Chapter 2

A Framework for Specifying Group Decision-Making Mechanisms

Agents collaborating on a group activity need to make decisions about how they will carry out that activity. People make group decisions using both informal and formal mechanisms. For instance, people collaborating on projects such as putting a swing set together or taking a cross-country trip may informally debate and negotiate about the best way to do their activity. In contrast, the bylaws of a corporation may specify detailed mechanisms for certain corporate decisions. Although people can be quite adept at engaging in free-form group decision-making, computer agents generally are not. This chapter presents a framework for specifying group decision-making mechanisms that groups of computer agents can use to coordinate their group decision-making activity.

Throughout this chapter, we will use the example of our planning to go to the movies together. We need to make various planning decisions. The content of such decisions can be expressed in terms of intention-update operations. For example, if we decide to see *The Wizard of Oz*, we need to update our intentions accordingly: instead of intending that we see *some* movie, each of us now intends that we see *The Wizard of Oz*. Next, we may need to decide which theater to attend, and so on. Thus, agents make group decisions to coordinate the planning and eventual execution of the group activity to which they are committed. Decisions made in the context of existing intentions are called *context-bound* decisions.

Agents must also be able to make decisions outside of any pre-existing context. For example, prior to being committed to working together, a group of agents need to be able to decide whether to *establish* such a commitment in the first place. Such decisions are called *context-free* decisions. For example, our initial decision to commit to seeing a movie together was a context-free decision. Context-free decisions establish not only a group's commitment to a new activity, but also their commitment to doing that activity *collaboratively*. In other words, in collaborative activity, agents must be committed not only to *doing* the activity together, but also to *planning* the activity together.

To plan their activity together, agents must be able to make decisions together. They must be able to generate group decisions. The purpose of a group decision-making mechanism is to provide computer agents with a well-defined, reliable means of generating group decisions. This chapter presents a framework for rigorously specifying group decision-making mechanisms (GDMMs) that agents can use to coordinate their planning activity.

The GDMM framework can be used to specify a wide variety of group decision-making mechanisms. In a proposal-based mechanism, agents make proposals, vote on proposals and announce group decisions. In an auction-based mechanism, agents invoke auctions, submit bids, and announce auction results. The purpose of rigorously specifying a group decision-making mechanism is to allow important properties of the mechanism to be proven and to make explicit the assumptions required to ensure that those properties hold.

The mechanisms specified according to the GDMM framework may be used to generate context-free decisions—those that establish a group's initial commitment to some new activity—or context-bound decisions—those used to coordinate an activity to which the group is already committed.

To illustrate the use of the GDMM framework, a sample proposal-based mechanism is used as a running example throughout this chapter. The chapter concludes with a formal analysis of the properties of the sample, proposal-based mechanism. Chapter 3 uses the GDMM framework to specify a mechanism based on a combinatorial auction for solving the Initial-Commitment Decision Problem.

2.1 Background

The group decision-making mechanisms specified according to the GDMM framework make use of declarative speech-acts, intention-update operations and social obligations.

2.1.1 Declarative Speech-Acts

Austin (1962) observed that in certain circumstances, merely saying something can make it so. For example, a Justice of the Peace can, in certain circumstances, cause a couple to become married, merely by declaring them to be married. Declarative speech-acts—not to be confused with declarative *sentences* such as "The sky is blue"—are unique among speech-acts in that, under appropriate circumstances, they have the power to make their propositional content true merely by their being uttered (Searle, 1995).

It is exclusively through declarative speech-acts that agents participate in the group decision-making mechanisms specified according to the GDMM framework. For example, in the proposal-based mechanism, agents, in effect, make declarations such as, "I declare that I have made a proposal", "I declare that I have voted to accept your proposal", or "I declare that the group has made a decision to accept my proposal." Similarly, in the auction-based mechanism, agents make declarations such as, "I declare that I have submitted a bid for tasks A, B and C", or "I declare that your bid has been awarded."

Declarative speech-acts have the power to establish the truth of their propositional content only if certain conditions apply. In general, these conditions, which are called *authorization conditions*, depend on the type of propositional content. For example, the authorization conditions for a declaration of marriage are usually not the same as the authorization conditions for a declaration of war. In the GDMM framework, a specification for a group decision-making mechanism includes not only the classes of allowable propositional content (e.g., proposals, votes, announcements), but also the authorization conditions for each such class. In the formal specification of authorization conditions for declarative speech-acts, this chapter draws from the approach taken by Dignum and Weigand (1995a; 1995b).

