Commitment-Based Multiagent Decision Making

Viji R. Avali and Michael N. Huhns

Department of Computer Science and Engineering, University of South Carolina
Columbia, SC 29208 USA

Abstract. In a cooperative system, multiple dynamic agents work together and share resources to achieve common goals, while simultaneously pursuing their individual goals. Interactions among the agents in such a cooperative system are critical to its successful behavior, and we believe that commitments are the proper abstraction to characterize the interactions. Commitments then become the basis for monitoring and controlling the system and tracking the progress towards its goals.

Commitments are binary relationships that bind two agents: a "debtor agent" that promises to provide a particular service for a "creditor agent". Their role is to represent agreements between the agents and prevent potential conflicts while the agents collaborate to achieve the system's common goals, which are imposed from outside. But the willingness to participate in achieving the goals comes from within the agent and that is why the beliefs, desires, and intentions of the agents are crucial in formalizing commitments. In this paper, we have formalized commitments in terms of the agents' internal states of mind—their beliefs, desires, and intentions. This formalization addresses what it means for a participating agent to promise or to satisfy a commitment. The formalization uses a branching-time computational tree logic framework with commitment definitions and operations to define a commitment-centric cooperative multiagent environment.
Keywords: Commitments; BDI; CTL*.

1 Introduction and Motivation

In a cooperative system, multiple dynamic entities work together and share their resources to achieve common goals, while simultaneously pursuing their individual goals. In real-world business environments, participants interact by exchanging goods and providing services for each other. In seeking and providing services, the participants form associations, make promises, commit to levels of functionality and quality, satisfy what they promised, and attempt to achieve their intended goals. We believe that in an environment where software agents are the participants, it is the binary relationship of *commitment* [1, 10, 11, 5] that associates the agents with one another and represents multiagent interactions. Commitments can characterize—from an external viewpoint—not only the interactions between the agents, but also the overall multiagent system behavior.

Recent work on the concept of commitments has provided ways for an agent to evaluate a commitment and decide whether or not to promise it (as the debtor

of the commitment) or accept it (as its creditor). However, current theories for commitments deal with only a single commitment and do not provide any help to an agent in relating or comparing several commitments. For example, if an agent has made two or more commitments, in which order should the agent work to satisfy them?

Our approach to this problem is to use the agent's beliefs, desires, and intentions to make decisions about commitments. The agent can then decide rationally when to accept, abandon, cancel, or devote resources to a commitment. The agent can also decide rationally in which order to satisfy its commitments. Moreover, a *commitment-driven decision theory* can be utilized to expressively model a cooperative multiagent environment. Development of this comprehensive theory is one of our research objectives.

Such a development should relate commitments to their effects on each of the participating agents' own *internal state of mind*. What does an autonomous agent believe when it creates a commitment? What does such an agent desire when it cancels its commitment? Many similar questions need to be addressed in order to develop a commitment-driven decision theory.

There has been a lot of work done on the belief, desire, and intention (BDI) architecture. Cohen and Levesque [2] explore the rational balance needed among beliefs, goals, actions, and intentions using a linear-time model. Rao and Georgeff [9] present a possible-worlds formalism for the BDI architecture using Computation Tree Logic (CTL). This work mentions that the BDI architecture could be extended to commitments by considereing them as part of multiagent scenarios.

There is also a rich literature on commitments. They are now well defined [12] and there is a formal representation for commitment operations [6]. Branching-time computational tree logic has been used to describe a commitment's typical structure [13], life-cycle, and various operations that are involved throughout its existence. However, most of the endeavors have focused on the external structure, properties, and verification of commitments. They do not explicitly formalize how commitments are understood by the participating agents themselves. These two areas (the BDI architecture and commitments) have been addressed separately and there has been little attempt to combine them (cf. [4]). Our work aims to integrate the two areas and formalize commitments in terms of the agent's beliefs, desires, and intentions in a CTL* framework, as indicated in Figure 1.

An agent's beliefs, desires and intentions define its *internal state of mind*. This paper formally defines commitments in terms of participating agents' beliefs, desires, and intentions. We use Rao and Georgeff's BDI framework [9] and Emerson's CTL framework [3], as well as earlier definitions for commitments [12] and operations on them [6].

