TU Delft

Delft University of Technology

Maintenance commitments

Conception, semantics, and coherence

Telang, Pankaj R.; Singh, Munindar P.; Yorke-Smith, Neil

DOI
10.1016/j.artint.2023.103993

Publication date
2023

Document Version
Final published version

Published in
Artificial Intelligence

Citation (APA)
Telang, P. R., Singh, M. P., & Yorke-Smith, N. (2023). Maintenance commitments: Conception, semantics, and coherence. *Artificial Intelligence*, *324*, Article 103993. https://doi.org/10.1016/j.artint.2023.103993

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

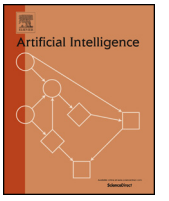
This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Artificial Intelligence 324 (2023) 103993

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



Maintenance commitments: Conception, semantics, and coherence

Pankaj Telang [a], Munindar P. Singh [b], Neil Yorke-Smith [c,*]

[a] SAS Institute, Cary, NC 27513, USA
[b] North Carolina State University, Raleigh, NC 27695, USA
[c] Delft University of Technology, Delft, 2600GA, the Netherlands

ARTICLE INFO

Article history:
Received 7 June 2021
Received in revised form 7 August 2023
Accepted 9 August 2023
Available online 19 August 2023

Keywords:
Social commitments
Maintenance
Goals
Belief-Desire-Intention agents
Coherence

ABSTRACT

Social commitments are recognized as an abstraction that enables flexible coordination between autonomous agents. We make these contributions. First, we introduce and formalize a concept of a *maintenance commitment*, a kind of social commitment characterized by a maintenance condition whose truthhood an agent commits to maintain. This concept of maintenance commitments enables us to capture a richer variety of real-world scenarios than possible using achievement commitments with a temporal condition. Second, we develop a rule-based operational semantics, by which we study the relationship between agents' achievement and maintenance goals, achievement commitments, and maintenance commitments. Third, we motivate a notion of coherence between an agents' achievement and maintenance cognitive and social constructs, and prove that, under specified conditions, the goals and commitments of both rational agents individually and of a multiagent system altogether are coherent. Fourth, we illustrate our approach with a detailed real-world scenario from an aerospace aftermarket domain.

© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Social commitments enable flexible coordination between autonomous agents. The literature primarily focuses on achievement commitments [1–4]. This form of commitment captures a bilateral social contract between two agents: if one agent (the *creditor*) brings about a condition in the world, then the other agent (the *debtor*) will bring about some other condition. Achievement commitments do not capture scenarios in which an agent commits to *maintain* a condition in the world.

Prior research has not adequately addressed maintenance commitments, treating them instead as achievement commitments for temporal formulae of the form 'always in the future p' [5,6], where p is some *maintenance condition*. Such formulation requires the maintenance condition to be always true. Specifically, it disallows the cases in which the maintenance condition becomes false and the agent acts to restore it. This is typical of many real-world maintenance scenarios.

Indeed, such a formulation cannot capture aspects of real-world situations:

- Consider a lawn-care scenario in which a landscape gardener commits to a homeowner to maintaining the lawn in a green condition. The lawn may occasionally turn non-green. In that case, the gardener may treat it and restore it to

* Corresponding author.
E-mail addresses: ptelang@gmail.com (P. Telang), singh@ncsu.edu (M.P. Singh), n.yorke-smith@tudelft.nl (N. Yorke-Smith).

https://doi.org/10.1016/j.artint.2023.103993
0004-3702/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

green. If a commitment is formulated using the above temporal formula, it will violate if the lawn turns non-green, and it will fail to accommodate such execution.

- Consider a mortgage scenario in which a homeowner commits to paying a lender incrementally until a loan is discharged. The homeowner may occasionally miss a payment due to some economic issues. In that case, the homeowner may make the missed payment (with some penalty) later. If a commitment is formulated using the above temporal formula, it will violate if the homeowner misses a payment, and it will fail to accommodate such execution.
- Consider an aerospace scenario in which an aircraft manufacturer commits to an airline operator to maintaining an aircraft engine in running condition. Over time, the engine may not stay in running condition due to mechanical wear. In that case, the aircraft manufacturer may service the engine and restore it to its running condition. If a commitment is formulated using the above temporal formula, it will be violated if the engine is not in running condition, and it will fail to accommodate such execution.

Such situations highlight the need for understanding maintenance. We motivate a new family of social commitments wherein a debtor agent commits to a creditor agent that if some *antecedent* condition holds it would *maintain* a *consequent* condition until some *discharge* condition holds. Maintenance here means ensuring that the consequent condition does not become false or, if it does become false, then to re-establish its truthhood. 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 [7].

Social commitments are a natural basis for modeling interaction between agents by representing the meanings of communication [8–11] and for reasoning about safety and control [12]. Understanding maintenance commitments opens up the realm of social interaction. As such, a major theme of this article is capturing the dynamic relationships between an agent's beliefs and goals (i.e., cognitive state) and its commitments (i.e., social state). Maintenance commitments enable richer relationships than otherwise possible, thereby supporting expanded forms of collaboration. Specifically, a commitment can relate to goals and a goal can relate to commitments. For example, (1) an end goal of paying for a house may lead one to a maintenance commitment to the lender of the mortgage: paying the lender incrementally until the loan is paid up. (2) The mortgage commitment may lead one to adopt a maintenance goal of making loan payments, which (3) may lead one to commit to doing a job for an employer if the employer pays a salary every month.

In contrast to existing approaches, we treat maintenance commitments as a first-class construct, accommodating a reactive interpretation, and incorporating cases wherein the condition may be falsified and then made true again. Our formal operational semantics develops two sets of conditional rules. First, *life cycle* rules specify the mandatory progression of goal and commitment states as the agents update their beliefs or perform 'social actions' on the goals and commitments. Second, *practical* rules represent patterns of reasoning that specify potential social actions for an agent based on its goals and commitments.

Our previous research characterizes *coherence* between a rational agent's (achievement) commitments and its (achievement) goals [4]. Building on this approach, we motivate an enhanced notion of coherence and with it study the synergy between an agent's maintenance commitments and its achievement and maintenance goals. We also show how coherence applies to a multiagent system as a whole with respect to specific maintenance commitments and achievement and maintenance goals.

As first demonstrated in our previous work, the value of interconnecting individual practical reasoning (involving goals) with the social aspects of reasoning (here, the use of commitments)—in the form of a unified theory of commitments and goals—is significant for the following two reasons. First, it would close the theoretical gap in present understanding between the organizational and individual perspectives, which are both essential in a comprehensive account of rational agency in a social world. Second, it would provide a basis for a comprehensive account of the software engineering of multiagent systems going from interaction-orientation (with commitments) to agent-orientation (with goals). In the sequel we discuss the motivation and implications of interconnecting both achievement and maintenance cognitive and social constructs.

This article advances the state of the art as follows. First, we introduce a new powerful type of social commitment, along with its life cycle. Second, we specify a formal semantics of multiagent system configuration, encompassing both achievement and maintenance commitments and goals. Third, we provide a methodology and an exemplar set of practical rules. Fourth, we define coherence and prove conditions under which it is maintained in the multiagent system.

A preliminary version of this work appeared at a conference [13]. This article significantly extends on that conference report by giving a full description of the operational semantics, including proofs of theoretical results, and developing a detailed example in a real-world scenario. The remainder of the article is structured as follows. Section 2 further motivates the value of combined reasoning with commitments and goals. Section 3 provides necessary background. Section 4 introduces maintenance commitments. Section 5 specifies life cycle rules. Section 6 relates goals and commitments by specifying a set of practical rules. Section 7 illustrates our operational semantics with the aerospace aftermarket case study. Section 8 provides formal coherence theorems. Section 9 reviews related literature. Section 10 concludes the article.

2. Motivation for integrated reasoning

The introduction expresses the need for an adequate concept of maintenance commitment, and the benefits of such a first-class construct. We posit that developing a unified theory of achievement and maintenance commitments and goals would be significant: from both agent-theoretical and software engineering perspectives in temporally-extended scenarios.

Therefore, this article studies the dynamic relationships between a rational agent's cognitive state (i.e., beliefs and goals) and its social state (i.e., commitments). First-class maintenance commitments are a powerful new type of social commitments that enable richer relationships than otherwise possible, thereby supporting expanded forms of collaboration. The earlier examples demonstrated in an informal way a set of scenarios that are not expressible by achievement commitments.

Our formal operational semantics, presented later in the article, adds value to the understanding of intra-agent deliberation and inter-agent collaboration, and to the specification and implementation of agent systems. Our formalization captures the combined life cycles of commitments and goals—both achievement and maintenance—and provides a set of practical rules by which agents may reason about their commitments and goals in tandem. Further, under suitable constraints about the autonomy of the agents and some assumptions about belief consistency between the agents, we analytically prove results about the coherence of commitments and goals in the multiagent system.

Consider two examples of the dynamic interplay between commitments and goals. First, goal-to-commitment: An agent Alice may have a goal that Alice cannot satisfy on her own for reasons such as a lack of capabilities or a lack of resources. In this case, Alice may create a commitment to another agent, Bob, such that Bob may be enticed to bring about the condition of Alice's goal in return for what Alice has committed to do.

Second, commitment-to-goal: The agent may create a goal to bring about the antecedent or consequent conditions of a commitment. Having the goal is the first step in achieving the relevant condition, e.g., by allocating resources to it. An example of a goal to bring about the antecedent of a commitment is the homeowner paying the invoice from the gardener (who thus takes care of the lawn). An example of a goal to bring about the consequent of a commitment is the gardener adopting a goal to inspect (and treat if necessary) the lawn weekly.

Thus, practical reasoning rules help ensure the agents do not have any unnecessary goals or commitments. We introduce formal concepts of 'support', by which one cognitive construct is instrumental to the agent's pursuit of another. For example, if an agent creates a goal in support of a commitment, and subsequently cancels the commitment, then the agent may reasonably cancel the goal (assuming that goal does not support some other commitment). Similarly, if an agent creates a commitment in support of a goal, and subsequently cancels the goal, then the agent may reasonably cancel the commitment (assuming that commitment does not support some other goal). In particular, we introduce the notion of *antecedent support*, which enables an agent to operate on goals in order to bring about the antecedent condition of a maintenance commitment.

While we select models of maintenance commitments (and the other cognitive constructs), and social reasoning patterns, our methodology in this article generalizes. For instance, an emphasis on norm enforcement, as in work by Dastani et al. [14], would result in a different set of maintenance commitment actions than in this article, but the methodology to develop the operational semantics would hold the same.

The practical benefit of our contribution to the agent designer is through tasks such as automatic protocol generation [15], failure handling, and high-level agent programming of BDI-style agents with the social state [16]. Maintenance commitments are particularly prominent in scenarios involving extended transactions and service contracts, as Section 7 illustrates.

3. Preliminaries and background

This section provides the necessary background for the contributions that follow in the article.

3.1. Agent architecture

Our formal approach follows the methodology of Telang et al. [4], hereinafter TSY. Thus Fig. 1, adapted from TSY, describes how an agent operates with respect to its beliefs, goals, and commitments. The simple agent architecture provides an illustrative context for our semantics; it is not intended to be an alternative to the fully-fledged architectures in the literature.

Each agent maintains a set of beliefs, goals (both achievement and maintenance), and commitments (both achievement and maintenance), denoted by small caps labels. The agent executes iteratively in a control loop. Based on its perception of the environment, the agent updates its beliefs and updates the goal and commitment states according to their life cycles. The agent then executes the *practical rules* of reasoning that apply. These practical rules, described in Section 6.2, capture patterns of pragmatic reasoning that agents may or may not adopt under different circumstances. In that sense, practical rules are the rules of an agent program. They are specified by the designers of the agents.

The practical rules apply according to the state of a commitment, a goal, or a commitment–goal pair. For each commitment and each goal, the agent selects at most one of the applicable practical rules to execute. Each selected practical rule can yield one or more possible actions; from the set of actions, the agent selects at most one action for each commitment and each goal. For example, the agent is not allowed to simultaneously select two practical rules on a commitment, one to cancel and the other to satisfy it. Each selected action corresponds to at most one transition in the life cycle of each

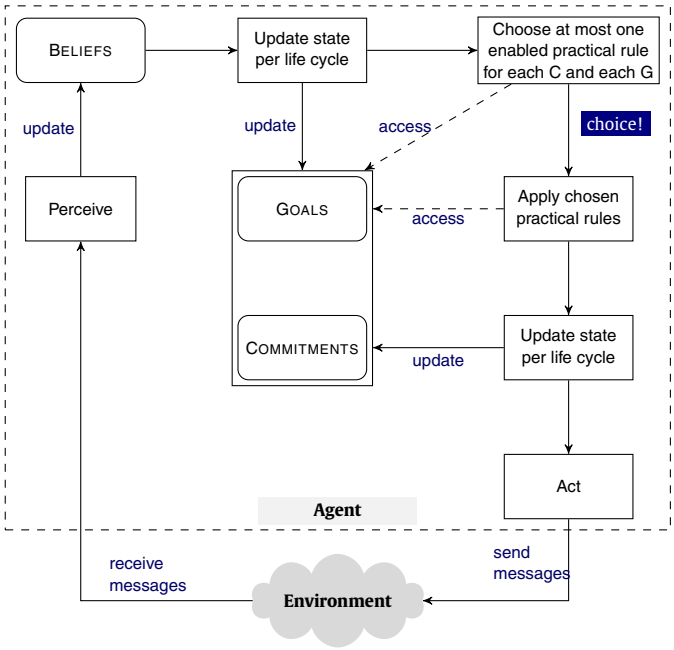


Fig. 1. Simple agent architecture and operations (adapted from Telang et al. [4]).

commitment and each goal and results in updating the state of any affected commitment or goal, i.e. the agent's beliefs about them. All such transitions are executed in parallel.

Next, after the practical rule selection and subsequent goal and commitment state updates, the agent proceeds with 'standard' BDI (Belief–Desire–Intention) deliberation about goals and intentions (or plans), such as plan selection [17–19]. This can result in goal (and plan) state updates, i.e. changes to the agent's beliefs about them, and is not shown separately in the figure.

Finally, in addition to modifying its own state, the agent acts to modify the environment by sending messages. These environmental 'actions', whether from practical rules, plan actions, or otherwise, are shown occurring together in the 'Act' box in the figure. In a message, the agent communicates its action on a commitment, or its (attempt at) bringing about an objective condition in the environment. The control loop repeats with the next perception. The success of environmental actions (and corresponding updates to agents' beliefs) is perceived at the start of the next execution cycle.

Although the agent may consider multiple actions, in each deliberation cycle the agent can *choose* at most one action (based on an enabled practical rule) for each commitment and goal. This is a convenient simplification in that we can reasonably view each agent as a locus of action, even when we decide to report the intentions or abilities of a group of agents [20,21]. We treat the agent's operations on its cognitive and social state through our practical rules. We do not treat the agent's plans or domain actions (box 'Act'). Since we do not model an agent's domain actions, we do not reason about the agent's success or failure with its goals and commitments, just about the coherence of the goals and commitments of a single agent or of a multiagent system.

3.2. Important assumptions

We make the following simplifying assumptions:

- The agents have sufficiently accurate sensors that their observations of the common reality are in agreement.
- The agents have compatible belief-world models and hold consistent beliefs. Any observations made by an agent are consistent with its belief-world model.
- The belief-world model does not change over the course of an interaction. Moreover, an agent's beliefs are temporally consistent—if an agent believes that a proposition is forever true, it forever believes that proposition.
- An agent's goals are mutually consistent.
- The creditor and debtor of a commitment agree as to its state.
- The goals progress in that an a-goal eventually reaches a terminal state and that an m-goal does not remain indefinitely active.
- Each achievement goal eventually reaches a terminal state, either positive (e.g., `Satisfied`) or negative (e.g., `Failed`); and that a maintenance goal will not remain indefinitely `Active`.

$$\begin{array}{lcl} \text{prop} & \longrightarrow & \phi \\ \phi & \longrightarrow & \phi \wedge \phi \mid \phi \vee \phi \mid \neg\phi \mid \ell \\ \ell & \longrightarrow & a \mid \neg a \end{array}$$

Fig. 2. BNF syntax of a simple propositional formula. $\ell \in \overline{\Omega}$ is a literal and $a \in \Omega$ is an atom.

$$\begin{array}{lcl} \text{world} & \longrightarrow & \mathsf{B}(x, \Box E) \\ E & \longrightarrow & E \wedge E \mid \text{Conj} \rightarrow a \mid \text{Conj} \rightarrow \Box a \mid \text{Disj} \rightarrow \neg a \mid \text{Disj} \rightarrow \Box\neg a \\ \text{Conj} & \longrightarrow & a \wedge \text{Conj} \mid a \\ \text{Disj} & \longrightarrow & a \vee \text{Disj} \mid a \end{array}$$

Fig. 3. BNF syntax of a belief-world model. The modal operators express necessity and are placed within a belief to indicate subjectivity. Here, $a \in \Omega$ is an atom.

3.3. Introducing key concepts

In the following, we recall from prior works definitions of an agent's beliefs, achievement commitments (a-comms), achievement goals (a-goals), and maintenance goals (m-goals). Then in Section 4 we define the new concept of maintenance commitments (m-comms).

We suppose a finite set of agents, $x_1, x_2, \ldots \in \mathscr{A}$, and a finite set of propositional atoms, $a_1, a_2, \ldots \in \Omega$. A literal is a positive or negated atom; beliefs are directly on atoms. We write $\overline{\Omega}$ for the set of all literals, and Ψ for the set of all propositional formulae over Ω. Fig. 2 summarises the syntax in BNF. The symbol $\top$ abbreviates $a \vee \neg a$ for any literal a, and the symbol $\bot$ abbreviates $\neg\top$. We adopt classical propositional logic. Specifically, given a set of propositions $\Phi \subseteq \Psi$ and a proposition $\psi \in \Psi$, $\Phi \models \psi$ denotes that Φ entails ψ.

Below, we adopt the notational convention where the calligraphic type (as in $\mathcal{B}$, $\mathcal{G}$, $\mathcal{C}$, and $\mathcal{M}$) refers to state functions. We consider these functions as sets of input-output pairs.

3.4. Beliefs

We first define an agent's beliefs. Beliefs are directly expressed as atoms (which we can think of as readings from sensors). However, we lift beliefs to more complex formulas based on what we term a *belief-world model*.

Definition 1 *(Belief).* A *belief* is a tuple $\langle x, a\rangle$, where $x \in \mathscr{A}$ is an agent, and $a \in \Omega$ is an atom.

We write a belief as $\mathsf{B}(x, a)$. As an example, in a lawn-care scenario, $\mathsf{B}(x, \mathsf{green\_lawn})$ is agent x's belief that green_lawn is true.

We now define an agent's belief set.

Definition 2 *(Belief set).* A *belief set* of $x \in \mathscr{A}$ is a set $\boldsymbol{B}$ of beliefs $\{\langle x, \cdot\rangle\}$.

For example, in the lawn-care scenario, $\boldsymbol{B} = \{\mathsf{B}(x, \mathsf{green\_lawn}), \mathsf{B}(x, \mathsf{hot\_temperature}), \mathsf{B}(x, \mathsf{summer})\}$ is a belief set of agent x.