2.1.2 Intention-Update Operations

If I intend to go to the movies sometime next week, I need to make certain planning decisions. For example, I need to pick a movie to see and a theater to see it in. If I decide to see *The Wizard of Oz*, I must update my intention. No longer do I intend to see any old movie; I intend to see *The Wizard of Oz*. Although agents collaborating on a group activity need to be careful about *how* they make their planning decisions, the result of each such planning decision is always some kind of intention-update operation.[1]

The formalization in this chapter presumes a set of intention-update operations that agents can use to update their intentions in the context of single-agent or group activity. For example, that act of updating the content of an intention identified by the constant I to reflect the binding of a parameter ?P to the value v can be represented by the following action expression:[2]

$$Update(I, \texttt{Bind}(\texttt{?P}, \texttt{v})).$$

In general, if υ is an intention-update act-type and $a_1, \ldots, a_n$ is a set of arguments appropriate for υ, then $Update(I, \upsilon(a_1, \ldots, a_n))$ represents an intention-update operation to be applied to the intention identified by I.

In addition to intention-update operations, this chapter presumes an intention-adoption act-type, $AdoptInt$, where $AdoptInt(I, \alpha, GR, \Upsilon)$ represents the (single-agent) action of adopting an intention to be associated with the identifier I, concerning the doing of an action of type α by the group GR; in the case of single-agent activity, the group GR is a singleton. The optional fourth argument, Υ, represents an ordered list of intention-update operations (e.g., a set of temporal constraints) to be applied to the new intention immediately upon its being established.

[1]There are some kinds of intention-update operations (e.g., selecting an agent or subgroup to do some constituent task) that only make sense in the context of group activity.

[2]Ortiz (1999) defines a generic $Update$ operator that can accommodate arbitrary intention updates in Hypothetical Logic, but does not explicitly categorize intention-update operations as described here.

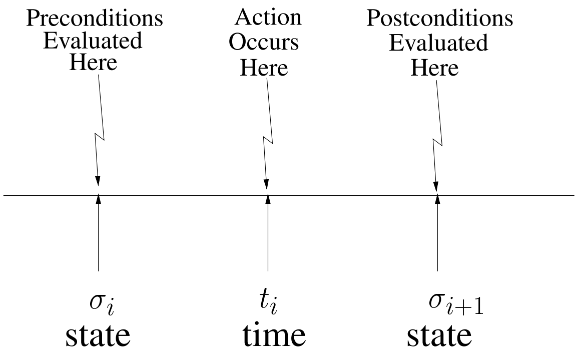

Figure 2.1: The semantics of DDLTLB

2.1.3 Social Obligations

The mechanisms specified by the GDMM framework establish group decisions in terms of *social obligations*.[3] A social obligation has the form

$$\mathcal{O}(G, GR, X),$$

which represents that agent G is obliged toward the group GR to do the action X. This thesis restricts attention to social obligations for which the obliged action is a (single-agent) mental action, either to adopt a new intention or to update the contents of an existing intention.

2.1.4 Dynamic Deontic Linear Time Temporal Logic (DDLTLB)

This chapter uses Dynamic Deontic Linear Time Temporal Logic (DDLTLB), developed by Dignum and Kuiper (1997), to represent speech-acts (in dynamic logic) and social obligations (in deontic logic) within a temporal framework.[4] The semantics of DDLTLB allows dynamic operators to be combined with temporal operators by evaluating the preconditions at a state σ_i that can be arbitrarily close to, but *before* the time t_i that an action is done, and the postconditions at a state σ_{i+1} that can be arbitrarily close to, but *after* t_i, as illustrated in Figure 2.1.

Consider the following typical expression from dynamic logic: $[A]\phi$, which represents that the doing of action A would result in ϕ holding. If this expression evaluates to `True` at state σ_i, then the doing of A at time t_i would ensure that ϕ evaluates to true at state σ_{i+1}, as illustrated in Figure 2.2.

---

[3] Singh (1991; 1996; 1999; 2000) and Castelfranchi (1995) refer to what we are calling social obligations as "social commitments". Royakkers and Dignum (1999) use expressions of the form $COMM(i, X, \beta)$ "to stand for 'agent i commits to the group X to perform action β'." Cohen and Levesque (1990), Levesque, Cohen and Nunes (1990) and Cohen, Levesque and Smith (1997) discuss joint commitments of agents engaged in group activity.

[4] The "B" in DDLTLB indicates that the temporal logic has "both" past and future tenses.

