[1]		$\langle\{C_3^C,C_4^C,C_5^C\}\rangle$		$\langle\{C_3^C,C_4^C,C_5^C\}\rangle$
19	DETACH		$\langle\{C_3^C,C_4^C,C_5^C\}\rangle$	consider(G_5[1]) $\wedge$ activate(G_5[1])	$\langle\{\boldsymbol{G_5^A[1]}\},\{C_3^C,C_4^C,C_5^C\}\rangle$
20	(structural)		$\langle\{\boldsymbol{C_3^D[1]},C_3^C,C_4^C,C_5^C\}\rangle$	service_requested[1]	$\langle\{\boldsymbol{G_5^S[1]}\},\{\boldsymbol{C_3^D[1]},C_3^C,C_4^C,C_5^C\}\rangle$
21	DELIVER	consider(G_6[1]) $\wedge$ activate(G_6[1])	$\langle\{\boldsymbol{G_6^A[1]}\},\{C_3^D[1],C_3^C,C_4^C,C_5^C\}\rangle$		$\langle\{C_3^D[1],C_3^C,C_4^C,C_5^C\}\rangle$
22	(structural)	service_provided	$\langle\{\boldsymbol{G_6^S[1]}\},\{\boldsymbol{C_3^S[1]},C_3^C,C_4^C,C_5^C\}\rangle$		$\langle\{\boldsymbol{C_3^S[1]},C_3^C,C_4^C,C_5^C\}\rangle$
23			$\langle\{C_3^C,C_4^C,C_5^C\}\rangle$		$\langle\{C_3^C,C_4^C,C_5^C\}\rangle$

Winikoff [16] develops a mapping from commitments to BDI-style plans. He modifies SAAPL, an agent programming language, to include commitments in an agent's belief-base and operational semantics update the commitments. Our operational semantics addresses goals (more abstract than plans) and commitments. It will be interesting to combine Winikoff's work with ours to develop a joint semantics for commitments, goals, and plans.

Avali and Huhns [1] relate an agent's commitments to its beliefs, desires, and intentions using BDI_{CTL*}. In contrast, we relate an agent's commitments to its goals. We consider goal lifecycle in our semantics, and propose practical reasoning rules for coherence with commitments.

Telang and Singh [11] enhance Tropos, an agent-oriented software engineering methodology, with commitments. They describe a methodology that starts from a goal model and derives commitments. Our operational semantics complements by providing a formal underpinning.

Telang and Singh [12] propose a commitment-based business metamodel, a set of modeling patterns, and an approach for formalizing the business models and verifying message sequence diagrams with respect to the models. Our combined operational semantics of commitments and goals can provide a basis for how a business model can be enacted and potentially support the derivation of suitable message sequence diagrams.

van Riemsdijk et al. [9] and Thangarajah et al. [14] propose abstract architectures for goals, on which is based the simplified goal lifecycle that we consider. These and other authors formalize the goal operationalization. In contrast, our work formalizes the combined operational semantics of goals and commitments. A future extension of our work is to address the different goal types that have been suggested [9, 18]. Our work is complementary also to exploration of goals that have temporal extent (e.g., [2, 7]). Moreover, we have considered each goal to be private to an agent. Works that study coordination of agents via shared proattitudes, such as shared goals, include for example Grosz and Kraus [6] and Lesser et al. [8].

7 Conclusion and Future Work

This paper studied the complementary aspects of commitments and goals by establishing an operational semantics of the related lifecycles of the two concepts. We have distinguished the purely semantic aspects of their lifecycles from the pragmatic aspects of how a cooperative agent may reason, and stated desirable properties such as convergence of mental states. These proporties need to be formally proved. From the viewpoint of agent programming, we have sought to provide a foundational set of rules that is complete in a technical sense; their sufficiency in practice will be found through use.

Our work carries importance because of its formalization of the intuitive complementarity between goals and commitments. Directions for building on this foundation include considering a hierarchy of prioritized goals or commitments, and extending our semantics to include maintenance goals, shared goals, or plans.

We are also interested in examining convergence properties when there are more than two agents working collaboratively.

Acknowledgments. We gratefully acknowledge the suggestions of the anonymous reviewers of the ProMAS'11 workshop and the post-proceedings volume, and the discussions with the participants at the workshop.

References

1. Avali, V.R., Huhns, M.N.: Commitment-based multiagent decision making. In: Proc. Cooperative Information Agents. pp. 249–263 (2008)
2. Braubach, L., Pokahr, A.: Representing long-term and interest BDI goals. In: Proc. ProMAS Workshop (2009)
3. Chopra, A.K., Dalpiaz, F., Giorgini, P., Mylopoulos, J.: Reasoning about agents and protocols via goals and commitments. In: Proc. AAMAS. pp. 457–464 (2010)
4. Dalpiaz, F., Chopra, A.K., Giorgini, P., Mylopoulos, J.: Adaptation in open systems. In: Proc. 29th Conf. Conceptual Modeling. pp. 31–45 (2010)
5. Desai, N., Chopra, A.K., Singh, M.P.: Amoeba: A methodology for modeling and evolution of cross-organizational business processes. ACM Trans. Software Engineering and Methodology 19(2), 6:1–6:45 (2009)
6. Grosz, B., Kraus, S.: Collaborative plans for complex group action. Artificial Intelliegence 86(2), 269–357 (1996)
7. Hindriks, K.V., van der Hoek, W., van Riemsdijk, M.B.: Agent programming with temporally extended goals. In: Proc. AAMAS. pp. 137–144 (2009)
8. Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P., Zhang, X.: Evolution of the GPGP/TAEMS Domain-Independent Coordination Framework. J. Autonomous Agents and Multi-Agent Systems 9(1), 87–143 (2004)
9. van Riemsdijk, M.B., Dastani, M., Winikoff, M.: Goals in agent systems. In: Proc. AAMAS. pp. 713–720 (2008)
10. Singh, M.P.: An ontology for commitments in multiagent systems. AI and Law 7, 97–113 (1999)
11. Telang, P.R., Singh, M.P.: Enhancing Tropos with commitments. In: Conceptual Modeling: Foundations and Applications. pp. 417–435. LNCS 5600, Springer (2009)
12. Telang, P.R., Singh, M.P.: Specifying and verifying cross-organizational business models. IEEE Trans. Services Comput. 4 (2011)
13. Telang, P.R., Singh, M.P., Yorke-Smith, N.: A coupled operational semantics for goals and commitments (2012), working paper
14. Thangarajah, J., Harland, J., Morley, D., Yorke-Smith, N.: Operational behaviour for executing, suspending and aborting goals in BDI agent systems. In: Proc. DALT Workshop. pp. 1–17 (2010)
15. van Aart, C.J., Chábera, J., Dehn, M., Jakob, M., Nast-Kolb, K., Smulders, J.L.C.F., Storms, P.P.A., Holt, C., Smith, M.: Use case outline and requirements. Deliverable D6.1, IST CONTRACT Project (2007), http://tinyurl.com/6adejz
16. Winikoff, M.: Implementing commitment-based interactions. In: Proc. AAMAS. pp. 873–880 (2007)
17. Winikoff, M., Padgham, L., Harland, J., Thangarajah, J.: Declarative and procedural goals in intelligent agent systems. In: Proc. KR. pp. 470–481 (2002)
18. Winikoff, M., Dastani, M., van Riemsdijk, M.B.: A unified interaction-aware goal framework. In: Proc. ECAI. pp. 1033–1034 (2010)