The structure of this paper is as follows: Section 2 describes the major domain-independent types of commitments we have identified. Section 3 introduces a commitment-driven service-oriented multiagent environment and presents its underlying assumptions. Section 4 revisits the BDI_{CTL*} framework to describe agents in this environment. Section 5 introduces commitments and operations on them. Section 6 develops our formalization of commitments in a

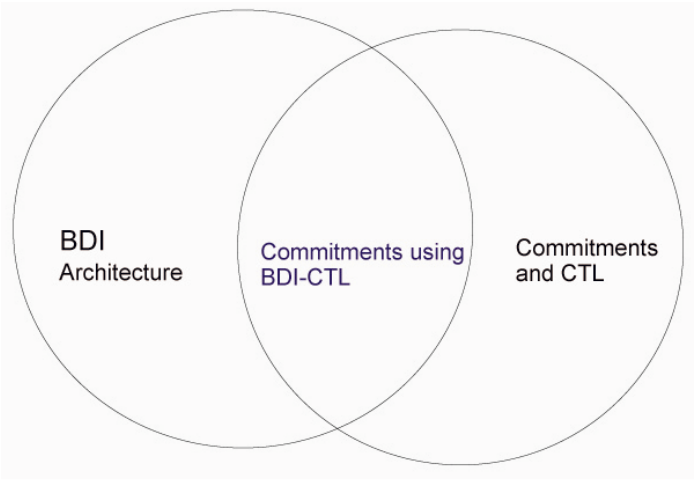

Fig. 1. Our results bridge BDI architectures, which are internal to agents, and commitments, which are public and involve the intentions among two or more agents. Our results are formalized using CTL*.

BDI_{CTL*} framework and provides definitions for all commitment operations. Section 7 presents an example of how our formalization can be used to explain, interpret, and model real-world multiagent systems. Lastly, Section 8 summarizes our formalization and presents future directions for this research.

2 Types of Commitments

Commitments associate one agent (the creditor) to another (the debtor) and are directed from the debtor to the creditor. They can be categorized into two basic types: discrete and continuous.

Discrete commitments have a lifetime, are created, remain active, and, at some point, cease to exist.
Example: agent Alice commits to pay $5 to agent Bob. The commitment ceases to exist when Alice pays Bob the money.

Continuous commitments are created and remain active indeterminately and until canceled. As an example of this type, a control system in a nuclear plant has a continuous commitment to maintain the coolant temperature within a desired range. Unlike discrete commitments, which are public, a continuous commitment might be visible only to the agent and is driven by the beliefs, desires, and intentions of the agent.

Each type of commitment can in turn be of two types. The first is the type that has a specific creditor, and these are what are typically thought of as commitments. The second type is when there is no specific creditor, and we call this

an *obligation*. An obligation might be viewed as a commitment or a promise that one makes to oneself or to *society*. A society serves as an abstract creditor, which has been modeled as a Sphere of Commitment (SoCom) [12]. Examples of continuous obligations are where one feels obligated to "honor your parents," "hold a door for the next person," "do not litter," and "protect the environment." Note that the potential actions involved in these might be positive (do something), negative (do not do something), or abstract (honor).

The two types of commitments are depicted in Figure 2. In this paper, we focus on formalizing discrete commitments.

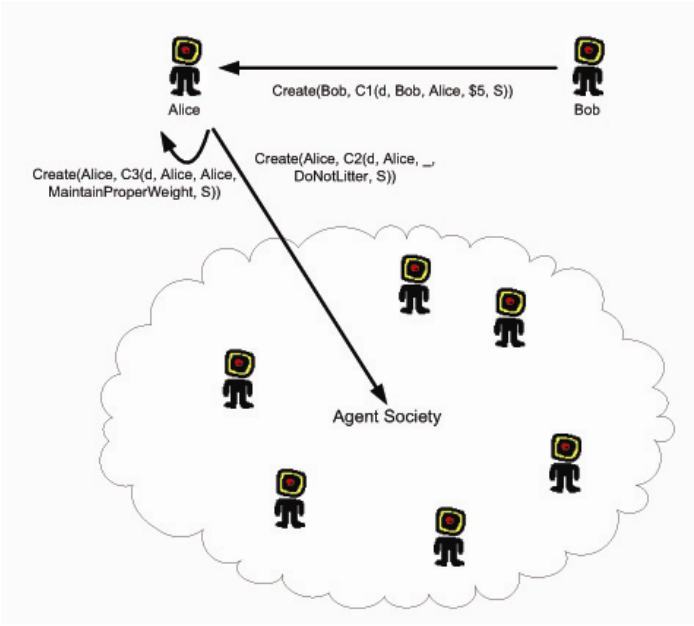

Fig. 2. The two major types of commitments are discrete ones between two agents and continuous ones between an agent and its society or itself

3 A Commitment-Driven Multiagent System

Our formalization considers a multiagent environment that is partially observable, stochastic, sequential, and dynamic. The environment is cooperative and consists of two classes of participating agents: service providers and service seekers. Service providers and service seekers associate or bind with each other via the binary relationship of commitments. In addition to these participating agents, there is a class of *nonparticipating agents* in the environment that behave as impartial arbiters. The arbiters provide the context to a commitment relationship, as a SoCom. Every agent in the environment is autonomous; hence, at any

point in time, a providing agent may choose to either abide by its commitment or stray from it. The arbiters can be used to capture a participating agent's behavior with regard to its commitments. Historical information about a participating agent's behavior can be utilized to measure its commitment adherence for future interactions, which is an area for further research.