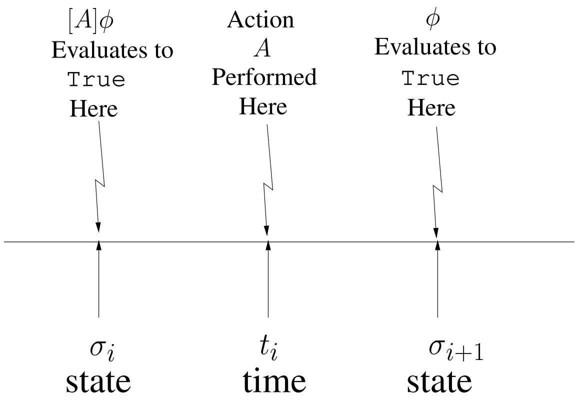

Figure 2.2: The semantics of $[A]\phi$ in DDLTLB

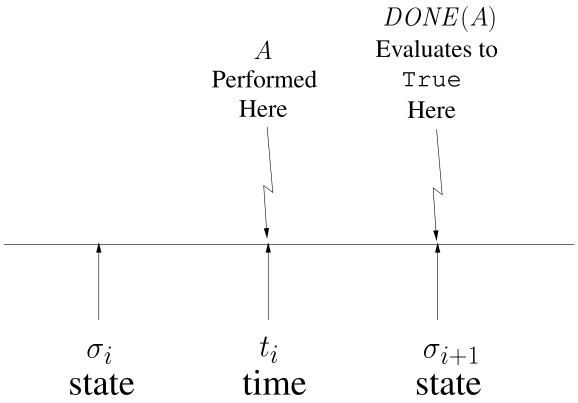

Figure 2.3: The semantics of $DONE$ in DDLTLB

DDLTLB also includes a $DONE$ operator used to represent that an action was just done. For example, the expression $DONE(A)$ would evaluate to `True` at the state σ_{i+1} if the action A was done at the time-point t_i, as illustrated in Figure 2.3.

Temporal Operators. DDLTLB includes many temporal operators, but only three are used in this chapter: $\mathcal{P}$, $\Diamond^-$ and $\Box$. Expressions of the form $\mathcal{P}\phi$ represent that ϕ held at the *previous state*. In other words, $\mathcal{P}\phi$ evaluates to `True` at the state σ_{i+1} if ϕ evaluates to `True` at the state σ_i, as illustrated in Figure 2.4. Similarly, expressions of the form $\Diamond^-\phi$ represent that ϕ held at some time in the (possibly distant) past. This chapter also uses the following abbreviation:

$$\Diamond^{\leftarrow}\phi \quad \equiv \quad \phi \vee \Diamond^-\phi,$$

which represents that ϕ holds now or held at some point in the past. Finally, the definition of the coordinated-cultivation requirement (in Section 2.4) uses propositions of the form, $\Box\phi$, representing that ϕ holds now and at all times in the future.

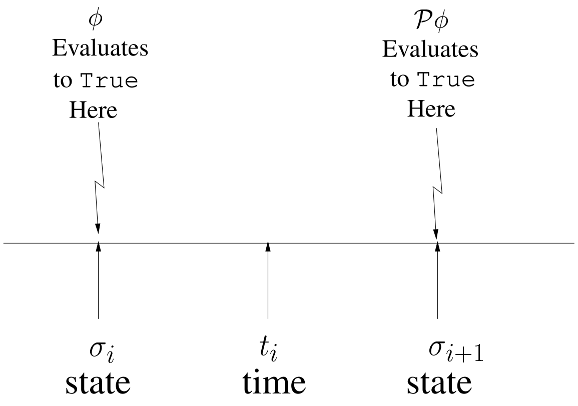

Figure 2.4: The semantics of $\mathcal{P}$ in DDLTLB

The Obligation Operator. In DDLTLB, the obligation operator $\mathcal{O}$ is applied to propositions. For example, $\mathcal{O}(\phi)$ represents that ϕ is obliged to hold in the current state. Our application of obligation operators to actions can easily be accommodated, for example, by the following translation:

$$\mathcal{O}(A) \mapsto \mathcal{O}(DONE(A)),$$

where $DONE$ is the DDLTLB operator seen earlier. Furthermore, for each agent G and each group GR, the construction $\mathcal{O}(G, GR, \ldots)$ may be viewed as a distinct social-obligation operator.

2.2 Specifying a Group Decision-Making Mechanism

In the GDMM framework, the specification of a group decision-making mechanism includes:

- a specification of the allowable *content* of agent declarations (e.g., "that a proposal has been made" or "that a bid has been submitted"); and
- a specification of the authorization conditions for each type of allowable content (e.g., "an agent can only vote to accept a proposal that it has not already rejected" or "an agent can announce a group decision to accept a proposal only if every other agent voted to accept that proposal").

For a mechanism to generate group decisions, it must be possible for some combination of declarations to establish conditions that authorize some agent to declare on behalf of the group that the group has made a decision.

2.2.1 Declarations in the GDMM Framework

Agent inputs to a group decision-making mechanism are restricted to declarations. The representation of declarative speech-acts presented in this section draws from Dignum and

Weigand (1995a; 1995b).

The declaration of an agent G to a group of agents GR that a proposition ϕ holds is represented by the following action expression:

$$\delta(G, GR, \phi).$$

For convenience, we define the following abbreviation:

$$declared(G, GR, \phi) \equiv DONE(G, \delta(G, GR, \phi)).$$

We assume that the speaker of a declaration is always one of the hearers.

$$\models declared(G, GR, \phi) \Rightarrow (G \in GR)$$

However, in the case of one-on-one communication, we may write $\delta(G_1, G_2, \phi)$ instead of the more cumbersome $\delta(G_1, \{G_1, G_2\}, \phi)$.