Each agent has a *belief-world model* that determines what belief sets are possible for the agent. To focus on the contributions of this article, we keep the belief-world model simple. Accordingly, we express an agent's belief-world model in propositional logic augmented with one modality $\Box$, which indicates necessity—specifically, future-temporal necessity understood as *always in the future*. That is, $\Box\psi$, where $\psi \in \Psi$ is a proposition, means ψ always holds in the future. A belief-world model is subjective to an agent and interpreted within the scope of the belief modality. We limit the syntax as defined in Fig. 3 so that belief additions are unambiguous. Beliefs themselves are added to an agent's belief-world model through its *belief addition function*, which we introduce in Definition 3 below. An agent can also add beliefs through reasoning over its belief-world model, as we explain next.

Fig. 3 summarises the form of a belief-world model. Note that we admit negation only on atoms. Specifically, a belief-world model for agent x is given by a set of beliefs of the form $\mathsf{B}(x, \Box E)$, where E is an expression (and may itself be a modal expression). For example, for an agent x to believe that the lawn cannot be both green and brown at the same time, its belief-world model would include the belief that $\Box(\mathsf{green\_lawn} \rightarrow \neg\mathsf{brown\_lawn})$. That is, $\mathsf{B}(x, \Box(\mathsf{green\_lawn} \rightarrow \neg\mathsf{brown\_lawn}))$. Similarly, the contradiction between summer and winter may be expressed as $\mathsf{B}(x, \Box(\mathsf{summer} \rightarrow \neg\mathsf{winter}))$. More interestingly, for the agent to believe that a dead lawn can never become green again, its belief-world model would include the belief that $\mathsf{B}(x, \Box(\mathsf{dead\_lawn} \rightarrow \Box\neg\mathsf{green\_lawn}))$.

The belief-world model captures general facts about the world as believed by the agent, as indicated by the $\Box E$ notation. We assume for simplicity that the belief-world model does not change—that is, the same belief-world model persists from one agent configuration (see Definition 25) to the next even though the agent's beliefs may change. That is, beliefs about what always holds are temporally consistent—e.g., if an agent believes that the lawn is forever brown, it forever believes that the law is brown. We also assume that the belief-world model is consistent and that observations made by an agent are consistent with the belief-world model.

An agent x's changing beliefs refer only to literals. However, it helps to reason about beliefs regarding more complex propositional expressions. Thus, we lift beliefs to propositions in general. Reasoning with a belief-world model can proceed as below.

- $\mathsf{B}(x, \Box p) \models \mathsf{B}(x, p)$
  Whatever holds in the belief-world model as a necessity also holds as a belief of the agent.
- $\mathsf{B}(x, \Box p) \models \Box\mathsf{B}(x, \Box p)$
  The world model is persistent.
- $\mathsf{B}(x, p) \wedge \mathsf{B}(x, q) \models \mathsf{B}(x, p \wedge q)$
  Belief in two propositions entails belief in their conjunction.
- $\mathsf{B}(x, p \rightarrow q) \wedge \mathsf{B}(x, p) \models \mathsf{B}(x, q)$
  Beliefs are closed under implication.

The belief-world model is useful in two places in our approach. First, it guides an agent's belief updates: when a belief is added (with the belief addition function defined next), any of the current beliefs that conflict under the belief-world model with the belief being added are removed. Second, an agent relies on the belief-world model to determine when some relevant fact has become forever false and responds accordingly. For example, when the antecedent or consequent of a commitment becomes believed to be forever false, the agent transitions in the commitment's life cycle as appropriate.

Next, we define a belief addition function. This function adds a belief to an agent's belief set. This belief is an atom, i.e., a positive literal. The addition function uses the belief-world model to infer beliefs based on the added belief as well as any beliefs that contradict it. The function adds the inferred beliefs to the belief set and, if the belief set contains any beliefs that contradict the belief being added, the function retracts those contradictory beliefs.

Definition 3 *(Belief addition function).* The *belief addition function* is $+\colon \mathbf{B} \times \mathsf{B} \rightarrow \mathbf{B}$. We write the belief addition function as $\mathbf{B}' = \mathbf{B} + \mathsf{B}(\cdot, a)$. The function adds the new belief; retracts all contradictory beliefs; and adds all inferred beliefs: $\mathbf{B}' = \mathbf{B} \cup \{\mathsf{B}(x, a)\} \setminus \{\mathsf{B}(x, b) \mid a \models \neg b\} \cup \{\mathsf{B}(x, b) \mid a \models b\}$.

For example, let $\boldsymbol{B} = \{\mathsf{B}(x, \mathsf{green\_lawn}), \mathsf{B}(x, \mathsf{hot\_temperature}), \mathsf{B}(x, \mathsf{summer})\}$. Suppose agent x's belief-world model is $\mathsf{B}(x, \Box(\mathsf{brown\_lawn} \rightarrow \neg\mathsf{green\_lawn}))$. Now suppose agent x adds $\mathsf{B}(x, \mathsf{brown\_lawn})$ to $\boldsymbol{B}$. Then the new belief set $\boldsymbol{B}' = \boldsymbol{B} + \mathsf{B}(x, \mathsf{brown\_lawn}) = \{\mathsf{B}(x, \neg\mathsf{green\_lawn}), \mathsf{B}(x, \mathsf{brown\_lawn}), \mathsf{B}(x, \mathsf{hot\_temperature}), \mathsf{B}(x, \mathsf{summer})\}$.

Belief revision is an extensive topic of research [22]. Much complexity arises in tackling revision with respect to arbitrary propositions. Here, we consider updates only with respect to atoms, to indicate observations by an agent. The received observation is added as a belief along with whatever is entailed by it and any inconsistent propositions are removed. The syntax we use is limited so that the updates are unambiguous, meaning that a unique set of propositions is to be dropped when an observation is added.

In order to evaluate a propositional formula with respect to an agent's beliefs, we define a belief state function. Intuitively, it tells us if an agent believes a given proposition.

Definition 4 *(Belief state function).* A *belief state function* $\mathcal{B}\colon \mathscr{A} \times \Psi \rightarrow \{\top, \bot\}$ applies to agent–propositional formula pairs and returns $\top$ iff the proposition evaluates to true on the agent's beliefs $\mathbf{B}$.

As an example, with the above belief set $\boldsymbol{B}$, the belief state function $\mathcal{B}$ for agent x and proposition $\mathsf{green\_lawn} \wedge \mathsf{summer}$ will return $\top$, that is, $\mathcal{B}(x, \mathsf{green\_lawn} \wedge \mathsf{summer}) = \top$. In order to make it easier to distinguish the belief states from the tuples in a belief set, we use a subscript notation in which we write the agent as a subscript: $\mathcal{B}_x(\mathsf{green\_lawn} \wedge \mathsf{summer}) = \top$.

As an example, with the above belief set $\boldsymbol{B}$, $\mathcal{B}_x(\mathsf{green\_lawn} \wedge \mathsf{summer}) = \top$, $\mathcal{B}_x(\mathsf{green\_lawn} \wedge \mathsf{winter}) = \bot$, and $\mathcal{B}_x(\mathsf{green\_lawn} \vee \mathsf{winter}) = \top$.

3.5. Achievement commitments

So far we have defined a simple model of agents' beliefs. We now turn to recall the definition of achievement commitments; we adopt achievement commitment as defined by TSY. A *commitment* expresses a social or organizational relationship between two agents. Specifically, a commitment $C = \mathsf{C}(\textsc{debtor}, \textsc{creditor}, \mathsf{antecedent}, \mathsf{consequent})$ denotes that the DEBTOR commits to the CREDITOR for bringing about the consequent if the antecedent becomes true [23]. We write $ant(C)$ to denote the antecedent of commitment C and $cons(C)$ to denote its consequent.

Next, we formally define an achievement commitment ('a-comm' for short).

Definition 5 *(Achievement commitment).* An *achievement commitment* is a tuple consisting of two agents (its *debtor* and *creditor*, denoted $x \in \mathscr{A}$ and $y \in \mathscr{A}$, respectively), and two propositions (its *antecedent* and *consequent*, denoted $p \in \Psi$ and $q \in \Psi$, respectively), i.e., $\langle x, y, p, q\rangle$, where $x \neq y$, and $p \not\models q$ and $p \not\models \neg q$.

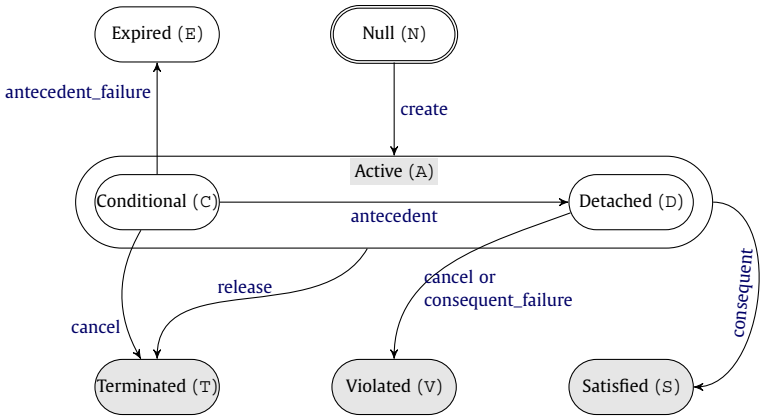


Fig. 4. Life cycle of an achievement commitment.

We write an achievement commitment as $C = \mathsf{C}(x, y, p, q)$. For example, $\mathsf{C}(\textsc{owner}, \textsc{gardener}, \mathsf{green\_lawn}, \mathsf{paid})$ is OWNER's commitment to paying GARDENER if the lawn is green.

Note that the semantics of a commitment has nothing to do with who brings about the antecedent. Three cases are important. First, the creditor may bring about the antecedent. Consider a commitment that expresses an offer, such as from Alice to Bob: if you pay me \$1, I will give you a coffee. Normally, Bob (if interested) will pay \$1, thereby making the antecedent true. Second, the antecedent may come about through the environment. For example, Alice may commit to Bob that if it rains, she will loan him her umbrella. Third, the debtor may bring about the antecedent. For example, Alice may commit to Bob that if she fails to bring a dish to a potluck lunch, she will take everyone out for drinks later. Singh [3] discusses the intuitions behind a-comms at length.

Fig. 4 shows the life cycle of a commitment adapted from TSY by removing the `Pending` states. A labelled rounded rectangle represents a commitment state, and a directed edge represents a transition. We label each transition with an agent action, an agent belief update, or some combination of the two. A double-rounded rectangle indicates an initial state. The terminal states are highlighted in grey.

The labels create and cancel are the debtor agent actions, and release is the creditor agent action. The label antecedent means the antecedent p holds, that is, p, and the label consequent means the agent believes the consequent q holds, that is, q. The label antecedent_failure means the antecedent p has failed, that is, $\Box\neg p$. Similarly, the label consequent_failure means the consequent q has failed, that is, $\Box\neg q$.

Let us here explain the use of the temporal modality. The modality $\Box$ is appropriate when a permanent condition is relevant. For example, if the antecedent of a newly created achievement commitment is false, the commitment remains in the `Conditional` state. If the antecedent becomes true, the commitment transitions to the `Detached` state. However, if in the `Conditional` state, the antecedent were to become forever false, then it would transition to the `Expired` state. We call this forever falsehood antecedent_failure. We formalize the antecedent as l and antecedent_failure as $\Box\neg l$, with the $\Box$ indicating the permanence of the negation of l. The same reasoning applies to consequent and consequent_failure. In general, when there is a choice between transitioning upon a condition, waiting, and taking the 'opposite' transition, the $\Box$ helps capture when waiting would be futile.

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 has failed, while the commitment was `Conditional`), `Satisfied` (when its consequent is brought about while the commitment was `Active`, regardless of its antecedent), `Violated` (when its antecedent has been true but its consequent has failed, or if the commitment is cancelled when `Detached`), `Terminated` (when cancelled while `Conditional` or released while `Active`). `Active` has two substates: `Conditional` (when its antecedent is neither true nor false) and `Detached` (when its antecedent has become true).

Observe that the commitment life cycle disallows some pathological transitions from the `Null` state. For example, an agent may not create a commitment if the antecedent has failed since it is useless to create such a commitment. As a result, the life cycle lacks a transition from the state `Null` to the state `Expired`. We describe such disallowed transitions:

- A debtor agent x may not create a commitment if the antecedent p has failed, that is, if $\Box\neg p$. So there is no transition from the `Null` to `Expired` state.
- A debtor agent x may not create a commitment if the consequent q has failed, that is, if $\Box\neg q$. So there is no transition from the `Null` to `Violated` state.
- A debtor agent x may not create a commitment if the consequent q holds, that is, if q. So there is no transition from the `Null` to `Satisfied` state.

Following TSY, we now define the notion of commitment strength. We employ commitment strength in defining maximally strong commitment sets, below.

Definition 6 *(A-comm strength).* A commitment $C_1 = \mathsf{C}(x, y, r, u)$ is stronger than $C_2 = \mathsf{C}(x, y, s, v)$, written $C_1 \succeq C_2$ or $C_2 \preceq C_1$, iff $s \models r$ and $u \models v$.

Next, we define a set of commitment state labels in line with Fig. 4.

Definition 7 *(A-comm states).* The *commitment states* are a set of labels $\chi_C = \{\mathtt{N}, \mathtt{C}, \mathtt{E}, \mathtt{D}, \mathtt{T}, \mathtt{V}, \mathtt{S}\}$.

Next, in order to query the state of an achievement commitment, we define a commitment state function that returns the state of a given commitment. Such state functions are useful in Section 5.

Definition 8 *(A-comm state function).* The *commitment state function* $\mathcal{C}$ returns the state of a commitment $\mathsf{C}(x, y, p, q)$, where $\mathcal{C}(\mathsf{C}(x, y, p, q)) \in \chi_C$.

To simplify the notation, we write $\mathcal{C}(\mathsf{C}(x, y, p, q))$ as $\mathcal{C}(x, y, p, q)$. We write $\mathcal{C}_x$ for the set of all non-`Null` commitments in which agent x is either debtor or creditor.

We allow an extension to TSY's achievement commitments in that we permit commitment antecedents to express the states of other commitments. This is useful in stating practical rules involving maintenance commitments (Section 6.2). For example, $\mathsf{C}(\textsc{owner}, \textsc{gardener}, \mathcal{C}(C_1) = D, \mathsf{pay})$—this commitment states that OWNER commits to GARDENER to pay if some (other) commitment is in state `D`.

Further, we allow commitment antecedents and consequents to refer to the creation of other commitments: the other commitments may be created by either the debtor or the creditor. This is useful in expressing how one commitment leads to (or depends) on the existence of other commitments, such as in the aerospace aftermarket scenario (Section 7). For example, we can state commitments such as $\mathsf{C}(\textsc{oper}, \textsc{mfr}, \mathsf{engine} \wedge \mathsf{create}(S_1), \mathsf{paid} \wedge \mathsf{create}(S_2))$. Note that typically, the antecedent may refer to commitments to be created by the creditor, and the consequent may refer to commitments to be created by the debtor.

Some of the practical rules operate over the maximally strong commitments, motivating the next definition. Intuitively, the maximally strong commitments w.r.t. a set of commitment states Σ are those commitments in any state $\sigma \in \Sigma$ for which there is no strictly stronger commitment in the same state σ.

Definition 9 *(Maximally strong commitment set).* Let $\Sigma \subseteq \chi_C$ be a set of commitment states. $maxc(\Sigma) = \{\mathsf{C}(x, y, s, v) \in \mathcal{C}_x \mid \exists \sigma \in \Sigma \text{ and } \mathcal{C}(x, y, s, v) = \sigma, \text{ and } (\forall r, u : \mathcal{C}(x, y, r, u) = \sigma, \mathsf{C}(x, y, r, u) \succeq \mathsf{C}(x, y, s, v) \Rightarrow \mathsf{C}(x, y, r, u) = \mathsf{C}(x, y, s, v))\}$.

Consider the atoms: book-provided meaning a book is provided, pen-provided meaning a pen is provided, money-paid meaning some money is paid, flight-ticket meaning a flight ticket is provided, and hotel-room meaning a hotel room is booked. Further, consider the set of commitments that use these atoms, $\{C_1, C_2, C_3, C_4\}$, where

- $\{C_1 = \mathsf{C}(x, y, \text{money-paid}, \text{book-provided} \wedge \text{pen-provided})$
- $C_2 = \mathsf{C}(x, y, \text{money-paid}, \text{book-provided})$
- $C_3 = \mathsf{C}(x, y, \top, \text{flight-ticket} \wedge \text{hotel-room})$
- $C_4 = \mathsf{C}(x, y, \top, \text{flight-ticket})\}$

Here, commitments C_1 and C_2 are in state `Conditional`, and C_3 and C_4 are in state `Detached`. Then, C_1 is a maximally strong commitment in state `Conditional`, and C_3 is a maximally strong commitment in state `Detached`, that is, $C_1 \in maxc(\mathtt{C})$ and $C_3 \in maxc(\mathtt{D})$.

Although each commitment is in a single state at any time, the $maxc()$ function finds the maximal commitments with respect to a set of states. The concept of support sets below uses $maxc()$ over a set of two states; hence note we cannot eliminate sets from Definition 9.

3.6. Achievement goals

Having recalled the definition of achievement commitments, next we recall the definition of achievement goals, again following TSY.

A *achievement goal* expresses a state of the world that an agent wishes to bring about. Specifically, a goal $G = \mathsf{G}(x, s, f)$ of an agent x has a *success condition* s that defines the success of G, and a *failure condition* f that defines its failure. A goal is successful if and only if s becomes true prior to f: that is, the truth of s entails the satisfaction of the goal only if f does not intervene. Note that s and f should be mutually exclusive. We write $succ(G)$ to denote the success condition of a goal G, i.e., $succ(G) = s$.

Next, we formally define an achievement goal ('a-goal' for short).

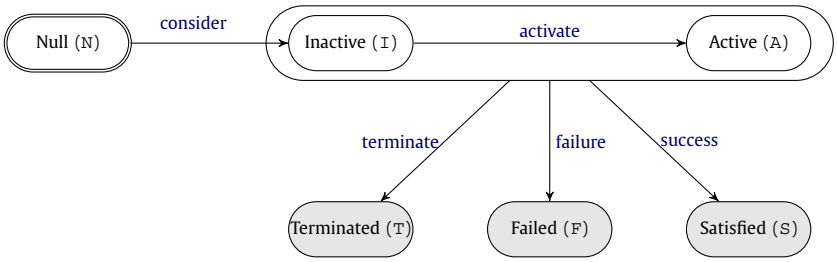


Fig. 5. Life cycle of an achievement goal.

Definition 10 *(Achievement goal).* An *achievement goal* is a tuple consisting of an agent x and two propositions: its *success* and *failure* conditions (denoted $s \in \Psi$ and $f \in \Psi$ respectively), i.e. $\langle x, s, f \rangle$, where $s \models \neg f$.

We write an achievement goal as $G = \mathsf{G}(x, s, f)$. For example, G(gardener, lawn_checked, car_broken_down) means gardener has a goal for checking the lawn with a failure condition of car breakdown.

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.

The life cycle for achievement goals is also adapted from TSY by removing the `Suspended` state. Fig. 5 depicts the resulting goal life cycle.

Similar to the commitment life cycle, a labelled rounded rectangle represents a goal state, and a directed edge represents a transition. We label each transition with an agent action or an agent belief update. A double-rounded rectangle indicates an initial state. The terminal states are highlighted in grey. The labels consider, activate, and terminate are agent actions. The label success means the success condition s is true, that is, s, and the label failure means the failure condition f is true, that is, f.