Our formalization assumes that the participating agents have already identified each other and have already become part of a commitment relationship. How service seekers and service providers locate each other, how they identify compatible providers or seekers, how they interact or negotiate to form a binary commitment relationship, and what structure of communication and protocol they use are questions beyond the scope of our formalization.

It is further assumed that, in this commitment-driven cooperative environment, the partial view that an agent has is governed solely by the commitment relationships in which it participates. In other words, agents have knowledge of other agents with whom they are associated via commitment relationships. Furthermore, it is assumed that the knowledge about a commitment relationship is governed by commitment operations, i.e., an agent has knowledge about a commitment association only through operations that affect that commitment. For example, when a service-seeking agent and a service-providing agent participate in a commitment relationship, each will have knowledge of the other agent's commitment actions and each will have knowledge of when that commitment gets created, satisfied, canceled, etc. However, knowledge such as how that commitment is satisfied, why it was or was not satisfied, or why it was canceled is not available to the participating agents.

The typical environment for commitments is dynamic and nondeterministic, hence its temporal dimension is best represented as branching time. As the underlying temporal parameter moves forward, choices of actions by agents introduce branches, thus forming a tree. The following section describes in further detail this temporal structure as it relates to an agent's state of mind.

4 BDI in a Branching-Time CTL* Framework

In this section, we first restate Rao and Georgeff's BDI_{CTL} formalism, which is an extension of Emerson's Computation Tree Logic (CTL). In this formalism, the world is modeled with the help of an underlying branching temporal structure called a time tree, which has a single past and a branching time future, i.e., each moment on this infinite time tree may have many successor moments. It is assumed that along each path in this tree the corresponding timeline is isomorphic to $\mathbb{N}$. The maximal set of linearly ordered moments along a timeline makes a world and any point in a particular world is called a situation.

CTL operators are used to quantify over possible paths and states, and the temporal operators *A, E, X, U, F,* and *G* have their usual meanings (*A*: for all futures, *E*: for some futures, *X*: next, *U*: until, *F*: eventually, and *G*: always). BDI operators B, D, and I are used to represent the agent's internal state of mind.

4.1 Syntax and Semantics of BDI_{CTL*}

A particular point in a particular world is called a situation. A structure M with many such situations is a *Kripke* structure.

$$M = \langle S, R, B_a, D_a, I_a, L \rangle$$

where,

- S is a set of states.
- R is a binary relation $R \subseteq S \times S$.
- $L : S \rightarrow PowerSet(AtomicPropositions)$ is a labeling that associates with each state s an interpretation $L(s)$ of all atomic propositions at state s.

The relations B_a, D_a, and I_a map the agent's current situation to its belief, desire, and intention-accessible worlds. The structure M at a particular time point or moment m is denoted by M_m.

Assuming n agents, we define a set of admissible rules for States and Paths (true or false for states and paths) as follows:

State formulas:

(S1) Each atomic proposition is a state formula.
(S2) If α and β are state formulas, then so are $\alpha \wedge \beta$ and $\neg\alpha$.
(S3) If α is a path formula, then $E\alpha$ and $A\alpha$ are state formulas.
(S4) If α is a state formula,then $B_a(\alpha), D_a(\alpha), I_a(\alpha)$ are state formulas as well.

Path formulas:

(P1) Any state formula is also a path formula.
(P2) If α and β are path formulas, then so are $\alpha \wedge \beta$ and $\neg\alpha$.
(P3) If α is a path formula, then $X\alpha$ and $\alpha U \beta$ are path formulas.

A formula is interpreted with respect to a situation structure M. A *fullpath* x is an infinite sequence $s_0, s_1, s_2, ...$ of states, such that $\forall i (s_i, s_{i+1}) \in R$. A *suffix path* x^i is an infinite sequence $s_i, s_{i+1}, s_{i+2}, ...$ of states. We write $M \models_{m_0} p$ to mean that state formula p is true in structure M at moment m_0. We write $M \models_x p$ to mean that path formula p is true in structure M at fullpath x. $B_a(\alpha), D_a(\alpha)$, and $I_a(\alpha)$ are beliefs, desires and intentions of agent a about α.