Axiom 2.1 *The declaration of a conjunction is equivalent to the corresponding conjunction of simultaneous declarations.*[5]

$$\models declared(G, GR, \wedge_i \phi_i) \equiv \wedge_i \; declared(G, GR, \phi_i)$$

2.2.2 Authorization Conditions

In the GDMM framework, following the approach taken by Dignum and Weigand (1995a; 1995b), the "right circumstances" for making a declaration are specified in terms of "authorizing conditions." However, in the context of group activity, there is always an authorizing agent or group; thus, authorizing conditions for the GDMM framework include the authorizing party as an argument. The predicate

$$auth(G, GR, D)$$

represents that agent G is authorized by group GR to do action D, where D represents a declarative speech-act.

Axiom 2.2 shows how an authorization predicate defines the conditions under which a declaration establishes the truth of its propositional content, as illustrated in Figure 2.5.

Axiom 2.2

$$\models auth(G, GR, \delta(G, GR, \phi)) \Rightarrow [\delta(G, GR, \phi)]\, \phi$$

Notice that the intended interpretation of $[\delta(G, GR, \phi)]\, \phi$ in the above axiom is: if the agent G declares to the group GR that ϕ holds, then ϕ holds.

[5]Dignum and Weigand (1995a) make a similar assumption.

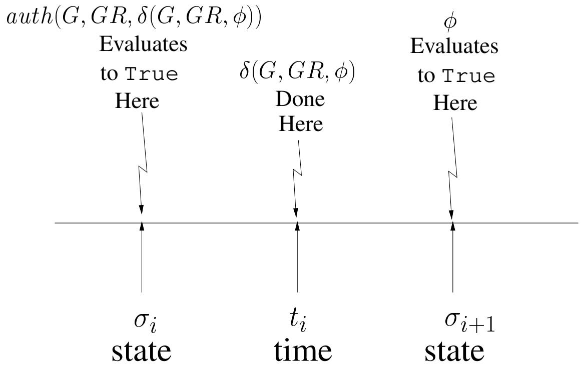

Figure 2.5: Authorization conditions for a declaration

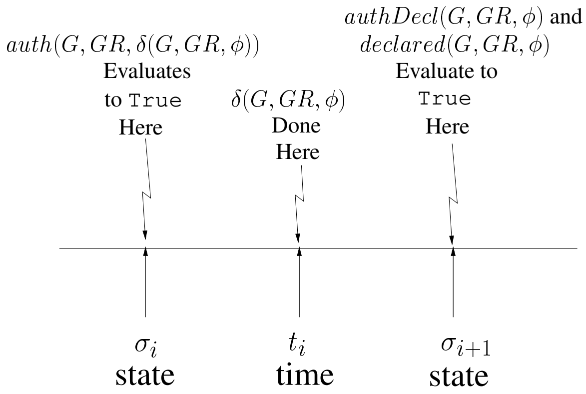

Figure 2.6: An authorized declaration

For convenience, we define an abbreviation for the occurrence of an authorized declaration.

$$authDecl(G, GR, \phi) \equiv \begin{cases} \mathcal{P}\ auth(G, GR, \delta(G, GR, \phi)) \\ \wedge \\ declared(G, GR, \phi) \end{cases}$$

where $\mathcal{P}$ is the "previous" operator described earlier. The semantics for an authorized declaration is illustrated in Figure 2.6.

2.2.3 Allowable content of agent declarations in a GDMM

The specification of a GDMM includes a specification of the classes of allowable content for declarations. For example, the content of declarations in the proposal-based mechanism is restricted to formulas involving special predicates such as $MadeCFP$, $AcceptedCFP$, and $RejectedCFP$. The following axiom stipulates that instances of such predicates hold only as a result of an authorized declaration.

Axiom 2.3 *If* $\phi(G, GR, a_1, a_2, \ldots, a_n)$ *is an instance of any of the special predicates associated with a group decision-making mechanism (e.g., those listed above for the proposal-based mechanism), then*

$$\models \quad \phi(G, GR, a_1, a_2, \ldots, a_n) \; \Leftrightarrow \; authDecl(G, GR, \phi(G, GR, a_1, a_2, \ldots, a_n)).$$

2.2.4 Declarations for a Proposal-based Mechanism: Context-Free Proposals

This section presents the allowed propositional content for a sample, proposal-based mechanism. The allowed declarations include making proposals, voting on proposals and announcing decisions. The proposal-based mechanism can be used both for context-free decisions (e.g., to establish a group's commitment to some proposed activity) or for context-bound decisions (e.g., to update intentions relating to some activity to which the group is already committed). This section restricts attention to the case of context-free proposals. Context-bound proposals are treated in Section 2.2.7.