A goal can be in one of the following states: `Null`, `Inactive`, `Active`, `Satisfied`, `Terminated`, or `Failed`. The last three collectively are *terminal states*: once a goal enters any of these states, it stays there forever. Note how both commitment and goal life cycles have `Satisfied` and `Failed` states, and also have `Null` and `Active` states. Before its creation, a candidate goal is in state `Null`. Once considered by an agent (its '*goal holder*'), a goal commences as `Inactive`. Upon activation, the goal becomes `Active`; the agent may pursue its satisfaction by attempting to achieve s. If s is achieved, the goal transitions to `Satisfied`. At any point, if the failure condition of the goal becomes true, the goal transitions to `Failed`. Lastly, the agent may terminate the goal at any point,[1] thereby moving it to the `Terminated` state.

Similar to the achievement commitment strength, following TSY, we define the notion of achievement goal strength.

Definition 11 *(A-goal strength).* A goal $G_1 = \mathsf{G}(x, s, f)$ is stronger than goal $G_2 = \mathsf{G}(x, t, h)$, written $G_1 \succeq G_2$ or $G_2 \preceq G_1$, iff $s \models t$ and $f \models h$.

We now define a set of goal state labels in line with Fig. 5.

Definition 12 *(A-goal states).* The *goal states* are a set of labels: $\chi_G = \{\mathtt{N}, \mathtt{I}, \mathtt{A}, \mathtt{T}, \mathtt{F}, \mathtt{S}\}$.

Next, in order to query the state of an achievement goal, we define a goal state function that returns the state of a given goal. As with commitments, such state functions are again useful in Section 5.

Definition 13 *(A-goal state function).* The *goal state function* $\mathcal{G}$ returns the state of a goal $\langle x, s, f \rangle$, that is, $\mathcal{G}(x, s, f) \in \chi_G$.

To simplify the notation, we write $\mathcal{G}(\mathsf{G}(x, s, f))$ as $\mathcal{G}(x, s, f)$. We write $\mathcal{G}_x$ for the set of all non-`Null` a-goals of agent x.

Some of the practical rules operate over the maximally strong goals, motivating the next definition, which is akin to Definition 9: for a set of goal states Σ, the maximally strong goals are those in some $\sigma \in \Sigma$ for which there is no strictly stronger goal in the same state σ.

Definition 14 *(Maximally strong goal set).* Let $\Sigma \subseteq \chi_G$ be a set of goal states. $maxg(\Sigma) = \{\mathsf{G}(x, s, f) \in \mathcal{G}_x \mid \exists \sigma \in \Sigma$ and $\mathcal{G}(x, s, f) = \sigma$, and $(\forall t, g\colon \mathcal{G}(x, t, g) = \sigma, \mathsf{G}(x, t, g) \succeq \mathsf{G}(x, s, f) \Rightarrow \mathsf{G}(x, t, g) = \mathsf{G}(x, s, f))\}$.

For example, consider the set of goals: {G_1=G(x, book-obtained $\wedge$ pen-obtained, insufficient-money), G_2=G(x, book-obtained, insufficient-money), G_3 = G(x, book-obtained $\wedge$ pen-obtained $\wedge$glasses-obtained, insufficient-money)}. Suppose goals G_1 and G_2

[1] We combine the drop or abort transitions of Harland et al. [24].

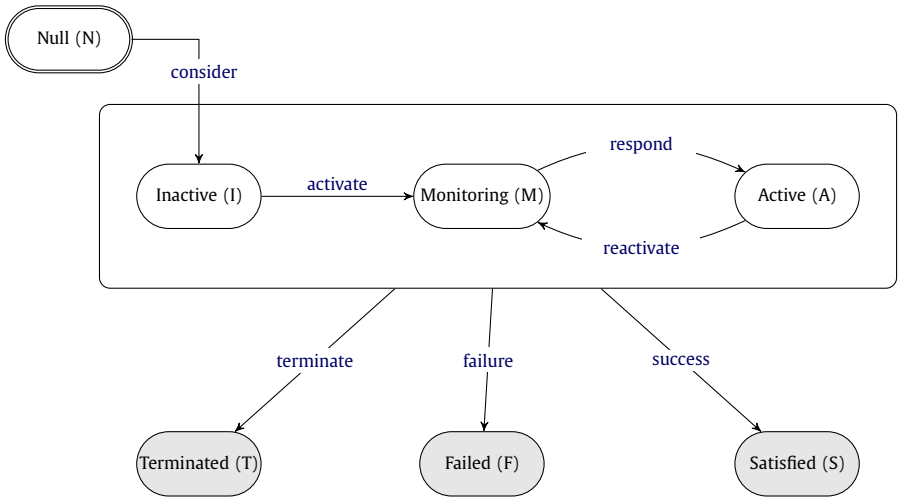


Fig. 6. Life cycle of a maintenance goal.

are `Inactive`, and goal G_3 is `Active`. Then, G_1 is a maximally strong goal in state `Inactive`, and G_3 is a maximally strong goal in state `Active`, that is, $G_1 \in maxg(\{\texttt{I}\})$ and $G_3 \in maxg(\{\texttt{A}\})$.

3.7. Maintenance goals

Having recalled the definitions of achievement commitments and achievement goals, the third and final piece to recall from the literature is that of maintenance goals. Here we follow Duff et al. [25,7] with some simplifications.

Definition 15 *(Maintenance goal, adapted from [7]).* A *maintenance goal* is a tuple $\langle x, m, s, f \rangle$, where $x \in \mathscr{A}$ is an agent, and $m, s, f \in \Psi$ are the goal's *maintenance*, *success*, and *failure* conditions, respectively, where $s \models \neg f$.

We write a maintenance goal ('m-goal' for short) as $M = \mathsf{M}(x, m, s, f)$. When m is false, x might adopt a recovery achievement goal in order to restore the truthhood of the maintenance condition.[2]

For example, $\mathsf{M}(\textsc{owner}, \mathsf{green\_lawn}, \mathsf{house\_vacated}, \mathsf{dead\_lawn})$ means OWNER has a goal to keep the lawn green with vacating the house as the success condition and dead lawn as the failure condition. If OWNER sees the lawn in turning brown, she might for instance adopt an achievement goal of hiring a gardener.

The life cycle for maintenance goals is adapted from Duff et al. by removing the state `Suspended`. Fig. 6 depicts the resulting goal life cycle. A maintenance goal can be in one of the following states: `Null`, `Inactive`, `Monitoring`, `Active`, `Terminated`, `Failed` or `Satisfied`. In terms of states, the sole difference from achievement goals is the `Monitoring` state. As identified by Duff et al. [7]: "there are three natural states for a maintenance goal: when the maintenance condition is not being monitored [i.e., `Inactive`], when the maintenance condition is being monitored but no violation ... has been detected [i.e., `Monitoring`], and when the maintenance condition has ... violated [i.e., `Active`]." Thus state `Monitoring` corresponds to when the maintenance condition m is being monitored by the agent, but currently no action needs to be taken by the agent to restore m. Unlike an achievement goal, upon activation, a maintenance goal moves to `Monitoring`. If m is violated, the goal moves to `Active`, whereupon the agent considers recovery achievement goals to re-establish the truthhood of m. When this is done, the goal moves back to `Monitoring`. The label respond corresponds to $\neg m$, and reactivate corresponds to m. Note that a maintenance goal persists until it terminates, fails, or succeeds.

As for a-comms and a-goals, we define the notion of goal strength that enables the subsequent defining of goal closure properties. Intuitively, a *maximal maintenance goal* w.r.t. a state σ is an m-goal in state σ such that no strictly stronger m-goal is in the same state σ. Below, we identify sets of maximal m-goals w.r.t. sets of states.

Definition 16 *(M-goal strength).* A maintenance goal $M_1 = \mathsf{M}(x, m, s, f)$ is stronger than maintenance goal $M_2 = \mathsf{M}(x, n, r, g)$, written $M_1 \succeq M_2$ or $M_2 \preceq M_1$, iff $m \models n$, $s \models r$, and $g \models f$.

[2] Duff et al. [7] provide the agent with reasoning mechanisms to *predict* the consequences of its actions. Thus, if the agent believes that m will become false (in the future) unless it acts appropriately, x will adopt a preventive achievement goal. We do not discuss such proactive maintenance behaviour further.

The motivation for this definition is that (1) $m \models n$: work needed to maintain the stronger goal is adequate to maintain the weaker goal; (2) $s \models r$: when the stronger goal is satisfied, so is the weaker goal; and (3) $g \models f$: when the weaker goal fails, so does the stronger goal.

As an example, let M_1 = M(x, green_lawn $\wedge$ blooming_shrubs, year_end $\wedge$ weed_free_lawn, dead_lawn), that is, agent x has a goal to maintain lawn and shrubs. M_1 satisfies when the year ends and the lawn is free of any weeds, and it fails if the lawn dies. Let M_2 = M(x, green_lawn, year_end, dead_lawn $\wedge$ dead_shrubs), that is, agent x has a goal to maintain lawn. M_2 satisfies when the year ends, and it fails if the lawn dies. Then M_1 is stronger than M_2.

We define a set of maintenance goal state labels in line with Fig. 6:

Definition 17 *(M-goal states).* The *maintenance goal states* are a set of labels: $\chi_M = \{\mathtt{N}, \mathtt{I}, \mathtt{M}, \mathtt{A}, \mathtt{T}, \mathtt{F}, \mathtt{S}\}$.

Next, in order to query the state of a maintenance goal, we introduce a maintenance goal state function. Such state functions are again useful in Section 5. The maintenance goal state function is semantic in that it specifies what goals are in what state. We can think of it as specifying the set of goals in each of the possible states. In this light, we introduce the intuition of a *closure property* to mean that the set of goals in a state must be closed with respect to the logical components of the goal. Satisfying the closure requirement ensures that the goals are not placed in logically incompatible states. The discussion of life cycles in Section 5 relies on these closure properties for the transitions on different goals to happen consistently. Later, for the same reason, we define closure properties for the maintenance commitments as well.

Definition 18 *(M-goal state function).* The *maintenance goal state function* $\mathcal{M}$ returns the state of a goal $\langle x, m, s, f \rangle$, that is, $\mathcal{M}(x, m, s, f) \in \chi_M$. The maintenance goal state function satisfies the following closure properties:

- If $\mathcal{M}(M_1) = \sigma$, where $\sigma \in \{\mathtt{A}, \mathtt{M}, \mathtt{S}\}$ and $M_1 \succeq M_2$, then $\mathcal{M}(M_2) = \sigma$.
- If $\mathcal{M}(M_1) = \sigma$, where $\sigma \in \{\mathtt{T}, \mathtt{F}\}$ and $M_2 \succeq M_1$, then $\mathcal{M}(M_2) = \sigma$.

We write $\mathcal{M}_x$ as the set of all non-`Null` m-goals of agent x.

Some of our practical rules operate over the maximal maintenance goals. This motivates the next definition: *maxm*() is for m-goals as *maxg*() is for a-goals.

Definition 19 *(Maximal m-goal set).* Let $\Sigma \subseteq \chi_M$ be a set of goal states. Then the *maximal maintenance goal set* is: $maxm(\Sigma) = \{\mathsf{M}(x, m, s, f) \in \mathcal{M}_x \mid \exists \sigma \in \Sigma$ and $\mathcal{M}(x, m, s, f) = \sigma$, and $(\forall n, r, g\colon \mathcal{M}(x, n, r, g) = \sigma, \mathsf{M}(x, n, r, g) \succeq \mathsf{M}(x, m, s, f) \Rightarrow \mathsf{M}(x, n, r, g) = \mathsf{M}(x, m, s, f))\}$.

4. Maintenance commitments

We are now ready to introduce the main topic of the present article. Informally, in a maintenance commitment, debtor x commits to creditor y that if the antecedent l holds true, then until the discharge condition d becomes true, agent x will sustain the maintenance condition m. The maintenance commitment is violated if the maintenance failure condition f becomes true. The requirement that $l \not\models d$ means that a maintenance commitment should not be discharged immediately upon being detached, and the requirement $m \not\models f$ means that the maintenance condition should not entail the maintenance failure condition.

Definition 20 *(Maintenance commitment).* A *maintenance commitment* is a tuple $\langle x, y, l, m, d, f \rangle$, where $x, y \in \mathscr{A}$ are agents, $x \neq y$, and $l, m, d, f \in \Psi$ are formulas, where $l \not\models d$ and $m \not\models f$. We call x and y *debtor* and *creditor* of this commitment and call l, m, d, and f respectively its *antecedent*, *maintenance condition*, *discharge condition*, and *maintenance failure condition*.

The symbol S stands for *sustains*. We write a maintenance commitment as $S = \mathsf{S}(x, y, l, m, d, f)$. For example, S(GARDENER, OWNER, contract_agreed, green_lawn, contract_expired, dead_lawn) means that upon the contract agreement, GARDENER commits to OWNER to maintaining the lawn as green until the contract expires. The failure condition of the commitment is a dead lawn.

Fig. 7 shows the life cycle of a maintenance commitment. Both debtor x and creditor y represent the changing states of a commitment according to this life cycle. Similar to the previous life cycles, a labelled rounded rectangle represents a commitment state, and a directed edge represents a transition. We label each transition with an agent action or an agent belief update. A double-rounded rectangle indicates an initial state. The terminal states are highlighted in grey.

The labels create and cancel are the debtor agent actions, and release is the creditor agent action. Let z be either the creditor or debtor agent of a commitment: $z \in \{x, y\}$. The label antecedent means the antecedent l holds, that is, l, and discharge means the discharge condition d holds, that is, d. The label antecedent_failure means the antecedent l has failed, that is, $\Box\neg l$. The label respond means the maintenance condition m is false, that is, $\neg m$, and sustained means the maintenance condition m is true, that is, m. The label maintenance_failure means the maintenance failure condition f is true, that is, f.

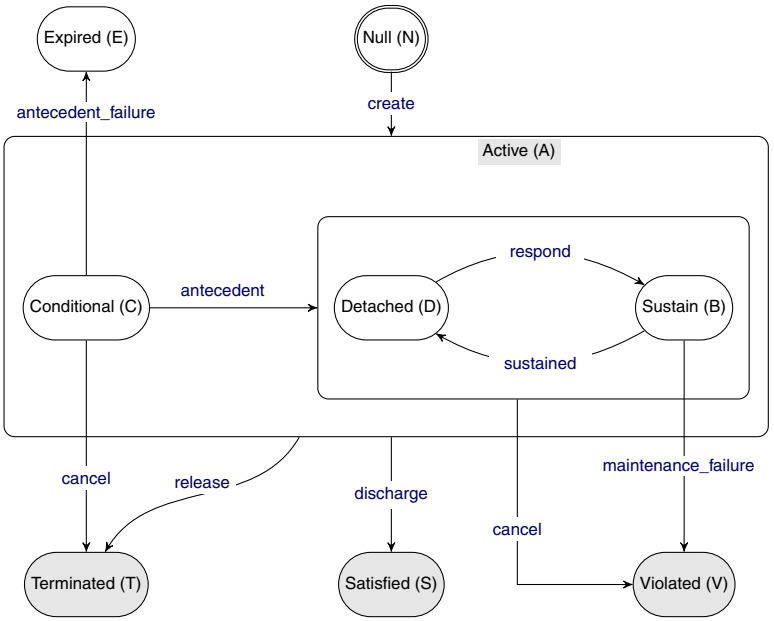


Fig. 7. Life cycle of a maintenance commitment.

A maintenance commitment can be in one of the following states: `Null` (before it is created), `Conditional` (when it is initially created), `Expired` (when its antecedent has failed, while the commitment was `Conditional`), `Detached` (when its antecedent is brought about and the maintenance condition is true), `Sustain` (when its maintenance condition becomes false when it is `Detached`), `Violated` (when its maintenance failure condition becomes true when it is in `Sustain` state or it is cancelled when it is in `Detached` or `Sustain`), `Terminated` (when cancelled while `Conditional` or released while `Active`). `Active` has three substates: `Conditional`, `Detached` and `Sustain`.[3]

As we did for goals, we now define *commitment strength*, enhancing prior definitions for achievement commitments [26,27], to enable defining the important commitment closure properties below. The motivation for this definition is that (1) $l_2 \models l_1$ (where the indices 1 and 2 refer to the stronger and weaker commitments, respectively): when the weaker commitment is detached, so is the stronger one, indicating that the stronger commitment places a demand on the debtor when the weaker one does; (2) $l_2 \wedge m_1 \models l_2 \wedge m_2$: given that both commitments are detached, work to maintain the stronger commitment is adequate to maintain the weaker one; (3) $l_2 \wedge d_1 \models l_2 \wedge d_2$: given that both commitments are detached, when the stronger commitment is discharged, so is the weaker commitment; and (4) $l_2 \wedge f_2 \models l_2 \wedge f_1$: given that both commitments are detached, when the weaker commitment fails, so does the stronger commitment. We include the antecedents in the above conditions to avoid 'false positives' in that if the weaker commitment were not detached, the prospects of maintaining or discharging it or failing at it would be moot.

Definition 21 *(M-comm strength).* Let $S_1 = \mathsf{S}(x, y, l_1, m_1, d_1, f_1)$ and $S_2 = \mathsf{S}(x, y, l_2, m_2, d_2, f_2)$ be maintenance commitments. Then S_1 is stronger than S_2, written $S_1 \succeq S_2$ or $S_2 \preceq S_1$, iff $l_2 \models l_1$, $l_2 \wedge m_1 \models l_2 \wedge m_2$, $l_2 \wedge d_1 \models l_2 \wedge d_2$, and $l_2 \wedge f_2 \models l_2 \wedge f_1$.

For example, let S_1 = S(x, y, paid, green_lawn $\wedge$ blooming_shrubs, year_end $\wedge$ weed_free_lawn, lawn_dead). In S_1, the debtor x commits to maintaining the lawn and shrubs in a healthy (green) condition if the creditor y pays for the service. The commitment successfully discharges when the year ends and the lawn is free of any weeds. The commitment fails if the lawn dies. Let S_2 = S(x, y, paid $\wedge$ fertilizer_provided, green_lawn, year_end, normal_weather $\wedge$ lawn_dead). In S_2, the debtor x commits to maintaining the lawn in a healthy (green) condition if the creditor y pays for the service. The commitment successfully discharges when the year ends. The commitment fails if the weather during the year is normal and the lawn dies. Then S_1 is stronger than S_2.

We define a maintenance commitment state labels in line with Fig. 7.

Definition 22 *(M-comm state function).* The maintenance commitment states are a set of labels: $\chi_S = \{\mathtt{N}, \mathtt{C}, \mathtt{E}, \mathtt{D}, \mathtt{B}, \mathtt{T}, \mathtt{V}, \mathtt{S}\}$.

As with the other cognitive constructs, maintenance commitments need a state function:

[3] For visual ease, the figure does not show which substate of `Active` is reached when create is called for a `Null` m-comm. Similarly for the transition from `Conditional` to the substate comprising `Detached` and `Sustain`. These are defined in Definition 29.