(S1) $M \models_{m_0} p$ iff $p \in L(m_0)$,
(S2) $M \models_{m_0} p \wedge q$ iff $M \models_{m_0} p$ and $M \models_{m_0} q$,
$M \models_{m_0} \neg p$ iff not $(M \models_{m_0} p)$,
(S3) $M \models_{m_0} Ep$ iff $\exists$ fullpath $x = (m_0, m_1, m_2, ...)$ in M that $M \models_{m_0} p$,
$M \models_{m_0} Ap$ iff $\forall$ fullpath $x = (m_0, m_1, m_2, ...)$ in M that $M \models_{m_0} p$,
(S4) $M, m_0 \models B_a(\alpha)$ iff $\forall m_1 \in S$ and $m_0 R_1 m_1, M, m_1 \models \alpha$ where 'R_1' is the accessibility relation.
(S5) $M, m_0 \models D_a(\alpha)$ iff $\forall m_1 \in S$ and $m_0 R_2 m_1, M, m_1 \models \alpha$ where 'R_2' is the accessibility relation.

(S6) $M, m_0 \models I_a(\alpha)$ iff $\forall m_1 \in S$ and $m_0 R_3 m_1, M, m_1 \models \alpha$ where 'R_3' is the accessibility relation.

(P1) $M \models_x p$ iff $M \models_{s_0} p$;
(P2) $M \models_x p \wedge q$ iff $M \models_x p$ and $M \models_x q$,
$M \models_x \neg p$ iff not$(M \models_x p)$;
(P3) $M \models_x pUq$ iff $\exists i[M \models_{x^i} q$ and $\forall j(j < i$ implies, $M \models_{x^j} p)]$

5 Commitments

Now that we have described our multiagent environment and the state of mind of its participating agents, we define commitments and the operations that the agents can perform on them. For this purpose, we briefly revisit Singh and Huhns's formalism of commitments [12] and extend the commitment properties and operations defined therein.

Our formalism considers social commitments that are legal abstractions associating one agent with another. Earlier works have described another class of commitments that are personal or internal to an agent and do not bind two separate agents. However, such unitary internal commitments are not relevant to our cooperative environment, which is driven solely by binary relationships between the agents. Commitments are accessible publicly and they represent an interaction between two participating agents. For example, service level agreements, online purchases, and service contracts are all real-world instances of commitments.

As per the commitment formalism developed by Singh and Huhns[12], the following are three key properties of commitments:

1. **Multiagency:** Commitments associate one agent with another. The agent that promises or commits to satisfying a condition is called the debtor agent and the other agent that wants the condition to be satisfied by the debtor is called the creditor agent. Each commitment is directed from its debtor to its creditor.
2. **Scope:** Commitments have a well-defined scope, which gives context to the commitment. A scope can be directed by a separate third-party organization (Sphere of Commitment: SoCom).
3. **Manipulability:** Commitments are modifiable. They can be satisfied, breached, or canceled.

We extend these properties by defining two additional ones:

1. **Lifetime:** Commitments have a lifetime; they are created, they live (remain active), and at some point they cease to exist. Continuous commitments are beyond the scope of this paper and a subject of future research.
2. **Degree:** When active, commitments do not necessarily remain in one constant state; in real situations, at the time when people make commitments, they intend to fulfill them. But situations change and the priorities of commitments might thus change. This is captured by a *degree of commitment*.

For a service-oriented environment, the degree of commitment changes with changing beliefs, desires, and intentions.

As an example, let us consider a travel agent who has a commitment to sell n tickets for airline A. If another airline (airline B) slashes their ticket prices and the customers want to buy those tickets, the travel agent reorders his commitments to satisfy his customers. Though he is still committed to A, his priority changes to selling airline B's tickets. Likewise, an individual agent can order any new commitment that he creates using a partial order. Anytime a change in his beliefs, desires, or intentions results in a change in preferences, he could reorder his commitments, thus mimicking the real life situation. The ordering method used would be dependent on the system.

Also, in the case of commitment cancelation or revocation, the commitment might not change from an active state to an inactive state instantaneously; instead, it might gradually decline in degree until it becomes inactive. This area is also a subject for future research.