The content of declarations used in the proposal-based mechanism in the case of context-free proposals is restricted to one of the following forms, each shown with its intended interpretation):

- $\delta(G_1, GR, \mathit{MadeCFP}(G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_1 proposes to the group GR that they establish a shared intention to do an action of type α, to be subject to the coordinated-cultivation requirement, where I is a unique identifier for the proposed intention (which also serves to identify the proposal itself) and Υ_0 is an (optional) ordered list of intention-update operations to be applied to the new intention, if adopted (e.g., Υ_0 might contain a set of *add-temporal-constraint* operations corresponding to a set of temporal constraints specific to the proposed activity).

- $\delta(G_2, G_1, \mathit{AcceptedCFP}(G_2, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_2 declares to agent G_1 that it accepts the proposal that the group GR commit to doing α together.

- $\delta(G_2, G_1, \mathit{RejectedCFP}(G_2, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_2 declares to agent G_1 that it rejects the proposal that the group GR commit to doing α together.

- $\delta(G_1, GR, CCR^+(GR, I, \alpha, \Upsilon_0))$

Agent G_1 declares that the group GR has decided to accept the proposal that the group GR commit to doing α together, where the formula, $CCR^{+}(GR, I, \alpha, \Upsilon_0)$, is an abbreviation for the following expression:[6]

$$\begin{cases} \Box\, CCR(GR, I) \\ \wedge \\ (\forall G \in GR)\ \mathcal{O}(G, GR, \mathit{AdoptInt}(I, \alpha, GR, \Upsilon_0)) \end{cases}$$

where $\Box$ is the "all-future-times" operator in DDLTLB (cf. Section 2.1.4), and the coordinated-cultivation requirement, $CCR(GR, I)$, is as defined below in Section 2.4.[7]

- $\delta(G_1, GR, \mathit{GroupRejectedCFP}(G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_1 declares that the group GR has decided to reject the proposal that the group GR commit to doing α together.

Notice that providing the rejection declarations is not strictly necessary, but may facilitate an agent's reasoning about the proposal-based mechanism.

2.2.5 Authorization for Declarations in the Proposal-Based Mechanism: Context-Free Proposals

This section specifies the conditions under which an agent is authorized to make the types of declarations allowed in the sample, proposal-based mechanism in the case of context-free proposals. Several of the authorization conditions employ the temporal operator $\Diamond^{\leftarrow}$ (cf. Section 2.1.4). For example, the authorization conditions for a voting declaration include the condition $\Diamond^{\leftarrow}\mathit{MadeCFP}(G_1, GR, I, \alpha, \Upsilon_0)$ (i.e., that the agent G_1 made the specified context-free proposal at some point in the past).

- **Context-Free Proposal**: Any member G_1 of a group GR may make a context-free proposal to the group:

  $$\begin{aligned} \models\ & \mathit{auth}(G_1, GR,\ \delta(G_1, GR, \mathit{MadeCFP}(G_1, GR, I, \alpha, \Upsilon_0)) \\ & \Leftrightarrow\ (G_1 \in GR) \end{aligned}$$

  where it is assumed that I is a unique identifier to be assigned to the new intention (if adopted), α is a complex act-type, and Υ_0 is an ordered list of intention-update operations.

- **Vote to Accept or Reject**: An agent G_2 is authorized to vote to accept or reject a proposal if and only if:

[6] The universally quantified clause is a convenient syntactic abbreviation for a conjunction of clauses, one for each agent in the group, made possible by the assumption that the contents of the group GR are known and fixed.

[7] The formal analysis of the proposal-based mechanism in Section 2.3 is independent on the definition of the coordinated-cultivation requirement.

(1) the proposal was made at some point in the past,

(2) the voting agent is not the originator of the proposal,

(3) the voter has not already voted to accept,

(4) the voter has not already voted to reject, and

(5) the originator of the proposal has not already made an announcement that the group rejected the proposal.

$$\begin{aligned}\models\ \ & auth(G_2, G_1, \delta(G_2, G_1, AcceptedCFP(G_2, G_1, GR, I, \alpha, \Upsilon_0))) \\ & \Leftrightarrow\ V_1 \wedge V_2 \wedge V_3 \wedge V_4 \wedge V_5\end{aligned}$$

where:

$$\begin{aligned}V_1 &\equiv \Diamond^{\leftarrow} MadeCFP(G_1, GR, I, \alpha, \Upsilon_0) \\ V_2 &\equiv (G_2 \neq G_1) \\ V_3 &\equiv \neg \Diamond^{\leftarrow}\ declared(G_2, G_1, AcceptedCFP(G_2, G_1, GR, I, \alpha, \Upsilon_0)) \\ V_4 &\equiv \neg \Diamond^{\leftarrow}\ declared(G_2, G_1, RejectedCFP(G_2, G_1, GR, I, \alpha, \Upsilon_0)) \\ V_5 &\equiv \neg \Diamond^{\leftarrow}\ declared(G_1, GR, GroupRejectedCFP(G_1, GR, I, \alpha, \Upsilon_0))\end{aligned}$$

The authorization for voting to reject a proposal is the same.