Definition 23 *(M-comm state function).* The *maintenance commitment state function* $\mathcal{S}$ maps each maintenance commitment to a state in χ_S. For simplicity, we write $\mathcal{S}(\mathsf{S}(x, y, l, m, d, f))$ as $\mathcal{S}(x, y, l, m, d, f)$. This function satisfies the following closure properties:

- If $\mathcal{S}(S_1) \in \{\mathtt{C}, \mathtt{S}, \mathtt{E}\}$, $S_1 \succeq S_2$, then $\mathcal{S}(S_2) = \mathcal{S}(S_1)$.
- If $\mathcal{S}(S_1) \in \{\mathtt{D}, \mathtt{B}, \mathtt{V}, \mathtt{T}\}$, $S_2 \succeq S_1$, then $\mathcal{S}(S_2) = \mathcal{S}(S_1)$.

We write $\mathcal{S}_x$ for the set of all non-`Null` m-commitments in which agent x is either debtor or creditor.

The closure properties ensure that an agent configuration is semantically viable. To understand the underlying intuition, note that a commitment may transition from one state to another due to either some logical condition becoming true or some relevant action (e.g., release). For the logical transitions, some transitions may take place in other commitments that involve the same propositions. In particular, if a commitment transitions into one of the states `Conditional`, `Satisfied`, and `Expired` (which it may do only upon some logical condition being met), then so does any weaker commitment. Likewise, since the transitions into the states `Detached` and `Sustained` may occur only due to some logical condition holding, the same argument applies for any stronger commitment. That leaves only the transition into `Violated` due to cancel and any transition into `Terminated` (due to cancel or release). For these states, it is natural to require that when a commitment enters either of these states, so must any stronger commitment. Otherwise, the weaker commitment would be immediately resurrected from a stronger commitment that was not `Violated` or `Terminated`. For example, using S_1 and S_2 as above, if S_2 is in `Detached`, then so must S_1 be, because of how their antecedents relate; and, if S_2 is released and thus `Terminated`, then so is S_1.

Intuitively, a *maximal maintenance commitment* w.r.t. a state σ is a commitment in σ such that no strictly stronger maintenance commitment is in the same state σ. We express practical rules over such commitments. Below, we identify sets of maximal commitments w.r.t. sets of states; *maxs*() is for m-comms as *maxc*() is for a-comms.

Definition 24 *(Maximal m-comm set, maxs(·)).* Let $\Sigma \subseteq \chi_S$ be a set of maintenance commitment states. Then: $maxs(\Sigma) = \{\mathsf{S}(x, y, l_1, m_1, d_1) \in \mathcal{S}_x \mid \exists \sigma \in \Sigma \text{ and } \mathcal{S}(x, y, l_1, m_1, d_1) = \sigma, \forall l_2, m_2, d_2 : \mathcal{S}(x, y, l_2, m_2, d_2) = \sigma,\ \mathsf{S}(x, y, l_1, m_1, d_1) \succeq \mathsf{S}(x, y, l_2, m_2, d_2) \Rightarrow \mathsf{S}(x, y, l_1, m_1, d_1) = \mathsf{S}(x, y, l_2, m_2, d_2)\}$.

For example, consider the set of maintenance commitments above: $\{S_1 = \mathsf{S}(x, y, \mathsf{paid}, \mathsf{green\_lawn} \wedge \mathsf{blooming\_shrubs}, \mathsf{year\_end} \wedge \mathsf{weed\_free\_lawn}, \mathsf{lawn\_dead}),\ S_2 = \mathsf{S}(x, y, \mathsf{paid} \wedge \mathsf{fertilizer\_provided}, \mathsf{green\_lawn}, \mathsf{year\_end}, \mathsf{normal\_weather} \wedge \mathsf{lawn\_dead})\}$. Suppose both commitments are in state `Conditional`. Then S_1 is a maximally strong m-comm in that state, that is $S_1 \in maxs(()\mathtt{C})$.

5. Configurations and life cycle rules

So far we have recalled the definitions of a-comms, a-goals and m-goals, and provided each with closure properties, and introduced the definition of m-comms. We now define the configuration of a multiagent system and begin to study its coherence according to the life cycle rules of commitments and goals.

5.1. Agent configuration

An agent's configuration comprises elements both of its cognitive state (i.e., beliefs and goals) and its social state (i.e., commitments of which the agent is creditor or debtor).

Definition 25. The *configuration* of an agent x is the tuple $S(x) = \langle \mathcal{B}_x, \mathcal{G}_x, \mathcal{M}_x, \mathcal{C}_x, \mathcal{S}_x \rangle$ where $\mathcal{B}_x$ is the state function for x's beliefs, $\mathcal{G}_x$ and $\mathcal{M}_x$ are state functions of achievement and maintenance goals respectively, and $\mathcal{C}_x$ and $\mathcal{S}_x$ are state functions for achievement and maintenance commitments respectively, in which agent x is either debtor or creditor.

To reduce clutter, we write the configuration of agent x with a single subscript as $\langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S} \rangle_x$ instead of $\langle \mathcal{B}_x, \mathcal{G}_x, \mathcal{M}_x, \mathcal{C}_x, \mathcal{S}_x \rangle$. Note the definition relies on the state functions introduced in the previous section, and that we use the state functions extensionally.

Following TSY, an agent's goals and commitments must be consistent with its beliefs. For example, if agent x believes in the success condition of a goal, then the goal's state must be `Null` (i.e., whereupon it is not in $\mathcal{G}_x$) or `Satisfied`. We also assume the goals are mutually consistent [28].

How goals and commitments cohere—within and across agents—is a main theme of this article, which we pick up in Section 6.

5.2. System configuration

In our model, computation in a multiagent system is realized entirely in its member agents. A goal is private to an agent. By contrast, each commitment is represented by both its creditor and its debtor. For simplicity, we assume for each

commitment that its creditor and debtor agree on its state. In this article, we elide the concerns of communication about commitments [29] and alignment [30] for simplicity.

Definition 26. The *system configuration* of a multiagent system of agents $\mathscr{A} = x_1, \ldots, x_n$ is given by n-tuple $\langle S(1), \ldots, S(n)\rangle$, where $S(i)$ is the configuration of x_i.

When required, we write the multiagent system configuration with each agent's configuration expanded to its beliefs, goals, and commitments: $\langle\langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S}\rangle_1, \langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S}\rangle_2, \ldots \langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S}\rangle_n\rangle$. A *trace*, as we will formalise in Section 8.1 is a (possibly infinite) sequence of system configurations.

The rules we introduce in the coming sections apply to each agent's internal representation separately. These rules constitute a labelled transition system, with the actions being the labels and the multiagent system configuration being the state, i.e., $S \xrightarrow{\alpha} S'$ where α is an action on a cognitive or social construct. As explained above, we do not model an agent's domain actions or plans. Thus, a single transition could potentially correspond to zero or more domain actions.

5.3. Action sets

We next define formally the life cycle of goals and commitments. For this, we need action sets for beliefs, and achievement and maintenance goals and commitments. These sets are the actions that agents could take on the respective constructs. For all agents combined, we define $\mathbb{B}$, $\mathbb{G}$, $\mathbb{M}$, $\mathbb{C}$, and $\mathbb{S}$ as the sets of all beliefs, achievement goals, maintenance goals, achievement commitments, and maintenance commitments, respectively.

Each agent can act on its own elements of the system configuration. The *belief actions* set is BACTS $= \{+\}$. The *achievement goal actions* set is GACTS $= \{$consider-G, activate-G, terminate-G$\}$. The *maintenance goal actions* set is MACTS $= \{$consider-M, activate-M, terminate-M$\}$. The *achievement commitment actions* set is CACTS $= \{$create-C, cancel-C, release-C$\}$. The *maintenance commitment actions* set is SACTS $= \{$create-S, cancel-S, release-S$\}$.

An *action instance* pairs an action and a corresponding belief, goal, or commitment. It is a specific action on a specific construct. For example, the action instance $\langle$activate-G, $G_1\rangle$ corresponds to the action of activating goal G_1. Valid action instances are consistent across the components; where an action concerns a goal or commitment condition such as a consequent, it must be consistent with changes to the agent's beliefs, and actions corresponding to that belief change must occur on all goals and commitments. When a goal or commitment is affected, so are weaker or stronger goals and commitments to preserve consistency and closure properties, e.g., as in Definition 16.

Formally, an *action set* is a set of concurrent action instances of the same agent:

Definition 27 *(Action sets).* The *action set* for each of the goal and commitment types is disjoint union of sets of pairs:

- *Belief action set*: $\mathbb{A}_{\mathsf{B}} = (\text{BACTS} \times \mathbb{B})$
- *Achievement goal action set*: $\mathbb{A}_{\mathsf{G}} = (\text{GACTS} \times \mathbb{G})$
- *Maintenance goal action set*: $\mathbb{A}_{\mathsf{M}} = (\text{MACTS} \times \mathbb{M})$
- *Achievement commitment action set*: $\mathbb{A}_{\mathsf{C}} = (\text{CACTS} \times \mathbb{C})$
- *Maintenance commitment action set*: $\mathbb{A}_{\mathsf{S}} = (\text{SACTS} \times \mathbb{S})$

5.4. Life cycle rules

We can now define a *life cycle* rule as a mapping from a system configuration and an action set into a resulting system configuration. The life cycle rule definition for achievement constructs is inherited from TSY, except that we remove any parts of a rule relating to `Pending` and `Suspended` states. We do not repeat those definitions here. The life cycles of maintenance goals and commitments capture Figs. 6 and 7 in logical terms.

The life cycle rule definition for maintenance constructs is in four parts for ease of reading. The first part of the definition specifies the maintenance goal transitions that occur due to belief updates.

Definition 28 *(Maintenance goal life cycle rule—belief update).* A *life cycle* rule for a maintenance goal is a function $L_{\mathsf{M}}\colon \mathbb{A}_{\mathsf{B}} \times \mathbb{B} \times \mathbb{M} \to \mathbb{B} \times \mathbb{M}$ such that: $\forall\langle +, b\rangle \in \mathbb{A}_{\mathsf{B}}$, $b = \langle x, p\rangle$:
$\langle \boldsymbol{B}', \mathcal{M}'\rangle = L_{\mathsf{M}}(\langle +, b\rangle, \boldsymbol{B}, \mathcal{M})$, where:

1. $\boldsymbol{B}' = \boldsymbol{B} + \langle x, p\rangle$ (x believes the newly added atom p)
2. if $\mathcal{M}(x, m, s, f) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, \mathcal{B}'_x(s) = \top$, then $\mathcal{M}'(x, m, s, f) = \mathtt{S}$
   (if x believes s, each maintenance goal $\mathsf{M}(x, m, s, f)$ that is `Inactive`, `Active`, `Monitoring`, succeeds)
3. if $\mathcal{M}(x, m, s, f) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, \mathcal{B}'_x(f) = \top$, then $\mathcal{M}'(x, m, s, f) = \mathtt{F}$
   (if x believes f, each maintenance goal $\mathsf{M}(x, m, s, f)$ that is `Inactive`, `Active`, `Monitoring`, fails)
4. if $\mathcal{M}(x, m, s, f) \in \{\mathtt{M}\}, \mathcal{B}'_x(\neg m \wedge \neg s \wedge \neg f) = \top$, then $\mathcal{M}'(x, m, s, f) = \mathtt{A}$
   (if x believes $\neg m \wedge \neg s \wedge \neg f$, each maintenance goal $\mathsf{M}(x, m, s, f)$ that is `Monitoring`, becomes `Active`)

5. if $\mathcal{M}(x,m,s,f) \in \{\texttt{A}\}, \mathcal{B}'_x(m \wedge \neg s \wedge \neg f) = \top$, then $\mathcal{M}'(x,m,s,f) = \texttt{M}$
   (if x believes $m \wedge \neg s \wedge \neg f$, each maintenance goal $\mathsf{M}(x,m,s,f)$ that is `Active`, transitions to state `Monitoring`)
6. if $\mathcal{M}(x,m,s,f) \in \{\texttt{T},\texttt{F},\texttt{S}\}$, then $\mathcal{M}'(x,m,s,f) = \mathcal{M}(x,m,s,f)$
   (maintenance goals that are `Terminated`, `Failed` or `Satisfied` remain unaffected)

The next definition specifies the maintenance goal transitions that occur due to the social actions that the agents execute.

Definition 29 *(Maintenance goal life cycle rule—social actions).* A *life cycle* rule is a function $L_{\mathsf{M}}\colon \mathbb{A}_{\mathsf{M}} \times \mathbb{B} \times \mathbb{M} \to \mathbb{B} \times \mathbb{M}$ such that: $\forall \langle mact, M\rangle \in \mathbb{A}_{\mathsf{M}}, mact \in MACTS, M = \mathcal{M}(x,m,s,f)$:
$\langle \boldsymbol{B}', \mathcal{M}'\rangle = L_{\mathsf{M}}(\langle mact, M\rangle, \boldsymbol{B}, \mathcal{M})$, where:

1. $\boldsymbol{B}' = \boldsymbol{B}$ (all beliefs are unaffected)
2. if $mact = \mathsf{consider}$ and $\mathcal{M}(x,m,s,f) = \texttt{N}$, then $\mathcal{M}'(x,m,s,f) = \texttt{I}$
   (if agent x considers a maintenance goal, the goal transitions from `Null` to `Inactive`)
3. if $mact = \mathsf{activate}$ and $\mathcal{M}(x,m,s,f) = \texttt{I}$, then $\mathcal{M}'(x,m,s,f) = \texttt{M}$
   (if agent x activates an `Inactive` maintenance goal, the goal transitions to `Monitoring`)
4. if $mact = \mathsf{terminate}$ and $\mathcal{M}(x,s,u) \in \{\texttt{I},\texttt{M},\texttt{A}\}$, then $\mathcal{M}'(x,m,s,f) = \texttt{T}$
   (if agent x terminates a `Inactive`, `Active`, `Monitoring`, maintenance goal, the goal transitions to `Terminated`)
5. if $\mathsf{M}(x,n,t,h) \not\trianglelefteq \mathsf{M}(x,m,s,f)$ and $\mathsf{M}(x,n,t,h) \not\trianglerighteq \mathsf{M}(x,m,s,f)$, then $\mathcal{M}'(x,n,t,h) = \mathcal{M}(x,n,t,h)$
   (maintenance goals that are unrelated to $\mathcal{M}(x,m,s,f)$ remain unaffected)

We now define the life cycle rule for maintenance commitments that considers only the belief updates.

Definition 30 *(Maintenance commitment life cycle rule—belief update).* A *life cycle* rule is a function $L_{\mathsf{S}}\colon \mathbb{A}_{\mathsf{B}} \times \mathbb{B} \times \mathbb{S} \to \mathbb{B} \times \mathbb{S}$ such that: $\forall \langle +, b\rangle \in \mathbb{A}$, $b = \langle x, p\rangle$:
$\langle \boldsymbol{B}', \mathcal{S}'\rangle = L_{\mathsf{S}}(+p, \boldsymbol{B}, \mathcal{S})$, where:

1. $\boldsymbol{B}' = \boldsymbol{B} + \langle x, p\rangle$ (x believes the newly added atom p)
2. if $\mathcal{S}(x,y,l,m,d,f) = \texttt{C}, \mathcal{B}'_x(l) = \top, \mathcal{B}'_x(d) = \bot$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{D}$
   (if x believes l and does not believe d, each maintenance commitment $\mathsf{S}(x,y,l,m,d,f)$ that is `Conditional`, detaches)
3. if $\mathcal{S}(x,y,l,m,d,f) \in \{\texttt{C},\texttt{D},\texttt{B}\}, \mathcal{B}'_x(d) = \top$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{S}$
   (if x believes d, each maintenance commitment $\mathsf{S}(x,y,l,m,d,f)$ that is `Conditional`, `Detached`, or `Sustain`, satisfies)
4. if $\mathcal{S}(x,y,l,m,d,f) = \texttt{C}, \mathcal{B}'_x(\neg l) = \top, \mathcal{B}'_x(d) = \bot$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{E}$
   (if x believes $\neg l$ and does not believe d, each maintenance commitment $\mathsf{S}(x,y,l,m,d,f)$ that is `Conditional`, expires)
5. if $\mathcal{S}(x,y,l,m,d,f) = \texttt{B}, \mathcal{B}'_x(f) = \top$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{V}$
   (if x believes f, each maintenance commitment $\mathsf{S}(x,y,l,m,d,f)$ that is in state `Sustain`, violates)
6. if $\mathcal{S}(x,y,l,m,d,f) = \texttt{D}, \mathcal{B}'_x(\neg m) = \top, \mathcal{B}'_x(d) = \bot$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{B}$
   (if x believes $\neg m$ and does not believe d, each maintenance commitment $\mathsf{S}(x,y,l,m,d,f)$ that is in state `Detached`, transitions to `Sustain`)
7. if $\mathcal{S}(x,y,l,m,d,f) = \texttt{B}, \mathcal{B}'_x(m) = \top, \mathcal{B}'_x(d) = \bot$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{D}$
   (if x believes m and does not believe d, each maintenance commitment $\mathsf{S}(x,y,l,m,d,f)$ that is in state `Sustain`, transitions to `Detached`)
8. if $\mathcal{S}(x,y,l,m,d,f) \in \{\texttt{T},\texttt{V},\texttt{S}\}$, then $\mathcal{S}'(x,y,l,m,d,f) = \mathcal{S}(x,y,l,m,d,f)$
   (all commitments $\mathsf{S}(x,y,l,m,d,f)$ that are `Terminated`, `Violated` or `Satisfied` remain unaffected)

Next, we define the life cycle rule for maintenance commitments that considers the social actions of the debtor and creditor.

Definition 31 *(Maintenance commitment life cycle rule—social actions).* A *life cycle* rule is a function $L_{\mathsf{S}}\colon \mathbb{A}_{\mathsf{S}} \times \mathbb{B} \times \mathbb{S} \to \mathbb{B} \times \mathbb{S}$ such that: $\forall \langle sact, S\rangle \in \mathbb{A}_{\mathsf{S}}, sact \in SACTS, S = \mathcal{S}(x,y,l,m,d,f)$:
$\langle \boldsymbol{B}', \mathcal{S}'\rangle = L_{\mathsf{S}}(\langle sact, S\rangle, \boldsymbol{B}, \mathcal{S})$, where:

1. $\boldsymbol{B}' = \boldsymbol{B}$ (all beliefs are unaffected)
2. if $sact = \mathsf{create\text{-}S}$ and $\mathcal{S}(x,y,l,m,d,f) = \texttt{N}$ and $\mathcal{B}_x(l) = \bot$, then $\mathcal{S}'(x,y,l,m,d,f) = \texttt{C}$
   (if agent x creates a maintenance commitment $\mathsf{S}(x,y,l,m,d)$ and does not believe l, the commitment transitions from `Null` to `Conditional`)

Table 1
Possible interactions between components. TSY = achievement-only case: not the topic of this article. Closure = follows from closure properties such as Definition 24. Practical = treated in this article in Section 6.2. N/A = not applicable.