5.1 Structure of Commitments

Commitments are represented by a predicate C and have the form $C(d, a, b, p, S, \delta)$, where

d: is a unique identifier,
***a*:** is the debtor agent,
***b*:** is the creditor agent,
***p*:** is the promise or the condition that the debtor will bring about,
***S*:** is the context, also known as the *sphere of commitment*, and
δ **:** is the degree of commitment.

For the sake of simplicity herein, we ignore δ.

5.2 Operations on Commitments

Our cooperative environment is commitment-driven and we assume the participating agents' knowledge is governed solely by commitment operations. Here we describe commitment operations as defined by [12, 6], where commitments are treated as abstract data types that associate a debtor, creditor, promise, and context. The six fundamental commitment operations are

1. *Create (a, C(d, a, b, p, S))*
2. *Discharge(a, C(d, a, b, p, S))*
3. *Cancel(a, C(d, a, b, p, S))*
4. *Release(b, C(d, a, b, p, S))*
5. *Assign(b, c, C(d, a, b, p, S))*
6. *Delegate(a, c, C(d,a, b, p, S))*

We use predicates to describe whether the commitment C has been satisfied, canceled, breached, or still holds, and these predicates will be written as *satisfied(C), canceled(C), breached(C), and active(C)*, respectively [7].

6 Commitment Formalization in BDI+CTL*

In this section we present our formalization that represents a combination of BDI and commitments. Multiagent associations are bound by commitments and each agent's knowledge of those commitments is through commitment operations. Informally, a commitment between two agents comes about through interactions (and often negotiations) between the agents, so both agents are necessarily aware of and believe in the commitment.

Definition 6.1: Creating a Commitment,
Create(a, C(d,a,b,p,S))

1. For all paths, *Agent a* believes that from the next moment onwards commitment C will be active until it is either satisfied or breached or canceled.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AB_a((XG(active(C)))U$
   $(satisfied(C) \vee breached(C) \vee canceled(C)))$
2. For all paths, *Agent a* believes that commitment C will eventually be satisfied.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AB_aF(satisfied(C))$
3. For all paths from the next moment onwards, *Agent a* intends the commitment C until it is either satisfied or breached or canceled.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AXG((I_a(C))U$
   $(satisfied(C) \vee breached(C) \vee canceled(C)))$
4. For all paths, *Agent a* believes that from the next moment onwards *Agent b* desires commitment C until it is either satisfied or canceled.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AB_a((XG(D_b(C)))U$
   $(satisfied(C) \vee canceled(C)))$
5. For all paths, *Agent b* believes that from the next moment onwards commitment C will be active until it is either satisfied or breached or canceled.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AB_b((XG(active(C)))U$
   $(satisfied(C) \vee breached(C) \vee canceled(C)))$
6. For all paths, *Agent b* believes that from the next moment onwards *Agent a* intends commitment C until it is either satisfied or breached or canceled.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AB_b((XG(I_a(C)))U$
   $(satisfied(C) \vee breached(C) \vee canceled(C)))$
7. For all paths, *Agent b* believes that commitment C will eventually be satisfied.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AB_bF(satisfied(C))$
8. For all paths from the next moment onwards, *Agent b* desires commitment C until it becomes inactive.
   $M \models_m Create(a, C(d, a, b, p, S)) \Rightarrow AXG((D_b(C))U(\neg active(C)))$

Note that *agent b* can not intend C to be satisfied, because b does not have any control over C and cannot force it.

Definitions of other commitment operations can be written similarly.

Definition 6.2: Revoking a Commitment,
Cancel(a, C(d,a,b,p,S))

1. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg active(C)))$
2. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AB_aF(\neg satisfied(C))$
3. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AXG(\neg I_a(C))$
4. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AB_b(XG(\neg active(C)))$
5. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AB_bF(\neg satisfied(C))$
6. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AXG(\neg(D_b(C)))$
7. $M \models_m Cancel(a, C(d, a, b, p, S)) \Rightarrow AB_b(XG(\neg I_a(C)))$

Definition 6.3: Discharging a Commitment,
Discharge(a, C(d,a,b,p,S))

1. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AB_a(XG(satisfied(C)))$
2. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg active(C)))$
3. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AXG(\neg(I_a(C)))$
4. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg(D_b(C))))$
5. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AB_b(XG(satisfied(C)))$
6. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg active(C)))$
7. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AXG(\neg(D_b(C)))$
8. $M \models_m Discharge(a, C(d, a, b, p, S)) \Rightarrow AB_b(XG(\neg(I_a(C))))$

Definition 6.4: Releasing a Commitment,
Release(b, C(d,a,b,p,S))