- **Announce Group Acceptance**: An agent G_1 is authorized to announce that the group GR has decided to commit to a proposed group activity if and only if:

(1) G_1 originated the proposal,

(2) G_1 has not already made an announcement that the group rejected the proposal,

(3) G_1 has not already made an announcement that the group accepted the proposal, and

(4) the rest of the agents in the group GR voted to accept the proposal.

$$\begin{aligned}\models\ \ & auth(G_1, GR,\ \delta(G_1, GR,\ CCR^{+}(GR, I, \alpha, \Upsilon_0))) \\ & \Leftrightarrow\ A_1 \wedge A_2 \wedge A_3 \wedge A_4\end{aligned}$$

where:

$$\begin{aligned}A_1 &\equiv \Diamond^{\leftarrow} MadeCFP(G_1, GR, I, \alpha, \Upsilon_0) \\ A_2 &\equiv \neg \Diamond^{\leftarrow}\ declared(G_1, GR, GroupRejectedCFP(G_1, GR, I, \alpha, \Upsilon_0)) \\ A_3 &\equiv \neg \Diamond^{\leftarrow}\ declared(G_1, GR, CCR^{+}(GR, I, \alpha, \Upsilon_0)) \\ A_4 &\equiv (\forall G \in GR, G \neq G_1)\ \Diamond^{\leftarrow} AcceptedCFP(G, G_1, GR, I, \alpha, \Upsilon_0)\end{aligned}$$

where the universally quantified proposition in A_4 is a syntactic abbreviation for a conjunction of propositions, one for each agent in the group other than G_1.

- **Announce Group Rejection**: An agent G_1 is authorized to announce that the group GR has decided to reject a proposal that they commit to doing α together if and only if:

  (1) G_1 originated the proposal;

  (2) G_1 has not already made an announcement that the group rejected the proposal; and

  (3) G_1 has not already made an announcement that the group accepted the proposal.

$$\begin{aligned} \models \quad & auth(G_1, GR,\ \delta(G_1, GR, GroupRejectedCFP(G_1, GR, I, \alpha, \Upsilon_0)) \\ & \Leftrightarrow\ A_1 \wedge A_2 \wedge A_3 \end{aligned}$$

  Thus, the originator is allowed to announce the group's rejection of a proposal even if all other agents voted in favor of it, which is useful if the originator has discovered, for example, that changes in the environment have made the proposal inconsistent with the group's shared intention.

2.2.6 Example: Using the Proposal-based Mechanism to Establish a Group's Commitment to some new Activity

Context-free proposals are used to establish a group's commitment to some new collaborative activity. The result of a group decision to accept a context-free proposal is that each agent is obliged to adopt a new intention relating to the group activity. Furthermore, that new intention shall be subject to the coordinated-cultivation requirement. This section illustrates how a simple sequence of declarations in the proposal-based mechanism can be used by a group GR of agents to establish a group decision to commit to some new collaborative activity α.

The Proposal. The proposal, made by an arbitrary agent $G_1 \in GR$, is a context-free proposal that the group commit to doing some activity α together:

$$declared(G_1, GR, MadeCFP(G_1, GR, I_{64}, \alpha, \Upsilon_0)),$$

where I_{64} shall be the unique identifier for the shared intention should the group accept the proposal, and Υ_0 represents an optional list of intention-update operations to be applied to the new intention, if adopted.

Since $G_1 \in GR$, the above declaration is authorized:

$$auth(G_1, GR, MadeCFP(G_1, GR, I_{64}, \alpha, \Upsilon_0)).$$

Thus, the propositional content is established:

$$MadeCFP(G_1, GR, I_{64}, \alpha, \Upsilon_0).$$

Voting on the Proposal. That G_1 made a proposal authorizes the rest of the agents in the group GR to vote either to accept or reject that proposal:

$$(\forall G \in GR, G \neq G_1)\ \mathit{auth}(G, G_1, \delta(G, G_1, \mathit{AcceptedCFP}(G, G_1, GR, I_{64}, \alpha, \Upsilon_0)))$$
$$(\forall G \in GR, G \neq G_1)\ \mathit{auth}(G, G_1, \delta(G, G_1, \mathit{RejectedCFP}(G, G_1, GR, I_{64}, \alpha, \Upsilon_0))).$$