↓ FROM	TO a-goal	m-goal	a-comm	m-comm
a-goal	TSY	N/A	TSY	practical
m-goal	practical	closure	N/A	N/A
a-comm	TSY	N/A	TSY	N/A
m-comm	practical	practical	N/A	closure

3. if $sact = \text{create-S}$ and $\mathcal{S}(x, y, l, m, d, f) = \texttt{N}$ and $\mathcal{B}_x(l) = \top$, then $\mathcal{S}'(x, y, l, m, d, f) = \texttt{D}$
   (if agent x creates a maintenance commitment $\mathsf{S}(x, y, l, m, d, f)$ and believes l, the commitment transitions from `Null` to `Detached`)
4. if $sact = \text{cancel-S}$ and $\mathcal{S}(x, y, l, m, d, f) = \texttt{C}$, then $\mathcal{S}'(x, y, l, m, d, f) = \texttt{T}$
   (if agent x cancels a `Conditional` maintenance commitment $\mathsf{S}(x, y, l, m, d, f)$, then the commitment transitions to `Terminated`)
5. if $sact = \text{cancel-S}$ and $\mathcal{S}(x, y, l, m, d, f) = \texttt{B}$, then $\mathcal{S}'(x, y, l, m, d, f) = \texttt{V}$
   (if agent x cancels a `Sustained` maintenance commitment $\mathsf{S}(x, y, l, m, d, f)$, then the commitment transitions to `Violated`)
6. if $sact = \text{release-S}$ and $\mathcal{S}(x, y, l, m, d, f) \in \{\texttt{C}, \texttt{D}, \texttt{B}\}$, then $\mathcal{S}'(x, y, l, m, d, f) = \texttt{T}$
   (if agent y releases a `Conditional`, `Detached`, or `Sustained` maintenance commitment, then the commitment transitions to `Terminated`)
7. if $\mathsf{S}(x, y, k, n, e, g) \not\trianglerighteq \mathsf{S}(x, y, l, m, d, f)$ and $\mathsf{S}(x, y, k, n, e, g) \not\trianglelefteq \mathsf{S}(x, y, l, m, d, f)$, then $\mathcal{S}'(x, y, k, n, e, g) = \mathcal{S}(x, y, k, n, e, g)$
   (all maintenance commitments unrelated to $\mathsf{S}(x, y, l, m, d, f)$ remain unaffected)

6. Relating commitments and goals

In the previous section we established the life cycle rules which maintenance goals and commitments must follow. We now turn to the 'optional' practical rules at the heart of our semantics. Recall that, in contrast to life cycle rules, practical rules capture patterns of pragmatic reasoning that agents may or may not adopt under different circumstances.

An agent's practical rules thus reflect its decision-making. Practical rules capture an agent's rational behaviour, for example: an agent would adopt commitments to help achieve or maintain its end goals and given its commitments, would create means goals to satisfy or sustain them. We provide one set of practical rules, noting that our methodology is generic in allowing other sets, although the theorems would need to be modified. Telang et al. [4] discuss the advantages of this ruleset-agnostic methodology.

To organize the practical rules, we note that beliefs do not directly give rise to actions. Therefore, we consider direct interactions between achievement and maintenance goals and commitments, giving rise to the $4^2 = 16$ possibilities in Table 1.

6.1. Support sets

Fig. 8 captures the relationships of a maintenance commitment or goal as pairs of functions. These functions help us define practical rules unambiguously; in this subsection we detail the functions. Let S, G, M respectively be a maintenance commitment, achievement goal, and maintenance goal. Then, for instance, $\text{GAS}(S)$ identifies achievement goals such that S's antecedent entails the success condition of the goals. The goals created by the creditor to detach S are in $\text{GAS}(S)$. For each of these functions, we define an 'inverse' as a function in the reverse direction.

An achievement goal provides antecedent support to a maintenance commitment if the success condition of that goal entails its antecedent. We now define a set of such achievement goals. Intuitively, GAS is a set of achievement goals which together are sufficient to support a given m-comm's antecedent.

Definition 32. (Set of achievement goals providing antecedent support to a maintenance commitment (GAS)) Let $S = \mathsf{S}(y, x, k, \cdot, \cdot, \cdot)$ and $G = \mathsf{G}(x, s, \cdot)$. $\text{GAS}(S) = \{G \mid S \in \mathit{maxs}(\{\texttt{C}, \texttt{D}, \texttt{B}, \texttt{P}\}), \mathcal{G}(G) \in \{\texttt{I}, \texttt{A}\}, k = \bigwedge k_i, s \models k_i\}$.

For example, consider the m-comm $S = \mathsf{S}(y, x, \text{invoice} \wedge \text{paid}, \cdot, \cdot)$, and a-goals $G_1 = \mathsf{G}(x, \text{invoice}, \cdot)$ and $G_2 = \mathsf{G}(x, \text{paid}, \cdot)$. Then, $G_1 \in GAS(S)$ and $G_2 \in GAS(S)$, if $\mathcal{G}(G_1) \in \{\texttt{I}, \texttt{A}\}$ and $\mathcal{G}(G_2) \in \{\texttt{I}, \texttt{A}\}$.

The inverse of GAS is a set of maintenance commitments whose antecedent is supported by an achievement goal.

Definition 33. (Set of maintenance commitments whose antecedent is supported by an achievement goal (GAS^{-1})) Let $G = \mathsf{G}(x, s, \cdot)$ and $S = \mathsf{S}(y, x, k, \cdot, \cdot, \cdot)$. $\text{GAS}^{-1}(G) = \{S \mid S \in \mathit{maxs}(\{\texttt{C}, \texttt{D}, \texttt{B}\}), \mathcal{G}(G) \in \{\texttt{I}, \texttt{A}\}, k = \bigwedge k_i, s \models k_i\}$.

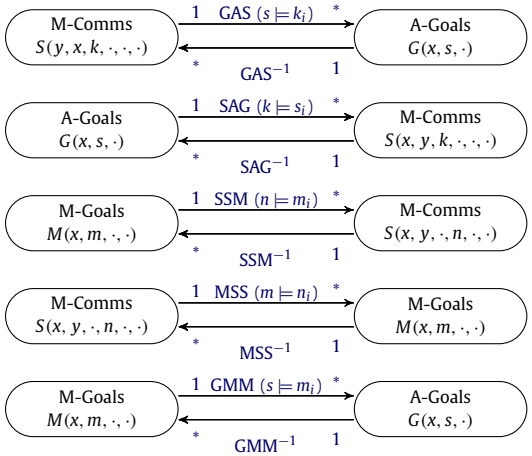


Fig. 8. Functions relating (s-)comms and (m-)goals.

For example, consider the a-goal $G = \mathsf{G}(x, \mathsf{invoice} \wedge \mathsf{paid}, \cdot)$, and s-commits $S_1 = \mathsf{S}(y, x, \mathsf{invoice}, \cdot, \cdot)$ and $S_2 = \mathsf{S}(y, x, \mathsf{paid}, \cdot, \cdot)$. Then $S_1 \in GAS^{-1}(G)$ and $S_2 \in GAS^{-1}(G)$, if $\mathcal{S}(S_1) \in \{\mathtt{C}, \mathtt{D}, \mathtt{B}, \mathtt{P}\}$ and $\mathcal{S}(S_2) \in \{\mathtt{C}, \mathtt{D}, \mathtt{B}\}$.

In the same manner as the set GAS and its inverse, we have four more sets to define. The motivation is that an agent can make commitments that lead, for example, to the satisfaction of one of its achievement goals.

A maintenance commitment provides antecedent support to an achievement goal if the maintenance commitment's antecedent (either partially or fully) entails the success condition of the achievement goal. We define set of such maintenance commitments.

Definition 34. (Set of maintenance commitments whose antecedent support an achievement goal (SAG)) Let $S = \mathsf{S}(x, y, k, \cdot, \cdot, \cdot)$ and $G = \mathsf{G}(x, s, \cdot)$. $\mathrm{SAG}(G) = \{S \mid S \in \mathit{maxs}(\{\mathtt{C}, \mathtt{D}, \mathtt{B}, \mathtt{P}\}), \mathcal{G}(G) \in \{\mathtt{I}, \mathtt{A}\}, s = \bigwedge s_i, k \models s_i\}$.

The inverse of SAG is a set of achievement goals that are supported by the antecedent of a maintenance commitment.

Definition 35. (Set of achievement goals that are supported by the antecedent of a maintenance commitment (SAG^{-1})) Let $S = \mathsf{S}(x, y, k, \cdot, \cdot, \cdot)$ and $G = \mathsf{G}(x, s, \cdot)$. $\mathrm{SAG}^{-1}(S) = \{G \mid S \in \mathit{maxs}(\{\mathtt{C}, \mathtt{D}, \mathtt{B}, \mathtt{P}\}), \mathcal{G}(G) \in \{\mathtt{I}, \mathtt{A}\}, s = \bigwedge s_i, k \models s_i\}$.

A maintenance goal supports a maintenance commitment if: (1) the goal's maintenance condition (either partially or fully) entails the maintenance condition of the maintenance commitment, and (2) the maintenance commitment's discharge condition entails the maintenance goal's success condition. We now define a set of such maintenance goals.

Definition 36. (Set of maintenance goals supporting a maintenance commitment (MSS)) Let $S = \mathsf{S}(x, y, \cdot, n, \cdot, \cdot)$ and $M = \mathsf{M}(x, m, \cdot, \cdot)$. $\mathrm{MSS}(S) = \{M \mid S \in \mathit{maxs}(\{\mathtt{C}, \mathtt{D}, \mathtt{B}\}), \mathcal{M}(M) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, n = \bigwedge n_i, m \models n_i, \mathit{dis}(S) \models \mathit{succ}(M)\}$.

The inverse of MSS is a set of maintenance commitments supported by a maintenance goal.

Definition 37. (Set of maintenance commitments supported by a maintenance goal (MSS^{-1})) Let $S = \mathsf{S}(x, y, \cdot, n, \cdot, \cdot)$ and $M = \mathsf{M}(x, m, \cdot, \cdot)$. $\mathrm{MSS}^{-1}(M) = \{S \mid S \in \mathit{maxs}(\{\mathtt{C}, \mathtt{D}, \mathtt{B}\}), \mathcal{M}(M) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, n = \bigwedge n_i, m \models n_i\}$.

A maintenance commitment supports a maintenance goal if its maintenance condition (either partially or fully) entails the maintenance condition of the maintenance goal. We define set of such maintenance commitments.

Definition 38. (Set of maintenance commitments supporting a maintenance goal (SSM)) Let $S = \mathsf{S}(y, x, \cdot, n, \cdot, \cdot)$ and $M = \mathsf{M}(x, m, \cdot, \cdot)$. Then $\mathrm{SSM}(M) = \{S \mid S \in \mathit{maxs}(\{\mathtt{C}, \mathtt{D}, \mathtt{B}\}), \mathcal{M}(M) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, m = \bigwedge m_i, n \models m_i\}$.

The inverse of SSM is a set of maintenance goals supposed by a maintenance commitment.

Definition 39. (Set of maintenance goals supported by a maintenance commitment (SSM^{-1})) Let $S = \mathsf{S}(y, x, \cdot, n, \cdot, \cdot)$ and $M = \mathsf{M}(x, m, \cdot, \cdot)$. Then $\mathrm{SSM}^{-1}(S) = \{M \mid S \in \mathit{maxs}(\{\mathtt{C}, \mathtt{D}, \mathtt{B}\}), \mathcal{M}(M) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, m = \bigwedge m_i, n \models m_i\}$.

An achievement goal supports a maintenance goal if its achievement condition (either partially or fully) entails the maintenance condition of the maintenance goal. We define set of such achievement goals.

Definition 40. (Set of achievement goals supporting a maintenance goal (GMM)) Let $G = \mathsf{G}(x, s, \cdot)$ and $M = \mathsf{M}(x, m, \cdot, \cdot)$. $\text{GMM}(M) = \{G \mid G \in maxg(\{\mathtt{I}, \mathtt{A}\}), \mathcal{M}(M) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, m = \bigwedge m_i, s \models m_i\}$.

The inverse of GMM is the set of maintenance goals supported by an achievement goal. This completes the set of support functions.

Definition 41. (Set of maintenance goals supported by an achievement goal (GMM^{-1})) Let $G = \mathsf{G}(x, s, \cdot)$ and $M = \mathsf{M}(x, m, \cdot, \cdot)$. $\text{GMM}^{-1}(G) = \{M \mid G \in maxg(\{\mathtt{I}, \mathtt{A}\}), \mathcal{M}(M) \in \{\mathtt{I}, \mathtt{A}, \mathtt{M}\}, m = \bigwedge m_i, s \models m_i\}$.

Altogether, these six sets and their inverses equip us with what we need to define practical rules in the sequel.

6.2. Practical rules

Practical rules represent patterns of reasoning that specify potential social actions for an agent based on its goals and commitments. We use TSY's syntax of a practical rule template of the form $E \xrightarrow{\text{RULENAME}} \alpha$. The expression E is a conjunction of the form of: this goal is (or is not) in some state and that commitment is (or is not) in some state; E concerns commitment and goal sets computed by the functions of Fig. 8 and their states. The expression α is a commitment (or goal) action to be performed on one or more commitments (or goals). We write $ant(\cdot)$ for the antecedent of a (m-) commitment, $maint(\cdot)$ for the maintenance condition of m-comm or m-goal, and $succ(\cdot)$ for the success condition of a (m-) goal.

6.2.1. A-goal to m-comm

We first describe the rules in which one or more m-commitments support an a-goal. With rule S-CREATE, an agent can create m-comms to satisfy an a-goal. With S-TERMINATE, such m-comms supporting no a-goal are terminated.

- S-CREATE: Suppose agent x has an active achievement goal $G = \mathsf{G}(x, ., .)$. Then create one or more maintenance commitments that can satisfy the goal G. Let $\omega = \bigwedge_i ant(S_i)$, where $S_i = \mathsf{S}(x, y, ., ., ., .)$ and $S_i \in \text{SAG}(G)$, and Φ be a set of commitments such that $\bigwedge_j ant(S_j) \wedge \omega \models succ(G)$ and $S_j \in \Phi$.

$$\mathcal{G}(G) = \mathtt{A} \wedge \omega \not\models succ(G) \xrightarrow{\text{S-CREATE}} \text{create}(\Phi)$$

- S-TERMINATE: Suppose a goal $G = \mathsf{G}(x, ., .)$ fails or is terminated. Then cancel each maintenance commitment supporting the goal that is not supporting some other goal.

$$\mathcal{G}(G) \in \{\mathtt{F}, \mathtt{T}\} \wedge S \in \text{SAG}(G) \wedge \text{SAG}^{-1}(S) \setminus G = \emptyset \xrightarrow{\text{S-TERMINATE}} \text{terminate}(S)$$

6.2.2. M-goal to a-goal

We describe the rules in which one or more achievement goals support a maintenance goal. With A-CONSIDER, an agent considers a-goals to restore an m-goal. With A-TERMINATE, such a-goals supporting no m-goal are terminated.

- A-CONSIDER: Suppose an m-goal M is in the active state, that is $maint(M)$ is false. Then consider one or more a-goals to restore $maint(M)$ to true. Let $\omega = \bigwedge_i succ(G_i)$, where $G_i \in \text{GMM}(M)$, and $\Phi = \{G_j\}$ is a set of new goals such that $\bigwedge_j succ(G_j) \wedge \omega \models maint(M)$.

$$\mathcal{M}(M) = \mathtt{A} \wedge \omega \not\models maint(M) \xrightarrow{\text{A-CONSIDER}} \text{consider}(\Phi)$$

- A-TERMINATE: Suppose an m-goal M fails or is terminated. Then terminate each achievement goal G supporting M that is not supporting some other m-goal.

$$\mathcal{M}(M) \in \{\mathtt{F}, \mathtt{T}\} \wedge G \in \text{GMM}(M) \wedge \text{GMM}^{-1}(G) \setminus M = \emptyset \xrightarrow{\text{A-TERMINATE}} \text{terminate}(G)$$

6.2.3. M-comm to m-goal

We describe the rules in which one or more maintenance goals support a maintenance commitment. With M-CONSIDER, an agent considers m-goals to maintain an m-comm. With M-TERMINATE, such m-goals supporting no m-comm are terminated.

- M-CONSIDER: Suppose an m-comm S is in the detached state. Then the debtor of S considers one or more m-goals to maintain the condition $maint(S)$. Let $\omega = \bigwedge_i maint(M_i)$, where $M_i \in \text{MSS}(S)$, and $\Phi = \{M_j\}$ is a set of new goals such that $\bigwedge_j maint(M_j) \wedge \omega \models maint(S)$.

$$\mathcal{S}(S) = \text{D} \wedge \omega \not\models maint(S) \xrightarrow{\text{M-CONSIDER}} \text{consider}(\Phi)$$

- M-TERMINATE: Suppose an m-comm S is expired or terminated. Then for both debtor and creditor, terminate each m-goal supporting S that is not supporting some other m-comm S'.

$$\mathcal{S}(S) \in \{\text{E}, \text{T}\} \wedge M \in \text{MSS}(S) \wedge \text{MSS}^{-1}(M) \setminus G = \emptyset \xrightarrow{\text{M-TERMINATE}} \text{terminate}(M)$$

6.2.4. M-goal to m-comm

We describe the rules in which one or more maintenance commitments support a maintenance goal. With C-CREATE, an agent creates a-comms to have another agent maintain an m-goal. Note that C-CREATE is not reducing an m-goal to a-goals; rather it creates one or more achievement commitments in order to get some other agent to maintain a condition. With MS-TERMINATE, supporting m-comms supporting no m-goal are terminated.

- C-CREATE: Suppose an m-goal $M = \text{M}(x, m, ., .)$ is in the monitoring state. Then create one or more commitments $C_j = \text{C}(x, y_j, \mathcal{S}(y_j, x, \top, m_j, \cdot, \cdot) = \text{D}, q_j)$ to persuade agent y_j to maintain the condition m_j. Note that the antecedent of C_j is a condition that the m-comm $S(y_j, x, \top, m_j, \cdot, \cdot)$ is detached. Thus this enhancement conforms to the structure of an a-comm (TSY). Let $\omega = \bigwedge_i maint(S_i)$, where $S_i \in \text{SSM}(M)$, and $\Phi = \{C_j\}$ is a set of new commitments such that $\bigwedge_j m_j \wedge \omega \models m$.

$$\mathcal{M}(M) = \text{M} \wedge \omega \not\models maint(M) \xrightarrow{\text{C-CREATE}} \text{create}(\Phi)$$

- MS-TERMINATE: Suppose an m-goal M fails or is terminated. Then cancel each m-comm supporting the goal M that is not supporting some other m-goal.

$$\mathcal{M}(M) \in \{\text{F}, \text{T}\} \wedge S \in \text{SSM}(M) \wedge \text{SSM}^{-1}(S) \setminus M = \emptyset \xrightarrow{\text{MS-TERMINATE}} \text{terminate}(S)$$

6.2.5. M-comm to a-goal

We describe the rules in which one or more a-goals support a maintenance commitment. With AD-CONSIDER, an agent considers a-goals to detach an m-comm. With AD-TERMINATE, such a-goals supporting no m-comm are terminated.

- AD-CONSIDER: Suppose an m-comm S is in the conditional state. Then the debtor of S considers one or more a-goals to detach S. Let $\omega = \bigwedge_i succ(G_i)$, where $G_i \in \text{GAS}(S)$, and $\Phi = \{G_j\}$ is a set of new goals such that $\bigwedge_j succ(G_j) \wedge \omega \models ant(S)$.

$$\mathcal{S}(S) = \text{C} \wedge \omega \not\models ant(S) \xrightarrow{\text{AD-CONSIDER}} \text{consider}(\Phi)$$

- AD-TERMINATE: Suppose an m-comm S is expired or terminated. Then for both debtor and creditor, terminate each a-goal supporting S that is not supporting some other m-comm S'.

$$\mathcal{S}(S) \in \{\text{E}, \text{T}\} \wedge G \in \text{GAS}(S) \wedge \text{GAS}^{-1}(G) \setminus S = \emptyset \xrightarrow{\text{AD-TERMINATE}} \text{terminate}(G)$$

In addition to the above practical rules, there are three practical rules which activate an inactive goal: A-ACTIVATE, M-ACTIVATE, and AD-ACTIVATE, which correspond to A-CONSIDER, M-CONSIDER, and AD-CONSIDER. Informally, a *-CONSIDER rule creates new goals in the inactive state such that the set of the existing and new goals together fully support some condition (such as the antecedent or the maintenance condition). On the other hand, A *-ACTIVATE rule activates all inactive goals that support some condition. Therefore, *-CONSIDER and *-ACTIVATE rules have different structures.