1. $M \models_m Release(b, C(d, a, b, p, S)) \Rightarrow AXG(\neg(D_b(C)))$
2. $M \models_m Release(b, C(d, a, b, p, S)) \Rightarrow AB_b(XG(\neg active(C)))$
3. $M \models_m Release(b, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg(I_a(C))))$
4. $M \models_m Release(b, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg(D_b(C))))$
5. $M \models_m Release(b, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg active(C)))$
6. $M \models_m Release(b, C(d, a, b, p, S)) \Rightarrow AXG(\neg(I_a(C)))$

Definition 6.5: Assigning a Commitment,
Assign(b, c, C(d,a,b,p,S))

1. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AXG(\neg(D_b(C)))$
2. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow B_bA(XG(\neg active(C)))$
3. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_b(XG(D_c(C)))$
4. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_b(XG(I_a(C)))$
5. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_c((XG(active(C)))U$
   $(satisfied(C) \vee breached(C) \vee canceled(C)))$
6. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_c((XG(I_a(C)))U$
   $(satisfied(C) \vee breached(C) \vee canceled(C)))$
7. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_cF(satisfied(C))$
8. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AXG((D_c(C))U$
   $(satisfied(C) \vee canceled(C)))$
9. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_a(XG(D_c(C)))$
10. $M \models_m Assign(b, c, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg D_b(C)))$

Definition 6.6: Delegating a Commitment, *Delegate(a, c, C(d,a,b,p,S))*

1. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AXG(\neg I_a(C))$
2. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_a(XG(I_c(C)))$
3. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_a(XG(\neg active(C)))$
4. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_cF(satisfied(C))$
5. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_c((XG(active(C)))$ U $(satisfied(C) \vee breached(C) \vee canceled(C)))$
6. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AXG((I_c(C))U$ $(satisfied(C) \vee canceled(C)))$
7. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_a(XG(D_b(C)))$
8. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_b((XG(I_c(C)))U(satisfied(C) \vee$ $breached(C) \vee canceled(C)))$
9. $M \models_m Delegate(a, c, C(d, a, b, p, S)) \Rightarrow AB_bF(satisfied(C))$

Theorem 1 *The debtor and creditor will never end up believing a commitment is still active after it has been discharged.*

Proof When a commitment C is discharged, from definition 5.3.2, *Agent a* (debtor) believes that C will be inactive globally from the next moment onwards. From definition 5.3.6, *Agent b* (creditor) believes that C is inactive from the next moment onwards.

Similar proofs can be given when a commitment is canceled or released.

7 Example Uses of the BDI Commitment Formalism

We present examples of how our formalization can be used to explain, interpret, and model real-world multiagent systems. We use the travel agent example presented by Xing and Singh [13], where a customer contacts her travel agent to book a trip to a city with many hotels and airports. The travel agent requests airline and hotel clerks to make appropriate reservations and send confirmations to the traveler. The customer, travel agent, airline agent, and the hotel agent are all autonomous entities (*persons or their representative agents*).

When a customer contacts the travel agent to book a trip, the travel agent creates a commitment. Per *definition 5.1.2*, the travel agent `believes` that such a commitment will eventually be satisfied. Similarly, the customer `believes` that the travel agent's commitment will be satisfied eventually, which is consistent with *definition 5.1.7*. Also, per *definitions 5.1.3* and *5.1.8*, the travel agent `intends` to satisfy its commitment and the customer `desires` for that commitment to be satisfied. When the reservations are made and the customer is satisfied, the commitment is discharged. In accord with *definition 5.3.5*, the travel agent now `believes` it has satisfied its commitment, which becomes inactive. The customer, per *definition 5.3.7*, no longer `desires` for the trip to be booked again (unless he initiates a new instance of a trip-booking commitment).

Carrying this example further, consider a scenario where a customer assigns its commitment to another agent. Per *definitions 5.5.1* and *5.5.2*, we can see that the customer does not `desire` the travel agent to book a trip for him. Instead, the customer `believes` that the agent to whom the commitment was assigned `desires` that trip, which is explained by *definition 5.5.3*.

Consider the example from [6] of a travel agent who wishes to book an airline ticket to a certain destination, a rental car to use while there, and a hotel room in which to stay. The four scenarios discussed in [6] are (1) the travel agent wanting the passenger to fly on a particular day while still reserving the right to choose any flight on that day, (2) the car rental company offering a one-week free rental at a later time, (3) a hotel offering an electronic discount coupon that expires today, but text on the coupon states that it can only be used during a future spring break, and (4) the car rental company offering a warranty that cannot be used during the period in which the warranty is valid. The first two scenarios can be implemented directly with our formalism, as any conditions in such scenarios can be specified as the condition p in our commitment structure.