Suppose that each of them votes to accept the proposal:

$$(\forall G \in GR, G \neq G_1)\ \mathit{declared}(G, G_1, \mathit{AcceptedCFP}(G, G_1, GR, I_{64}, \alpha, \Upsilon_0)).$$

Since these are authorized declarations, the corresponding propositional content is established:

$$(\forall G \in GR, G \neq G_1)\ \mathit{AcceptedCFP}(G, G_1, GR, I_{64}, \alpha, \Upsilon_0).$$

The Announcement of a Group Decision. Since every agent voted to accept G_1's proposal, G_1 is authorized to announce the group's acceptance of that proposal:

$$\mathit{auth}(G_1, GR,\ \delta(G_1, GR,\ CCR^+(GR, I_{64}, \alpha, \Upsilon_0))).$$

Suppose G_1 does so:

$$\mathit{declared}(G_1, GR,\ CCR^+(GR, I_{64}, \alpha, \Upsilon_0)).$$

Because this is an authorized declaration, its content

$$CCR^+(GR, I_{64}, \alpha, \Upsilon_0) \equiv \begin{cases} \Box\, CCR(GR, I_{64}) \\ \wedge \\ (\forall G \in GR)\ \mathcal{O}(G, GR, \mathit{AdoptInt}(I_{64}, \alpha, GR, \Upsilon_0)) \end{cases}$$

is established. Thus, the agents have established that each of them is obliged to adopt a new intention (to be identified by I_{64}), the cultivation of which shall be subject to the coordinated-cultivation requirement (cf. Section 2.4).

At this point, the agents are ready to begin cultivating their shared intention. Thus, they need to make context-bound decisions. The following sections add context-bound proposals to the sample mechanism.

2.2.7 Declarations for the Proposal-based Mechanism: Context-Bound Proposals

Context-bound proposals are used to coordinate the cultivation of existing, shared intentions. Thus, the content of a context-bound proposal specifies a list Υ of intention-update operations to be applied to the group's shared intention. This section presents the allowed propositional content for the sample, proposal-based mechanism in the case of context-bound proposals.

Some of the authorization conditions (e.g., a condition that *some* agent made an authorized declaration to establish the coordinated-cultivation requirement for this intention) would be naturally expressed using existential quantification; however, DDLTLB does not accommodate existential quantification. Therefore, to avoid the need for existential quantification in certain authorization conditions, the special predicates for context-bound proposals carry along the arguments from the context-*free* proposal that established the coordinate-cultivation requirement for the shared intention under consideration.

The content of declarations used in the proposal-based mechanism in the case of context-bound proposals is restricted to one of the following forms, each shown with its intended interpretation):

- $\delta(G_1', GR, \mathit{MadeCBP}(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_1' proposes to the group GR that they perform the intention-update operations listed in Υ. P is a unique identifier for the proposal. The rest of the arguments are carried over from the context-*free* proposal that established the group's commitment to doing α together.

- $\delta(G_2, G_1', \mathit{AcceptedCBP}(G_2, G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_2 declares to agent G_1' that it accepts the context-bound proposal identified by P.

- $\delta(G_2, G_1', \mathit{RejectedCBP}(G_2, G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_2 declares to agent G_1' that it rejects the context-bound proposal identified by P.

- $\delta(G_1', GR, \mathit{GrAccCBP}^+(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_1' declares that the group GR has decided to accept the context-bound proposal identified by P.

  In the above formula, $\mathit{GrAccCBP}^+(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)$ abbreviates

  $$\begin{cases} (\forall G \in GR)\ \mathcal{O}(G, GR, \mathit{Update}(I, \Upsilon)) \\ \wedge\ \mathit{GroupAcceptedCBP}(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0). \end{cases}$$

- $\delta(G_1', GR, \mathit{GroupRejectedCBP}(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))$

  Agent G_1' declares that the group GR has decided to reject the context-bound proposal identified by P.

2.2.8 Authorization for Declarations in the Proposal-Based Mechanism: Context-Bound Proposals

This section specifies the authorization conditions for the different classes of declarative speech-acts in the proposal-based mechanism in the case of context-bound proposals. The authorization conditions for voting on a context-bound proposal or announcing that the group has accepted or rejected a context-bound proposal are completely analogous to their context-free counterparts. In contrast, the authorization conditions for *initiating* a context-bound proposal include that the coordinated-cultivation requirement holds for the intention that the context-bound proposal concerns; thus, the authorization conditions refer to a past declaration that established the coordinated-cultivation requirement for that intention. As mentioned previously, carrying the arguments from the context-*free* proposal that established the coordinated-cultivation requirement for that intention enables the authorization conditions to avoid using existential quantification.