- A-ACTIVATE: Suppose a maintenance goal M is in the active state, and $\Phi = \{G_j\}$ is a set of goals such that $\bigwedge_j succ(G_j) \models maint(M)$. If a goal $G \in \Phi$ is inactive, then activate the goal.

$$\mathcal{M}(M) = \text{A} \wedge G \in \Phi \wedge \mathcal{G}(G) = \text{I} \xrightarrow{\text{A-ACTIVATE}} \text{activate}(G)$$

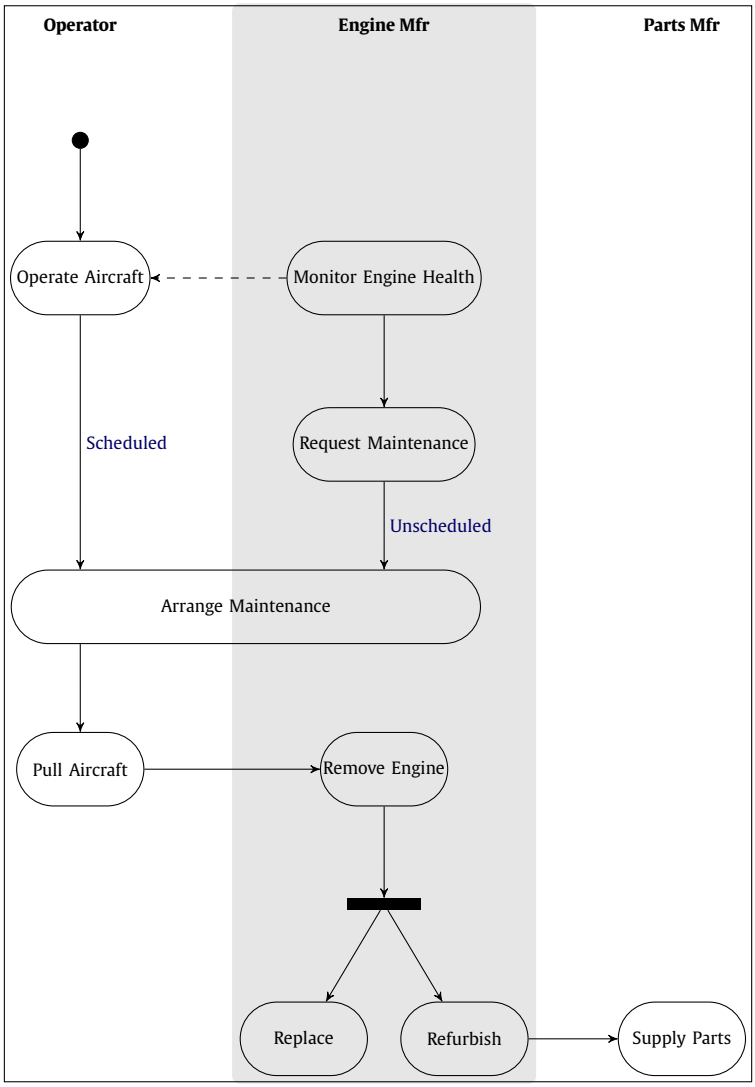


Fig. 9. Aerospace aftermarket [31].

- M-ACTIVATE: Suppose a maintenance commitment S is in the detached state, and $\Phi = \{M_j\}$ is a set of goals such that $\bigwedge_j maint(M_j) \models maint(S)$. If a goal $M \in \Phi$ is inactive, then debtor activates the goal.

$$\mathcal{S}(S) = \mathtt{D} \wedge M \in \Phi \wedge \mathcal{M}(M) = \mathtt{I} \overset{\text{M-ACTIVATE}}{\rightsquigarrow} \text{activate}(M)$$

- AD-ACTIVATE: Suppose a maintenance commitment S is in the conditional state, and $\Phi = \{G_j\}$ is a set of goals such that $\bigwedge_j succ(G_j) \models ant(S)$. If a goal $G \in \Phi$ is inactive, then debtor activates the goal.

$$\mathcal{S}(S) = \mathtt{C} \wedge G \in \Phi \wedge \mathcal{G}(G) = \mathtt{I} \overset{\text{AD-ACTIVATE}}{\rightsquigarrow} \text{activate}(G)$$

7. Applying the theory

We illustrate the value of integrated reasoning over maintenance commitments and goals with the aerospace aftermarket scenario of Fig. 9. The domain and scenario are found in van Aart et al. [31]. In the following Section 8 we will present formal theoretical results.

The figure contains a vertical 'swimlane' for each participant. The solid circle represents the start of the process flow and a rounded rectangle represents an activity. A solid directed edge represents a transition between a pair of activities. A dotted directed edge represents information exchange between the participants corresponding to a pair of activities. A label on an edge is a decoration to describe the transition. The solid horizontal bar represents an exclusive choice.

Table 2
Goals and commitments from the aerospace scenario.

ID	Goal, Commitment, or Event	Description
G_1	G(OPER, engine, ¬engine)	Operator's goal to buy an engine
G_2	G(MFR, engine, ¬engine)	Aircraft manufacturer's goal to provide an engine
G_3	G(OPER, paid, ¬paid)	Operator's goal to pay for an engine
G_4	G(MFR, paid, ¬paid)	Aircraft manufacturer's goal to receive the payment for an engine
C_1	C(OPER, MFR, engine $\wedge$ create(S_1), paid $\wedge$ create(S_2))	Operator's commitment to the manufacturer to pay and to report the engine health if the manufacturer provides the engine and commits to keeping it in a running condition
C_2	C(MFR, OPER, paid $\wedge$ create(S_2), engine $\wedge$ create(S_1))	Aircraft manufacturer's commitment to the operator to provide the engine and the commitment to keep the engine in a running condition if the operator pays and commits to reporting the engine health
S_1	S(MFR, OPER, $\top$, engine_running, aero_contract_expiry, engine_dead)	Aircraft manufacturer's commitment to the operator to keep the engine in a running condition
S_2	S(OPER, MFR, status_changed, health_reported, aero_contract_expiry, status_changed $\wedge\neg$ health_reported)	Operator's commitment to the manufacturer to report the engine health whenever the engine status changes
M_1	M(MFR, engine_running, aero_contract_expiry, engine_dead)	Aircraft manufacturer's goal to maintain the engine in a running condition
M_2	M(OPER, health_reported, aero_contract_expiry, engine_dead)	Operator's goal to regularly report the engine health
G_5	G(MFR, engine_running, engine_dead)	Aircraft manufacturer's goal to restore an engine to a running condition
G_6	G(OPER, health_reported, engine_dead)	Operator's goal to report engine health
G_7	G(MFR, penalty_paid, ¬penalty_paid)	Aircraft manufacturer's goal to pay a penalty
M_1	M(MFR, engine_running, aero_contract_expiry, engine_dead)	Aircraft manufacturer's goal to maintain the engine in a running condition
C_3	C(MFR, PMFR, engine_part, pfmr_paid)	Aircraft manufacturer's commitment to the parts manufacturer to pay if the parts manufacturer provides the engine part
C_4	C(PMFR, MFR, pfmr_paid, engine_part)	Parts manufacturer's commitment to the manufacturer to provide the engine part if the manufacturer pays
C_5	C(MFR, OPER, violate(S_1), penalty_paid)	Aircraft manufacturer's commitment to the operator to pay a penalty if it violates its maintenance commitment to keep the engine in a running condition
C_6	C(OPER, MFR, violate(S_2), release(S_1))	Operator's commitment to the manufacturer to release the latter from its commitment to maintaining the engine in a running state if the operator violates its maintenance commitment to report engine health

The participants in the scenario are an airline operator (OPER), an aircraft engine manufacturer (MFR), and a parts manufacturer (PMFR). The airline operator buys engines from the manufacturer. In addition to selling an engine, the manufacturer maintains the engine in an operational condition. In case a plane waits on ground for an engine to be serviced, the manufacturer pays a penalty to the operator. The manufacturer requires the operator: (1) to apprise the manufacturer of the engine health any time it changes (e.g., by providing engine sensor data), and (2) to allow the engine to be serviced during a scheduled maintenance window. After analyzing the engine health data, the manufacturer may request the operator for unscheduled maintenance on an engine. During the servicing of an engine, the manufacturer may either replace or refurbish the engine. The manufacturer procures parts for engine refurbishing from a parts manufacturer.

Table 2 shows the achievement and maintenance goals and commitments that model the aerospace aftermarket scenario. The achievement goals G_1 and G_2 are the airline operator's and manufacturer's goals for the engine, and G_3 and G_4 are their goals for the engine payment. The achievement commitments C_1 and C_2 are the mutual commitments between the operator and the manufacturer: the operator commits to paying for the engine and to providing the engine health (S_2), and the manufacturer commits to providing the engine and to keep it in a running condition (S_1). Observe that the goal G_2 provides antecedent support to the commitment C_1, and the goal G_3 provides antecedent support to the commitment C_2. S_1 is the unconditional maintenance commitment from the manufacturer to the operator to keep the engine in a running condition or to pay a penalty if the engine has a downtime. The manufacturer may create the maintenance goal M_1 to maintain the condition of S_1. Achievement goal G_5 is the manufacturer's goal to restore the engine to a running condition. The manufacturer creates G_5 each time it predicts that the engine needs maintenance. If the manufacturer is unable to restore the engine to a running condition, the manufacturer creates G_7 to pay a penalty for the engine downtime. S_2 is the

Table 3
Progression of configurations in an aerospace scenario.

Step	Rule	OPER Action	OPER State	MFR Action	MFR State
1	C-CREATE–Sec 6.2	create(C_1)	$\langle C_1^A \rangle$		$\langle C_1^A \rangle$
2	DETACH–TSY		$\langle C_1^D, S_1^D \rangle$	engine $\wedge$ create(S_1)	$\langle C_1^D, S_1^D \rangle$
3	M-CONSIDER–Sec 6.2		$\langle C_1^D, S_1^D \rangle$	consider(M_1)	$\langle M_1^I, C_1^D, S_1^D \rangle$
4	M-ACTIVATE–Sec 6.2		$\langle C_1^D, S_1^D \rangle$	activate(M_1)	$\langle M_1^M, C_1^D, S_1^D \rangle$
5	Life cycle–TSY	paid	$\langle C_1^S, S_1^D \rangle$		$\langle M_1^M, C_1^S, S_1^D \rangle$
6	Life cycle–Figs. 6, 7		$\langle \neg$engine_running, $S_1^B \rangle$		$\langle \neg$engine_running, $M_1^A, S_1^B \rangle$
7	A-CONSIDER–Sec 6.2		$\langle \neg$engine_running, $S_1^B \rangle$	consider(G_5)	$\langle \neg$engine_running, $G_5^I, M_1^A, S_1^B \rangle$
8	A-ACTIVATE–Sec 6.2		$\langle \neg$engine_running, $S_1^B \rangle$	activate(G_5)	$\langle \neg$engine_running, $G_5^A, M_1^A, S_1^B \rangle$
9	Life cycle–Figs. 6, 7		$\langle$engine_running, $S_1^D \rangle$	engine_running	$\langle$engine_running, $G_5^S, M_1^M, S_1^D \rangle$

unconditional maintenance commitment from the operator to the manufacturer to report the engine health regularly. The operator may create the maintenance goal M_2 to maintain the condition of S_2. At a regular interval, the operator creates G_6 to report the engine health to the manufacturer. The achievement commitments C_3 and C_4 are the mutual commitments between the aircraft manufacturer and the parts manufacturer: the aircraft manufacturer commits to the paying the parts manufacturer, and the parts manufacturer commits to providing the engine part. We assume that the agents agree on the states of the propositions in the commitments and goals.

Telang et al. [32, Section 5] model the aerospace aftermarket scenario using only achievement commitments and goals. Their model captures the maintenance aspects of the scenario in an unnatural manner by requiring an instance parameter for the repeating goals and commitments. In contrast, we employ the maintenance commitments and goals which naturally capture the maintenance aspects leading to a semantically rich and simpler model.

Table 3 shows a possible progression of the operator and manufacturer configurations. In Step 1, the operator employs C-CREATE rule to create C_1. In Step 2, MFR employs DETACH rule (TSY) and provides the engine and creates S_1, which detaches C_1. In Step 3, MFR employs M-CONSIDER rule to consider and activate the M_1. In Step 4, OPER pays the MFR, which satisfies commitment C_1 (TSY). In Step 5, suppose the engine fails and stops running. This causes M_1 to transition to `Active` and S_1 to transition to `Sustain`. In Step 6, MFR employs A-CONSIDER to consider and activate G_5 to restore the engine. In Step 7, MFR fixes the engine, which satisfies G_5 and causes M_1 to transition to `Monitoring`, and S_1 to `Sustain`.

8. Coherence and convergence

Goals and commitments, respectively, reflect the cognitive and social states of agents. How well these constructs cohere indicates how well a multiagent system is being enacted. Ideally, an agent should enter into commitments in accordance with its goals and take on goals that would lead to its commitments being satisfied or sustained. But an agent being autonomous may drop its goals and commitments arbitrarily.

We say an agent configuration is *coherent* if it satisfies the stated coherence properties over beliefs, goals, and commitments of an agent. These properties are given in Definition 45. Informally, the goals and commitments in a coherent configuration reflect the agent's rationality in that they 'line up' and the existence of one of them may be justified by the existence of others. For example, when an end goal is satisfied, an agent may drop its corresponding commitment and if a means goal fails, it may adopt another goal or decide to give up on the commitment.

A judicious set of practical rules would ensure that goals and commitments in a multiagent system remain coherent even though the agents act autonomously. The results in this section demonstrate that the set of practical rules given in Section 6.2 are such a set. As discussed at the end of the article, our methodology is generic in that the same approach can be used for alternative sets of practical rules.

8.1. Trace

Lemma 1 states that a life cycle rule maps a well-defined configuration to another well-defined configuration.

Lemma 1 *(Configuration update under life cycle rules). Let x be an agent with a configuration $S(x) = \langle \mathcal{B}, \mathcal{G}, \mathcal{M}, \mathcal{C}, \mathcal{S} \rangle$. Let L be a life cycle rule in which x applies action α. Let $S'(x) = \langle \mathcal{B}', \mathcal{G}', \mathcal{M}', \mathcal{C}', \mathcal{S}' \rangle$ be a tuple of functions for the beliefs, goals, and commitments of x, with the same signature as the respective state functions, after the application of α. Then $S'(x)$ is a unique configuration.*

Lemma 1 means we can write $S(x) \xrightarrow{\alpha} S'(x)$ where $S'(x)$ is the (unique) configuration of x after the application of action α.

We are now in the position to define the trace of system configurations, after a preliminary definition.

Definition 42 *(Successor configuration).* Let $\mathcal{M}$ be a system of agents $\mathscr{A} = x_1, \ldots, x_n$ and let S and S' be two system configurations of $\mathcal{M}$. Then S *progresses* to S' if and only if $\exists x \in \{1, \ldots, n\}$ and agent x applies action α, and $\forall y \in \{1, \ldots, n\}$: $S(y) \xrightarrow{\alpha} S'(y)$. We call S' a *successor system configuration* of S and say that S' *follows* from S.

Such an action α of agent x, which moves the system configuration from S to S', may be an action corresponding to a life cycle rule, or—if the agent follows our proposed practical rules of the next section—an action corresponding to a practical rule.

Intuitively, a trace is a sequence of successor configurations:

Definition 43. A *trace* is a (possibly infinite) sequence of system configurations $S_1, S_2, \ldots$, where for $i > 0$, configuration S_i follows from S_{i-1} as per Definition 42. Configuration S_0 is the *initial configuration* of the system. In S_0, the sets $\mathcal{G}$, $\mathcal{M}$, $\mathcal{C}$, and $\mathcal{S}$ from each agent's configuration are empty.

Our results later in this section centre on proving the convergence of traces under certain conditions:

Definition 44 *(Trace convergence).* A trace *converges infinitely often* to a configuration S if and only from every configuration S_i there is a $k \geq i$ and a configuration S_k such that $S_k = S$.

The intuition is that, on all paths in future, the converged state will be eventually reached. If a trace $S_1, S_2, \ldots$ converges infinitely often to a configuration S, and S is coherent, then the trace *sustains a coherent configuration*. Note that this repeated converge contrasts with the 'one time' convergence that is adequate for achievement commitments in TSY [4].

8.2. Coherence

Coherence is a property of the configuration of an agent. Together with the above notion of trace convergence, it will allow us to state our theorems below.

In informal terms, coherence seeks to characterize agent configurations wherein the beliefs, goals, and commitments of the agent make sense with respect to each other. That is, coherence captures an intuition of rationality, which goes beyond logical entailment to a weaker kind of internal 'consistency'. For example, an agent may adopt a certain goal only to attempt to satisfy a commitment of which it is the debtor: if it drops the commitment or the creditor releases it, the corresponding goal may become unnecessary (depending on what else relies on that goal). One of our theorems below shows that given our practical rules, if an agent loses coherence, that is only temporary in that it would eventually regain coherence.

The following definition is motivated by an analysis of the life cycles of the respective constructs, the definitions of maximality, and the support sets. As noted, our methodology is generic in that the same approach can be applied to other rule sets and coherence definitions, although of course the theorems may change accordingly.