When there is a violation of time constraints similar to scenarios three and four, temporal operators in our condition p can capture the time constraints and show that the commitment cannot be satisfied. Our CTL* framework takes care of all the time constraints in a commitment and the BDI architecture captures the commitments in terms of the states of mind of the participating agents.

When an agent has more than one commitment, the only link or relationship between them has to be through the agent's mind. These commitments need to be consistent with the agents's internal beliefs, which our formalization helps to achieve. BDI can also be used to determine which of the several commitments an agent does first. For example,

Alice commits to pay *Bob* $5

Bob commits to pay *Joe* $5

Because *Bob* believes it will get $5 from *Alice*, it then can form an intention to honor its commitment to *Joe*. The BDI system allows the agents to contend with multiple simultaneous commitments in the real world. As an example, if the beliefs and desires are not included in the model of the travel agent transaction scenario, the seller agent simply makes a commitment to the buyer agent based upon the ticket availability information it has received from the airlines.

Since desire is not modeled, the buyer and seller cannot barter, as they will make their commitments based purely upon the availability of tickets and money. If the seller has pricing from a single airline, it will make the commitment to sell, and the buyer will make the commitment to buy no matter what the price is as long as he has the money. The seller is not going to call other airlines, and the buyer is not going to call other travel agents due to lack of desires. Thus a better deal for both is not possible here.

If the desire of an agent is modeled, then the buyer's desire is to obtain the cheapest price and this will make it change its commitment to the seller. The commitment here is no longer "I will pay you whatever price you want," but it could be "I will buy the ticket only if it is the cheapest price."

On the other hand, the Seller's desire is to get the best price for the ticket, and the seller's commitment will be "I will you sell you the ticket only if you agree to pay cash within 24 hours and for credit I will charge you 10% extra." The seller has modified his commitment, because he has recognized the implicit threat in the buyer's commitment. The buyer has the desire to get the cheapest ticket as demonstrated by its commitment. If the seller does not modify its commitment, the buyer could go elsewhere. At the same time, to safeguard against a defaulted payment from the buyer and still make a profit, the seller is willing to offer or match the price if cash is paid.

Applying our formalization to the example, *Alice* commits to pay *Bob* \$5 means that *Alice* creates a commitment

Create(Alice, C(1,Alice,Bob,pay(\$5),S))
such that
1. *Alice* and *Bob* believe that commitment C will be active until it is either satisfied or breached or canceled or suspended.
$AB_{Alice}((XG(active(C)))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$;
$AB_{Bob}((XG(active(C)))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$
2. For all paths, *Alice* and *Bob* believe that commitment C will eventually be satisfied.
$AB_{Alice}F(satisfied(C))$; $AB_{Bob}F(satisfied(C))$
3. For all paths from the next moment onwards, *Alice* intends the commitment C until it is either satisfied or breached or canceled or suspended.
$AXG((I_{Alice}(C))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$
4. For all paths, *Alice* believes that from the next moment onwards *Bob* desires commitment C until it is either satisfied or canceled .
$AB_{Alice}((XG(D_{Bob}(C)))U(satisfied(C) \vee canceled(C)))$
5. For all paths, *Bob* believes that from the next moment onwards *Alice* intends commitment C until it is either satisfied or breached or canceled or suspended.
$AB_{Bob}((XG(I_{Alice}(C)))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$
6. For all paths from the next moment onwards, *Bob* desires commitment C until it becomes inactive.
$AXG((D_{Bob}(C))U(\neg active(C)))$
When *Bob* commits to pay *Joe* \$5, *Bob* creates a commitment

Create (Bob, C(2, Bob, Joe, pay(\$5), S))
such that
1. *Bob* and *Joe* believe that commitment C will be active until it is either satisfied or breached or canceled or suspended.
$AB_{Bob}((XG(active(C)))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$;
$AB_{Joe}((XG(active(C)))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$
2. For all paths, *Bob* and *Joe* believe that commitment C will eventually be satisfied.
$AB_{Bob}F(satisfied(C))$; $AB_{Joe}F(satisfied(C))$
3. For all paths from the next moment onwards, *Bob* intends the commitment C until it is either satisfied or breached or canceled or suspended.
$AXG((I_{Bob}(C))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$

4. For all paths, *Bob* believes that from the next moment onwards *Joe* desires commitment C until it is either satisfied or canceled.
$AB_{Bob}((XG(D_{Joe}(C)))U(satisfied(C) \vee canceled(C)))$
5. For all paths, *Joe* believes that from the next moment onwards *Bob* intends commitment C until it is either satisfied or breached or canceled or suspended.
$AB_{Joe}((XG(I_{Bob}(C)))U(satisfied(C) \vee breached(C) \vee canceled(C) \vee suspended(C)))$
6. For all paths from the next moment onwards, *Joe* desires commitment C until it becomes inactive.
$AXG((D_{Joe}(C))U(\neg active(C)))$

Bob believes that commitment 1 will be satisfied eventually and he will get the \$5. He intends to get the money from *Alice* and pay that to *Joe* and thus satisfy commitment 2 (to pay *Joe* \$5). Using this formalization, the system can have rules, dependent on its requirements and available resources, to represent these intentions. As a simple example, *Bob* can have a rule such as
$ReceiveMoney(Alice, \$5) \Rightarrow PayMoney(Joe, \$5)$ (When money is received from *Alice*, pay that to *Joe*).

The above examples demonstrate how our formalization can be utilized to understand, explain, interpret, and model a real-world, commitment-centric, multiagent system. Our formalization is an improvement over the temporal logic approaches in [6, 13], since it bridges BDI architectures and commitments.

8 Conclusion and Future Directions

Many real world systems are becoming cooperative. In a cooperative multiagent system, commitments represent agent associations and interactions, and a participant agent's beliefs, desires, and intentions about the commitments in which it is involved are critical to modeling agent behavior. With this formalization of commitments in terms of an agent's beliefs, desires, and intentions, we have provided the basic framework on which a more comprehensive commitment-driven decision theory can be developed. The advantage of this theoretical framework is that it blends two very robust and widely accepted theoretical frameworks that together can be utilized to model a cooperative multiagent system. These two frameworks are BDI_{CTL*} and commitments.

Our future research involves exploration of continuous commitments, how agents decide what to commit (earlier works on "capability" [8] can be integrated with commitments), when to cancel a commitment, how does a commitment "age," degree of commitment, and how can historical information of an agent's commitment adherence be utilized to predict its behavior. Commitment adherence and Sphere of Commitment can also be tied to *trust*. Moreover, with the help of either utility models or probabilities, a more comprehensive *commitment-driven decision theory* can be developed to model a cooperative multiagent environment expressively.

References

1. C. Castelfranchi, Commitments: From Individual Intentions to Groups and Organisations, Proceedings of the Int. Conf. on Multi-Agent Systems'96 (1996)
2. Philip R. Cohen and Hector J. Levesque, Intention Is Choice with Commitment, Artificial Intelligence, Volume 42 , Issue 2-3, 213 - 261 (1990)
3. E. A. Emerson, Handbook of Theoretical Computer Science: Formal Models and Semantics, 995 - 1072 MIT Press Cambridge, MA, USA (1991)
4. Maria Fasli, On Commitments, roles and obligations, Central and Eastern European Conference on Multi-Agent Systems, 93-102 (2001)
5. N. R. Jennings, Commitments and Conventions: The Foundation of Coordination in Multi-Agent Systems, The Knowledge Engineering Review Volume 8, Number 3, 223–250 (1993)
6. Ashok U. Mallya and Michael N. Huhns, Commitments Among Agents, IEEE Internet Computing Volume 7, Issue 4, 90 - 93 (2003)
7. Ashok U. Mallya and Munindar P. Singh, An Algebra for Commitment Protocols, Autonomous Agents and Multi-Agent Systems, Volume 14 , Issue 2, 143 - 163. (2007)
8. Lin Padgham and Patrick Lambrix, Agent Capabilities: Extending BDI Theory, AAAI/IAAI, 68-73 (2000)
9. Anand S. Rao and Michael P. Georgeff, Modeling Rational Agents within a BDI-Architecture, Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, 473–484 (1991)
10. Munindar P. Singh, Commitments Among Autonomous Agents in Information-Rich Environments, Modelling Autonomous Agents in a Multi-Agent World ,141-155 (1997)
11. Munindar P. Singh, An Ontology for Commitments in Multiagent Systems: Toward a Unification of Normative Concepts, Artificial Intelligence and Law, Volume 7, Number 1, 97-113 (1999)
12. Munindar P. Singh and Michael N. Huhns, Service-Oriented Computing: Semantics, Processes, Agents,363-370, Wiley, London, UK (2005)
13. Jie Xing and Munindar P. Singh, Engineering Commitment-Based Multiagent Systems: A Temporal Logic Approach, Proceedings of the second international joint conference on Autonomous agents and multiagent systems, 891 - 898 (2003)