Definition 45 *(Coherent configurations).* A configuration is *coherent* if and only if it satisfies all the properties below, in addition to those properties between achievement goals and achievement commitments of Telang et al. [4] (excluding those properties of TSY which pertain to suspension):

1. **M-comm to a-goal:** Suppose S is a maximal m-comm and Φ is a minimal subset of GAS(S) such that $G_i \in \Phi$ and $\bigwedge_i succ(G) \models ant(S)$. A coherent configuration satisfies:

   (a) If S is in conditional state, then each goal $G \in \Phi$ is active.
   $S \in maxs(\{\mathtt{C}\}) \implies \forall G \in \Phi, G \in maxg(\{\mathtt{A}\})$
   (b) If no goals exist in GAS(S), then S is expired, satisfied, terminated, detached or sustained.
   $\mathrm{GAS}(S) = \emptyset \implies S \in maxs(\{\mathtt{E}, \mathtt{S}, \mathtt{T}, \mathtt{D}, \mathtt{B}\})$
   (c) If all goals in Φ are satisfied, then S is detached.
   $\forall G \in \Phi, G \in maxg(\{\mathtt{S}\}) \implies S \in maxs(\{\mathtt{D}, \mathtt{B}\})$

2. **A-goal to m-comm**: Suppose G is a maximal a-goal and Φ is a subset of SAG(G) such that $S_i \in \Phi$ and $\bigwedge_i ant(S_i) \models succ(G)$.
   A coherent configuration satisfies:

   (a) If G is active, then each maintenance commitment $S \in \Phi$ is conditional.
   $G \in maxg(\{\mathtt{A}\}) \implies \forall S \in \Phi, S \in maxs(\{\mathtt{C}\})$
   (b) If no maintenance commitments exist that support G, then G is terminated or failed.
   $\mathrm{SAG}(G) = \emptyset \implies G \in maxg(\{\mathtt{T}, \mathtt{F}\})$
   (c) If each maintenance commitment $S \in \Phi$ is detached or sustained, then G is satisfied.
   $\forall S \in \Phi, S \in maxs(\{\mathtt{D}, \mathtt{B}\}) \implies G \in maxg(\{\mathtt{S}\})$

3. **M-goal to m-comm**: Suppose M is a maximal m-goal and Φ is a minimal subset of SSM(M) such that $S_i \in \Phi$ and $\bigwedge_i maint(S) \models maint(M)$. A coherent configuration satisfies:

(a) If M is in monitoring state, then each m-comm $S \in \Phi$ is detached.
$M \in maxm(\{\texttt{M}\}) \implies \forall S \in \Phi, S \in maxs(\{\texttt{D}\})$
(b) If no m-comms exist in SSM(M), then M is terminated, satisfied or failed.
$\text{SSM}(M) = \emptyset \implies M \in maxm(\{\texttt{T}, \texttt{S}, \texttt{F}\})$
(c) If an m-comm in Φ is in sustain state, then M is active, and if M is active, then an m-comm in Φ is in sustain state.
$\exists S \in \Phi, S \in maxs(\{\texttt{B}\}) \iff M \in maxm(\{\texttt{A}\})$

4. **M-comm to m-goal**: Suppose S is a maximal m-comm and Φ is a minimal subset of MSS(S) such that $M_i \in \Phi$ and $\bigwedge_i maint(M) \models maint(S)$. A coherent configuration satisfies:

(a) If S is detached, then each m-goal $M \in \Phi$ is in monitoring state.
$S \in maxs(\{\texttt{D}\}) \implies \forall M \in \Phi, M \in maxm(\{\texttt{M}\})$
(b) If no m-goals exist in MSS(S), then S is terminated, satisfied or violated.
$\text{MSS}(S) = \emptyset \implies S \in maxs(\{\texttt{T}, \texttt{S}, \texttt{V}\})$
(c) If an m-goal in Φ is in state `Monitoring`, then S is sustained, and if S is sustained, then an m-goal in Φ is in state `Monitoring`.
$\exists M \in \Phi, M \in maxm(\{\texttt{M}\}) \iff S \in maxs(\{\texttt{B}\})$

5. **M-goal to a-goal**: Suppose M is a maximal m-goal and Φ is a minimal subset of GMM(M) such that $G_i \in \Phi$ and $\bigwedge_i succ(G_i) \models maint(M)$. A coherent configuration satisfies:

(a) If M is in active state, then each goal $G \in \Phi$ is active.
$M \in maxm(\{\texttt{A}\}) \implies \forall G \in \Phi, G \in maxg(\{\texttt{A}\})$
(b) If no goals exist in GMM(M), then M is inactive, terminated, failed or satisfied.
$\text{GMM}(M) = \emptyset \implies M \in maxm(\{\texttt{I}, \texttt{T}, \texttt{F}, \texttt{S}\})$
(c) If each goal $G \in \Phi$ is satisfied, then M is in monitoring state.
$\forall G \in \Phi, G \in maxg(\{\texttt{S}\}) \implies M \in maxm(\{\texttt{M}\})$

A configuration that fails to satisfy any one of the above properties is *incoherent*.

8.3. Convergence results

We now proceed to prove the repeated convergence of traces under two assumptions.

First, convergence will not happen if commitments cannot be enacted upon. An assumption of *action fairness*—that all agents act towards achieving their commitments and goals—is not strong enough for our purpose, however. In fact, a stronger assumption is needed: *goal resolution*—we assume that an a-goal eventually reaches a terminal state, either positive (e.g., `Satisfied`) or negative (e.g., `Failed`); and that an m-goal will not remain indefinitely `Active`: that is, if an m-goal M is `Active` then the agent will consider a set of a-goals to restore $maint(M)$. Note that action fairness does not necessarily imply goal conclusion in all circumstances.

Second, for convergence, we cannot have forever-cycling commitments or goals. After a preliminary definition, the next two definitions explain what cycling is in the maintenance case.

Definition 46. Let τ be a trace of configurations $\langle S_0, S_1, \ldots\rangle$. Let $\langle S_i, S_j\rangle$, $j > i$ be a pair of configurations. We say that τ contains $\langle S_i, S_j\rangle$ if $S_i \in \tau$ and $S_j \in \tau$.

Definition 47. Let $S = \text{S}(x, y, l, m, d, f)$ be an m-comm and τ be a trace of configurations $\langle S_0, S_1, \ldots\rangle$. Suppose $\mathcal{S}(S) = \sigma$ in some configuration S_i and in some subsequent configuration S_j, where $j > i$. If τ contains infinite pairs of $\langle S_i, S_j\rangle$, $S_j \neq \texttt{S}$, then we say that S is *cycling* on τ.

Definition 48. Let $M = \text{M}(x, m, s, f)$ be an m-goal and τ be a trace of configurations $\langle S_0, S_1, \ldots\rangle$. Suppose $\mathcal{M}(M) = \sigma$ in some configuration S_i and in some subsequent configuration S_j, where $j > i$. If τ contains infinite pairs of $\langle S_i, S_j\rangle$, $S_j \neq \texttt{A}$, then we say that M is *cycling* on τ.

8.3.1. Results focusing on one agent

These theorems relate one agent's (s-) commitments and (a- and m-) goals. The first theorem says that an m-comm does not remain in `Sustain` infinitely long on a trace. For this, an agent might attempt to restore the m-comm to `Detached`, but if those attempts keep failing, eventually the agent will conclude that the commitment is not 'restorable'.

Definition 49 *(Restorable m-comm).* A maintenance commitment S of $x \in \mathscr{A}$ is *restorable* if, every time S transitions to `Sustain`, x can restore S to `Detached` in a finite time.

Theorem 1. *Suppose that an agent x will conclude that an m-comm is not restorable after a finite number of attempts. Let $S = \mathsf{S}(x, y, l, m, d, f)$ be an m-comm and τ be a trace of configurations $\langle S_0, S_1, \ldots\rangle$. Suppose in configuration S_c that $\mathsf{S}(S) = \mathtt{B}$. Then, $\exists$ a configuration $S_h, h > c$ such that $\mathsf{S}(S) \neq \mathtt{B}$.*

Proof of Theorem 1. Suppose in some configuration S_a where $a < c$, the m-comm S is detached: $\mathcal{S}(S) = \mathtt{D}$. In a configuration S_b, where $b > a$, agent x employs M-CONSIDER to consider a set of m-goals to support S. Let $\Phi_M = \{M_i\}$ be the minimal set of m-goals such that $\bigwedge_i maint(M_i) \models maint(S)$. Note that if S is restorable, this set cannot remain empty forever. In the configuration S_c, since S is in sustain, $maint(S) = \bot$. It follows that the maintenance condition of one or more m-goals $M_j \in \Phi_M$ is false.

Following the m-goal life cycle, all such m-goals M_j are active. For each goal M_j, in some configuration S_d, $d > c$, agent x employs A-CONSIDER to consider a set of a-goals to restore $maint(M_j)$.

Now let $\Phi_G = \{G_k\}$ be the minimal set of a-goals such that $\bigwedge_k succ(G_k) \models maint(M_j)$. We must consider four cases:

(a) In a configuration S_h, $h > d$, all goals in $\{G_k\}$ satisfy, thus making $maint(M_j) = \top$ for all M_j. From $\bigwedge_i maint(M_i) \models maint(S)$, it follows that $maint(S) = \top$. Following the m-comms life cycle, S transitions to `Detached`.
(b) In a configuration S_e, $e > d$, some a-goals in $\{G_k\}$ may fail, but the agent believes that $maint(M_j)$ is restorable. Agent x may reapply A-CONSIDER, and the proof follows case (a). However, since the agent concludes that M_j is not restorable after a finite number of attempts, case (b) cannot occur infinitely often, but will turn to case (c).
(c) In a configuration S_h, $h > d$, suppose the agent believes that $maint(M_j)$ is not restorable. Following the m-comm life cycle, S transitions to `Violated`.
(d) In a configuration S_h, $h > d$, S becomes respectively `Terminated`, `Satisfied` or `Violated` if released by y, discharged, or cancelled. □

The second theorem says that the trace sustains a coherent configuration.

Theorem 2. *Let $\tau = \langle S_0, S_1, \ldots\rangle$ be a trace. Then for any configuration S_i in τ, if S_i is not coherent, there is a subsequent configuration S_j, $j > i$, in τ such that S_j is coherent.*

Proof of Theorem 2. There are five cases to consider from Definition 45. Let S_i be the current agent configuration. We will show that if S_i is not coherent, then necessarily there exists a future configuration in the trace S_j, $j > i$ that is coherent.

Here we show the proof for the first case, that of m-comm to a-goal.

Suppose S is a maximal m-comm and Φ is a minimal subset of GAS(S) such that $G_i \in \Phi$ and $\bigwedge_i succ(G) \models ant(S)$.

(a) $S \in maxs(\{\mathtt{C}\}) \implies \forall G \in \Phi, G \in maxg(\{\mathtt{A}\})$: Suppose in state S_i, S is `Conditional` and maximal. By definition of GAS, $G \in \Phi \rightarrow \mathsf{G}(G) \in \{\mathtt{I}, \mathtt{A}\}$. If S is `Conditional`, then the debtor agent employs AD-CONSIDER to consider goals in Φ. Then the agent employs AD-ACTIVATE to activate goals in Φ. Hence in a subsequent configuration, $\mathsf{G}(G) = \mathtt{A}$.
(b) GAS(S) $= \emptyset \implies S \in maxs(\{\mathtt{E}, \mathtt{S}, \mathtt{T}, \mathtt{D}, \mathtt{B}\})$: If there are no a-goals supporting the antecedent of S, then necessarily S is `Expired`, `Satisfied`, `Terminated`, `Detached`, or `Sustain`. S cannot be `Conditional`: suppose in some configuration, GAS(S) is empty and $\mathsf{S}(S) = \mathtt{C}$. Then in a subsequent configuration, the agent would employ AD-CONSIDER to consider a set of goals $G \in$ GAS(S) making GAS(S) non-empty.
(c) $\forall G \in \Phi, G \in maxg(\{\mathtt{S}\}) \implies S \in maxs(\{\mathtt{D}, \mathtt{B}\})$: If all goals in Φ are `Satisfied`, then the antecedent of S must be true; hence S is `Detached`. If $maint(S) = \bot$, then S can be in `Sustain`.

The other four cases follow a similar pattern, with the equivalent subcases (a)–(c) for each. □

8.3.2. Results focusing on many agents

These theorems concern the m-comms in a multiagent system. They state that the agents together maintain their m-goals and commitments in a rational way.

As we showed above, a single agent reasons about and acts on its goals and commitments in accordance with its beliefs. When two or more agents are involved, their beliefs must be sufficiently in agreement or else they would not be able to interact felicitously. Beliefs have two sources in our approach (recall Section 3.4): an agent's beliefs arise from its observations as well as based on the agent's belief-world model. The former corresponds to sensing the world and the latter to the ontology in which the commitments and goals are expressed. We assume that (1) the agents have sufficiently accurate sensors that their observations of the common reality are in agreement, and (2) they have compatible belief-world models.

Theorem 3. *Suppose agent x in a multiagent system $\mathcal{M}$ has an m-goal $M = \mathsf{M}(x, m, s, f)$. Then the agents in $\mathcal{M}$ create minimal sets of m-comms, m-goals, and a-goals necessary to maintain the condition m.*

Proof of Theorem 3. There are two ways in which m can be maintained: either agent x maintains the m-goal on its own, or agent x persuades some other agent y to maintain x's m-goal via an m-comm $S = \mathsf{S}(y, x, l, m, d)$.

We treat the cases in turn. First, if x decides to maintain m itself, then should m become false, then x employs A-CONSIDER to consider a minimal set of achievement goals $\{G_i\}$ such that $G_i = \mathsf{G}(x, m_i, s_i, g_i)$ and $\bigwedge_i m_i \models m$.

In the second case, agent x employs C-CREATE to create an m-comm S as the debtor to some agent y; x and y share consistency of beliefs as above. Agent y in turn employs M-CONSIDER to consider a minimal set of m-goals $\{M_i\}$ such that $M_i = \mathsf{M}(y, m_i, s_i, f_i)$, and $\bigwedge_i m_i \models m$. Similar to the first case, if m_i becomes false, then agent y employs A-CONSIDER to consider a minimal set of a-goals to restore m_i.

In both cases, thus the minimal necessary sets are created. □

Theorem 4. *Suppose agent x in a multiagent system $\mathcal{M}$ has an m-comm $S = \mathsf{S}(x, y, l, m, d, f)$. Then the agents in $\mathcal{M}$ create minimal sets of m-goals and a-comms and a-goals necessary to maintain the condition m.*

Proof of Theorem 4. The proof for the case of an m-comm relies on the proof above for the case of an m-goal. Suppose x is the debtor of m-comm S as in the theorem statement.

First, by AD-CONSIDER if $\mathsf{S}(S) = \mathtt{C}$ then x considers an a-goal $\mathsf{G}(x, l, f_1)$ in order to detach S. Once $\mathsf{S}(S) = \mathtt{D}$ then by M-CONSIDER x considers an m-goal $\mathsf{M}(x, m, s, f_2)$ in order to maintain m. This m-goal are themselves sustained as in Theorem 3. □

9. Related work

This article draws on Telang et al.'s [4] study of achievement commitments and goals. That work does not consider the maintenance of either construct. Maintenance goals are handled by, for instance, Duff et al. [7], who do not consider commitments. The developments we provide to handle the maintenance of commitments are non-trivial, including new definitions of support functions, closure and coherence, and new life cycle rules and theorems. Other differences from TSY are that we handle suspension operationally rather than with life cycle states and practical rules, which reduces the complexity of our model. The theorems, naturally, are unique to the presence of maintenance constructs.

9.1. Maintenance as an independent construct

The few works that address maintenance commitments reduce them to achievement commitments albeit with more complex temporal formulae. Mallya et al. [5] write maintenance commitments as achievement commitments for temporal formulae of the form 'always m'. However, such a representation is inadequate because it does not capture potentially repeated interventions by the debtor to re-establish m should it fail. In other words, the normative force of a maintenance commitment is not toward achieving an inviolable 'always m' but in ensuring and restoring m as often as it takes. Further, these works do not study the life cycle of maintenance commitments, nor the connection to goals.

9.2. Commitments and time

Chesani et al. [6] define commitments with universally quantified properties during a time interval. They employ a Reactive Event Calculus framework, which supports greater temporal expressiveness than ours. Their formulation does not have an explicit notion of m-commitment, having only a limited maintenance life cycle. We anticipate that an approach such as ours could be developed for their technical framework.

Several lines of work study commitments and time, from modeling or verification perspectives. Among these, El-Menshawy et al. [33] introduce a temporal logic with modalities for commitments and their fulfillment and violation. The authors employ symbolic model checking over ordered binary decision diagrams. Bentahar and colleagues continued this line of work, for instance to model-checking probabilistic social commitments [34], and temporal knowledge and commitments [35]. Bataineh et al. [36] and especially El-Menshawy et al. [37] survey.

Chopra and Singh [38] define a commitment-specification language, Cupid, that is first-order and maps to event expressions using relational algebra. Cupid can capture commitments of the form of "for each insurance claim, I will provide a payment" but does not capture maintenance in that it handles only the achievement of the consequent: it does not handle a consequent being forever or repeatedly made true. Cupid and Custard [39] (an extension to norms) do support an explicit notion of time and have a subtle notion of the non-occurrence of events. They map each commitment or other norm specification to a relevant information model; such a model could be an agent's private information store or could be a notionally public information store for a multiagent system. The non-occurrence of events is crucial in practical uses of commitments and other norms but can be unwieldy to handle precisely in a query language for an information store. Cupid and Custard generate the relevant (usually, large) queries automatically. It would be interesting to extend these specification languages to include m-commitments.

9.3. Beliefs, goals, and monitoring

Günay et al. [40] propose a framework to enable agents to create a commitment protocol dynamically. Such an approach to agent interaction provides for runtime coordination between agents' goals. Their framework admits achievement commit-

ments. Fornara and Colombetti [41] describe how to monitor achievement commitments that express rich domain meanings using the OWL ontology language. Both these works could be fruitfully extended to maintenance commitments.

Al-Saqqar et al. [42] develop a logic that considers both agents' knowledge and commitments. [43] develop a formal semantics where, in the spirit of [44,45], trust is taken as a primitive. In conceptual terms, trust is the flip side of commitments since it expresses conditional expectations. In contrast, we develop an operational semantics relating achievement and maintenance goals and commitments. However, the coherence between an agent's goals and commitments would be a factor in assessing the agent to be trustworthy [46].

Determining whether an agent keeps to its (achievement) commitments is known as monitoring [47]. The act of monitoring has a maintenance-like ongoing nature. Extending the approach of, e.g., Kafalı and Torroni [48], to understand monitoring and responding to failures is future work. Another useful direction is incorporating dialectical commitments for handling disputes between agents as to the facts [49] and understanding what effects such disputes may have on coherence.

9.4. Analyzing sets of commitments

Criado et al. [50] discuss a notion of coherence where they seek to identify consistent sets of norms. They formulate coherence as constraint satisfaction where an agent can compute preferences across its norms with respect to its cognitive state. In a somewhat similar approach, Desai et al. [51] evaluate sets of commitments, viewed as contracts, from the perspective of the preferences of the participants. In contrast to these approaches, we seek not to evaluate coherence but to show how each agent is individually coherent and how, linked through their commitments, the agents are collectively coherent over time.

9.5. Understanding agent communications

Almost since the inception of the study of social commitments in AI, they have been used as a basis for formalizing agent communications [52–54]. The general idea is that, setting aside syntactical details, the meaning of a communication between autonomous agents must be expressed in terms of how it affects their social state. This view goes back to the earliest modern studies of communication [55], as more recent work in philosophy [56] makes clear.

Previous AI approaches on agent communications sought to define communications in terms of the mental states of agents. Such approaches, e.g., Bretier and Sadek [57], suffer from the problem that the mental state of an agent is not visible [58]. In addition, the naïve sincerity conditions that the mental approaches assume, such as that agents must tell the truth are largely unviable. Commitments provide the basis for validity conditions as motivated by Habermas [59] in terms of objective, subjective, and practical claims [11]. The subjective claims combine commitments with beliefs and goals (or intentions in some work); the practical claims could be expanded to incorporate beliefs and goals as well. The present article introduces m-commitments and relates m-commitments and a-commitments with beliefs and goals. In this way, it provides improved constructs for formulating a new social semantics for agent communications.

Interestingly, formal models of argumentation rely upon the notion of a commitment store [60,61] although those are motivated as dialectical commitments. Argumentation itself provides a foundation for agent communication [62] and dialogue, including leading up to practical (or 'action') commitments of the sort investigated in this article [63].

9.6. Commitments as a basis for communication protocols

A communication protocol identifies two or more roles as well as messages to be exchanged between agents playing those roles. Commitments provide a natural basis for specifying communication protocols by capturing the changing state of an enactment of the protocol. Specifically, zero or more commitments may exist initially and each message potentially causes a transition in the life cycles of the commitments (e.g., by creating, detaching, or satisfying them).

Venkatraman and Singh [52] express commitments incorporating temporal expressions and show how agents can determine compliance of other agents with respect to their commitments. Fornara and Colombetti [64] provide an operational model for protocols in which states and transitions are defined in terms of commitments. Yolum and Singh [65] and Chopra and Singh [66] show how to generate state machines for protocols given their messages and the meaning of each message. Chopra and Singh [67] take the above idea further by seeking to specify messages only in terms of their constitutive meanings, showing how to achieve interoperability in asynchronous settings. All of the above ideas could be enhanced to capture more subtle interactions through the expanded life cycle of maintenance commitments.

Commitments offer interesting opportunities for engineering multiagent systems based on protocols and adapters for agents who participate in those protocols. An early work in this regard is due to Desai et al. [68], who show how commitment-based protocols can be mapped to adapters (in the form of role skeletons) through which each agent can be implemented using rule-based policies to capture reasoning specific to that agent. Baldoni et al. [16] show how to implement commitment-based interactions using the JaCaMo+ framework that provides various cognitive and organizational primitives. Baldoni et al. [69] show how to type check the participation of a role in a protocol based on commitments, as a way to identify and prevent errors. The benefits of using commitments to decouple agents become all the more apparent when we consider the potential high complexity of BDI agents and the resultant challenges in testing them [70]. We posit

that commitment protocols, especially in light of the results of this article, can help produce agents that are simpler and more modular in how they deal with their beliefs and goals with respect to their interactions with others.

The foregoing approaches consider achievement commitments and would benefit from the introduction of maintenance commitments. Moreover, the relationship between maintenance commitments and goals established in this article can facilitate the development and verification of agents who would participate in protocols that rely upon maintenance commitments.

10. Conclusion

This article studies the dynamic relationships between a rational agent's cognitive state (i.e., beliefs and goals) and its social state (i.e., commitments). First-class maintenance commitments are a powerful new type of social commitments that enable richer relationships than otherwise possible, thereby supporting expanded forms of collaboration.

By formalizing the concept of a maintenance commitment, we defined an operational semantics based on life cycle rules and practical rules. Further, by motivating an extended notion of coherence, we proved that a system of rational agents following the practical rules will have coherence between the achievement and maintenance goals and commitments. In a real-world scenario, we demonstrated how maintenance commitments naturally capture a periodic contract relationship between multiple agents.

We proposed a set of practical rules that captures certain intuitions and leads to results on coherence. One can conceive of alternative sets of such rules: in this regard, our methodology is fully generic.

Future work is, first, to allow the explicit representation of time in commitments, as indicated by Chesani et al. [6] and Fornara and Colombetti [71]. Particularly interesting is how temporally qualified propositions interact with the nature of maintenance commitments. Second, our approach adopts propositional goals and commitments: a future direction is to consider enhanced representations that involve decidable fragments of first-order logic.

Another interesting future direction is to consider maintenance as an essential component of norms broadly, not only commitments. Such an approach would be an enabler of a new theory of resilient norms that maintain their coherence with each other and the cognitive constructs, supporting both probabilistic reasoning about norms [72] and new verification techniques [43]. We also conjecture that recent progress in formal models of protocols for vocabulary alignment [9] could benefit from how we characterize coherence between commitments and goals, the more so in light of maintenance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We thank the anonymous reviewers for constructive comments, and the reviewers of AAAI'21 for their suggestions on an earlier version of this work. MPS thanks the US National Science Foundation (grant IIS-1908374) for support. NYS was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under grant 952215.

References

[1] C. Castelfranchi, Commitments: from individual intentions to groups and organizations, in: Proceedings of International Conference on Multiagent Systems (ICMAS), 1995, pp. 41–48.

[2] M.P. Singh, Social and psychological commitments in multiagent systems, in: Proceedings of AAAI Fall Symposium on Knowledge and Action at Social and Organizational Levels, 1991, pp. 104–106, https://www.csc2.ncsu.edu/faculty/mpsingh/papers/mas/fall-symp-91-longer.pdf.

[3] M.P. Singh, Commitments in multiagent systems: some history, some confusions, some controversies, some prospects, in: F. Paglieri, L. Tummolini, R. Falcone, M. Miceli (Eds.), The Goals of Cognition: Essays in Honor of Cristiano Castelfranchi, College Publications, London, 2012, pp. 591–616, https://www.csc2.ncsu.edu/faculty/mpsingh/papers/mas/Commitments-for-MAS.pdf.

[4] P.R. Telang, M.P. Singh, N. Yorke-Smith, A coupled operational semantics for goals and commitments, J. Artif. Intell. Res. 65 (2019) 31–85, https://doi.org/10.1613/jair.1.11494.

[5] A.U. Mallya, P. Yolum, M.P. Singh, Resolving commitments among autonomous agents, in: Proceedings of the International Workshop on Agent Communication Languages (ACL), 2003, pp. 166–182.

[6] F. Chesani, P. Mello, M. Montali, P. Torroni, Representing and monitoring social commitments using the event calculus, Auton. Agents Multi-Agent Syst. 27 (2013) 85–130, https://doi.org/10.1007/s10458-012-9202-0.

[7] S. Duff, J. Thangarajah, J. Harland, Maintenance goals in intelligent agents, Comput. Intell. 30 (2014) 71–114, https://doi.org/10.1111/coin.12000.

[8] P. Chocron, M. Schorlemmer, Inferring commitment semantics in multi-agent interactions, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Stockholm, Sweden, 2018, pp. 1150–1158.

[9] P.D. Chocron, M. Schorlemmer, Vocabulary alignment in openly specified interactions, J. Artif. Intell. Res. 68 (2020) 69–107, https://doi.org/10.1613/jair.1.11497.

[10] A.U. Mallya, M.P. Singh, An algebra for commitment protocols, Auton. Agents Multi-Agent Syst. 14 (2007) 143–163, https://doi.org/10.1007/s10458-006-7232-1.
[11] M.P. Singh, A social semantics for agent communication languages, in: Proceedings of the 1999 IJCAI Workshop on Agent Communication Languages, 2000, pp. 31–45.
[12] E. Marengo, M. Baldoni, A.K. Chopra, C. Baroglio, V. Patti, M.P. Singh, Commitments with regulations: reasoning about safety and control in regula, in: Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'11), Taipei, Taiwan, 2011, pp. 467–474.
[13] P.R. Telang, M.P. Singh, N. Yorke-Smith, Maintenance of social commitments in multiagent systems, in: Proceedings of the 35th Conference on Artificial Intelligence, AAAI, Vancouver, BC, Canada, 2021, pp. 1–9.
[14] M. Dastani, L.W.N. van der Torre, N. Yorke-Smith, Commitments and interaction norms in organisations, Auton. Agents Multi-Agent Syst. 31 (2017) 207–249, https://doi.org/10.1007/s10458-015-9321-5.
[15] F. Meneguzzi, M.C. Magnaguagno, M.P. Singh, P.R. Telang, N. Yorke-Smith, GoCo: planning expressive commitment protocols, Auton. Agents Multi-Agent Syst. 32 (2018) 459–502, https://doi.org/10.1007/s10458-018-9385-0.
[16] M. Baldoni, C. Baroglio, F. Capuzzimati, R. Micalizio, Commitment-based agent interaction in JaCaMo+, Fundam. Inform. 159 (2018) 1–33, https://doi.org/10.3233/FI-2018-1656.
[17] A.S. Rao, M.P. Georgeff, An abstract architecture for rational agents, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR), Cambridge, MA, USA, 1992, pp. 439–449.
[18] K.L. Myers, N. Yorke-Smith, A cognitive framework for delegation to an assistive user agent, in: Proceedings of AAAI 2005 Fall Symposium on Mixed-Initiative Problem-Solving Assistants, Arlington, VA, USA, 2005, pp. 94–99.
[19] S. Visser, J. Thangarajah, J. Harland, F. Dignum, Preference-based reasoning in BDI agent systems, Auton. Agents Multi-Agent Syst. 30 (2016) 291–330, https://doi.org/10.1007/s10458-015-9288-2.
[20] M.P. Singh, The intentions of teams: team structure, endodeixis, and exodeixis, in: Proceedings of the 13th European Conference on Artificial Intelligence (ECAI), John Wiley, Brighton, United Kingdom, 1998, pp. 303–307.
[21] M.P. Singh, Group ability and structure, in: Y. Demazeau, J.-P. Müller (Eds.), Decentralized Artificial Intelligence, vol. 2, Elsevier/North-Holland, Amsterdam, 1991, pp. 127–145.
[22] S.O. Hansson, Logic of Belief Revision, in: E.N. Zalta (Ed.), Spring 2022 ed., The Stanford Encyclopedia of Philosophy, Metaphysics Research Lab, Stanford University, 2022.
[23] M.P. Singh, An ontology for commitments in multiagent systems: toward a unification of normative concepts, Artif. Intell. Law 7 (1999) 97–113, https://doi.org/10.1023/A:1008319631231.
[24] J. Harland, D.N. Morley, J. Thangarajah, N. Yorke-Smith, An operational semantics for the goal life-cycle in BDI agents, Auton. Agents Multi-Agent Syst. 28 (2014) 682–719, https://doi.org/10.1007/s10458-013-9238-9.
[25] S. Duff, J. Harland, J. Thangarajah, On proactivity and maintenance goals, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Hakodate, Japan, 2006, pp. 1033–1040.
[26] M.P. Singh, Semantical considerations on dialectical and practical commitments, in: Proceedings of the Conference on Artificial Intelligence, AAAI, Chicago, IL, USA, 2008, pp. 176–181.
[27] A.K. Chopra, M.P. Singh, Multiagent commitment alignment, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Budapest, Hungary, 2009, pp. 937–944.
[28] M. Winikoff, L. Padgham, J. Harland, J. Thangarajah, Declarative and procedural goals in intelligent agent systems, in: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR), Toulouse, France, 2002, pp. 470–481.
[29] T.C. King, A. Günay, A.K. Chopra, M.P. Singh Tosca, Operationalizing commitments over information protocols, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), IJCAI, Melbourne, Australia, 2017, pp. 256–264.
[30] A.K. Chopra, M.P. Singh, Generalized commitment alignment, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Istanbul, Turkey, 2015, pp. 453–461.
[31] C.J. van Aart, J. Chábera, M. Dehn, M. Jakob, K. Nast-Kolb, J.L.C.F. Smulders, P.P.A. Storms, C. Holt, M. Smith, Use Case Outline and Requirements, Deliverable D6.1, IST CONTRACT Project, 2007.
[32] P.R. Telang, M.P. Singh, N. Yorke-Smith, Relating goal and commitment semantics, in: Proceedings of the International Workshop on Programming Multi-Agent Systems (ProMAS), Taipei, Taiwan, 2011, pp. 22–37.
[33] M. El-Menshawy, J. Bentahar, H. Qu, R. Dssouli, On the verification of social commitments and time, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Taipei, Taiwan, 2011, pp. 483–490.
[34] K. Sultan, J. Bentahar, M. El-Menshawy, Model checking probabilistic social commitments for intelligent agent communication, Appl. Soft Comput. 22 (2014) 397–409, https://doi.org/10.1016/j.asoc.2014.04.014.
[35] F. Al-Saqqar, J. Bentahar, K. Sultan, W. Wan, E.K. Asl, Model checking temporal knowledge and commitments in multi-agent systems using reduction, Simul. Model. Pract. Theory 51 (2015) 45–68, https://doi.org/10.1016/j.simpat.2014.11.003.
[36] A.S. Bataineh, J. Bentahar, M. El-Menshawy, R. Dssouli, Specifying and verifying contract-driven service compositions using commitments and model checking, Expert Syst. Appl. 74 (2017) 151–184, https://doi.org/10.1016/j.eswa.2016.12.031.
[37] M. El-Menshawy, J. Bentahar, W.E. Kholy, P. Yolum, R. Dssouli, Computational logics and verification techniques of multi-agent commitments: survey, Knowl. Eng. Rev. 30 (2015) 564–606, https://doi.org/10.1017/S0269888915000065.
[38] A.K. Chopra, M.P. Singh, Cupid: commitments in relational algebra, in: Proceedings of the Conference on Artificial Intelligence (AAAI), Austin, TX, USA, 2015, pp. 2052–2059.
[39] A.K. Chopra, M.P. Singh Custard, Computing norm states over information stores, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Singapore, 2016, pp. 1096–1105.
[40] A. Günay, M. Winikoff, P. Yolum, Dynamically generated commitment protocols in open systems, Auton. Agents Multi-Agent Syst. 29 (2015) 192–229, https://doi.org/10.1007/s10458-014-9251-7.
[41] N. Fornara, M. Colombetti, Representation and monitoring of commitments and norms using OWL, AI Commun. 23 (2010) 341–356.
[42] F. Al-Saqqar, J. Bentahar, K. Sultan, On the soundness, completeness and applicability of the logic of knowledge and communicative commitments in multi-agent systems, Expert Syst. Appl. 43 (2016) 223–236, https://doi.org/10.1016/j.eswa.2015.08.019.
[43] N. Drawel, H. Qu, J. Bentahar, E.M. Shakshuki, Specification and automatic verification of trust-based multi-agent systems, Future Gener. Comput. Syst. 107 (2020) 1047–1060, https://doi.org/10.1016/j.future.2018.01.040.
[44] M.P. Singh, Trust as dependence: a logical approach, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Taipei, Taiwan, 2011, pp. 863–870.
[45] E. Paja, A.K. Chopra, P. Giorgini, Trust-based specification of sociotechnical systems, Data Knowl. Eng. 87 (2013) 339–353, https://doi.org/10.1016/j.datak.2012.12.005.
[46] A.M. Singh, M.P. Singh, Wasabi: a conceptual model for trustworthy artificial intelligence, IEEE Computer 56 (2023) 20–28, https://doi.org/10.1109/MC.2022.3212022.
[47] M. Dastani, L.W.N. van der Torre, N. Yorke-Smith, Commitments and interaction norms in organisations, Auton. Agents Multi-Agent Syst. 31 (2017) 207–249, https://doi.org/10.1007/s10458-015-9321-5.

[48] Ö. Kafalı, P. Torroni, Comodo: collaborative monitoring of commitment delegations, Expert Syst. Appl. 105 (2018) 144–158, https://doi.org/10.1016/j.eswa.2018.03.057.
[49] P.R. Telang, A.K. Kalia, J.F. Madden, M.P. Singh, Combining practical and dialectical commitments for service engagements, in: Proceedings of the 13th International Conference on Service-Oriented Computing (ICSOC), Goa, India, 2015, pp. 3–18.
[50] N. Criado, E. Black, M. Luck, A coherence maximisation process for solving normative inconsistencies, Auton. Agents Multi-Agent Syst. 30 (2016) 640–680, https://doi.org/10.1007/s10458-015-9300-x.
[51] N. Desai, N.C. Narendra, M.P. Singh, Checking correctness of business contracts via commitments, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Estoril, Portugal, 2008, pp. 787–794.
[52] M. Venkatraman, M.P. Singh, Verifying compliance with commitment protocols: enabling open Web-based multiagent systems, Auton. Agents Multi-Agent Syst. 2 (1999) 217–236, https://doi.org/10.1023/A:1010056221226.
[53] R.A. Flores, P. Pasquier, B. Chaib-draa, Conversational semantics sustained by commitments, Auton. Agents Multi-Agent Syst. 14 (2007) 165–186, https://doi.org/10.1007/s10458-006-0011-1.
[54] A.K. Chopra, A. Artikis, J. Bentahar, M. Colombetti, F. Dignum, N. Fornara, A.J.I. Jones, M.P. Singh, P. Yolum, Research directions in agent communication, ACM Trans. Intell. Syst. Technol. 42 (2013) 20:1–20:23, https://doi.org/10.1145/2438653.2438655.
[55] J.L. Austin, How to do Things with Words, Clarendon Press, Oxford, 1962.
[56] M. Sbisà, How to read Austin, Pragmatics 17 (2007) 461–473, https://doi.org/10.1075/prag.17.3.06sbi.
[57] P. Bretier, M.D. Sadek, A rational agent as the kernel of a cooperative spoken dialogue system: implementing a logical theory of interaction, in: Proceedings of the ECAI Workshop on Agent Theories, Architectures, and Languages (ATAL), Budapest, Hungary, 1996, pp. 189–203.
[58] M.P. Singh, Agent communication languages: rethinking the principles, IEEE Comput. 31 (1998) 40–47, https://doi.org/10.1109/2.735849.
[59] J. Habermas, The Theory of Communicative Action, Volumes 1 and 2, Polity Press, Cambridge, United Kingdom, 1984.
[60] P. McBurney, S. Parsons, Posit spaces: a performative model of e-commerce, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Melbourne, Australia, 2003, pp. 624–631.
[61] D.N. Walton, E.C.W. Krabbe, Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning, State University of New York Press, Albany, New York, 1995.
[62] L. Amgoud, N. Maudet, S. Parsons, An argumentation-based semantics for agent communication languages, in: Proceedings of the 15th European Conference on Artificial Intelligence (ECAI), Amsterdam, Netherlands, 2002, pp. 38–42.
[63] P. McBurney, S. Parsons, Argument schemes and dialogue protocols: Doug Walton's legacy in artificial intelligence, IfCoLog J. Log. Appl. 8 (2021) 263–290, https://collegepublications.co.uk/ifcolog/?00043.
[64] N. Fornara, M. Colombetti, Operational specification of a commitment-based agent communication language, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Melbourne, Australia, 2002, pp. 535–542.
[65] P. Yolum, M.P. Singh, Commitment machines, in: Proceedings of the International Workshop on Agent Theories, Architectures, and Languages (ATAL), Seattle, WA, USA, 2002, pp. 235–247.
[66] A. Chopra, M.P. Singh, Nonmonotonic commitment machines, in: Advances in Agent Communication: Proceedings of the International Workshop on Agent Communication Languages (ACL 2003), Melbourne, Australia, 2004, pp. 183–200.
[67] A.K. Chopra, M.P. Singh, Constitutive interoperability, in: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), Estoril, Portugal, 2008, pp. 797–804.
[68] N. Desai, A.U. Mallya, A.K. Chopra, M.P. Singh, Interaction protocols as design abstractions for business processes, IEEE Trans. Softw. Eng. 31 (2005) 1015–1027, https://doi.org/10.1109/TSE.2005.140.
[69] M. Baldoni, C. Baroglio, F. Capuzzimati, R. Micalizio, Type checking for protocol role enactments via commitments, Auton. Agents Multi-Agent Syst. 32 (2018) 349–386, https://doi.org/10.1007/s10458-018-9382-3.
[70] M. Winikoff, S. Cranefield, On the testability of BDI agent systems, J. Artif. Intell. Res. 51 (2014) 71–131, https://doi.org/10.1613/jair.4458.
[71] N. Fornara, M. Colombetti, Specifying and enforcing norms in artificial institutions, in: Declarative Agent Languages and Technologies VI, Revised Selected and Invited Papers, Springer, 2009, pp. 1–17.
[72] S. Cranefield, F. Meneguzzi, N. Oren, B.T.R. Savarimuthu, A Bayesian approach to norm identification, in: Proceedings of the European Conference on Artificial Intelligence (ECAI), Amsterdam, Netherlands, 2016, pp. 622–629.