- **Context-Bound Proposal**: Any member of a group may make a context-bound proposal concerning a shared intention (identified by I) for which the coordinated-cultivation requirement holds. The following axiom says that an agent G_1' is authorized to make a context-bound proposal if and only if at some point in the past a possibly different agent G_1 made an authorized declaration establishing the group's commitment to that activity.

$$\begin{aligned} \models\ & auth(G_1', GR, \delta(G_1', GR, MadeCBP(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))) \\ & \Leftrightarrow\ (G_1' \in GR)\ \wedge\ \Diamond^{\leftarrow} authDecl(G_1, GR, CCR^{+}(GR, I, \alpha, \Upsilon_0)), \end{aligned}$$

  where $CCR^{+}(GR, I, \alpha, \Upsilon_0)$ is as defined in Section 2.2.4. Notice that without carrying along the arguments from the context-free proposal, this authorization condition would be difficult to state without existential quantification.

- **Vote to Accept or Reject**:

$$\begin{aligned} \models\ & auth(G_2, G_1', \delta(G_2, G_1', AcceptedCBP(G_2, G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))) \\ & \Leftrightarrow\ V_1' \wedge V_2' \wedge V_3' \wedge V_4' \wedge V_5' \end{aligned}$$

  where:

$$\begin{aligned} V_1' &\equiv \Diamond^{\leftarrow} MadeCBP(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0) \\ V_2' &\equiv (G_2 \neq G_1') \\ V_3' &\equiv \neg \Diamond^{\leftarrow}\ declared(G_2, G_1', AcceptedCBP(G_2, G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)) \\ V_4' &\equiv \neg \Diamond^{\leftarrow}\ declared(G_2, G_1', RejectedCBP(G_2, G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)) \\ V_5' &\equiv \neg \Diamond^{\leftarrow}\ declared(G_1', GR, GroupRejectedCBP(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)). \end{aligned}$$

  The authorization to vote to reject a proposal is the same.

- **Announce Group Acceptance**:

$$\begin{aligned} \models\ & \mathit{auth}(G_1', GR, \delta(G_1', GR, \mathit{GrAccCBP}^+(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))) \\ & \Leftrightarrow\ A_1' \wedge A_2' \wedge A_3' \wedge A_4' \end{aligned}$$

where:

$$\begin{aligned} A_1' &\equiv \Diamond^{\leftarrow}\mathit{MadeCBP}(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0) \\ A_2' &\equiv \neg\Diamond^{\leftarrow}\ \mathit{declared}(G_1', GR, \mathit{GroupRejectedCBP}(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)) \\ A_3' &\equiv \neg\Diamond^{\leftarrow}\ \mathit{declared}(G_1', GR, \mathit{GrAccCBP}^+(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)) \\ A_4' &\equiv (\forall G \in GR, G \neq G_1')\ \Diamond^{\leftarrow}\mathit{AcceptedCBP}(G, G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0) \end{aligned}$$

- **Announce Group Rejection**:

$$\begin{aligned} \models\ & \mathit{auth}(G_1', GR, \delta(G_1', GR, \mathit{GroupRejectedCBP}(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)) \\ & \Leftrightarrow\ A_1' \wedge A_2' \wedge A_3' \end{aligned}$$

2.2.9 Example: Using the Proposal-based Mechanism to Make a Group Decision to do an Intention-Update Operation

The example in Section 2.2.6 ended with the agent G_1's authorized declaration establishing the group's decision to commit to doing α together:

$$\mathit{authDecl}(G_1, GR, \mathit{CCR}^+(GR, I_{64}, \alpha, \Upsilon_0)),$$

where

$$\mathit{CCR}^+(GR, I_{64}, \alpha, \Upsilon_0) = \begin{cases} \Box\, \mathit{CCR}(GR, I_{64}) \\ \wedge \\ (\forall G \in GR)\ \mathcal{O}(G, GR, \mathit{AdoptInt}(I_{64}, \alpha, GR, \Upsilon)) \end{cases}.$$

A Context-Bound Proposal. As a result of the above declaration, each agent $G \in GR$ is now authorized to make context-bound proposals concerning candidate intention-update operations for the newly established intention (identified by I_{64}). For example, an agent G_1' is authorized to propose that the group bind the parameter $?\mathtt{K}$ to the value $\mathtt{Car}_{39}$:

$$\mathit{auth}(G_1', GR, \mathit{MadeCBP}(G_1', \mathtt{BIND}(?\mathtt{K}, \mathtt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0)).$$

Suppose G_1' makes such a proposal.

$$\mathit{declared}(G_1', GR, \mathit{MadeCBP}(G_1', \mathtt{BIND}(?\mathtt{K}, \mathtt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0)).$$

Since it is an authorized proposal, it establishes its propositional content:

$$\mathit{MadeCBP}(G_1', \mathtt{BIND}(?\mathtt{K}, \mathtt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0).$$

Voting on the Proposal. The above proposal authorizes the rest of the agents in the group to vote on it. Suppose that each of them votes to accept the proposal:

$$\begin{array}{l}(\forall G \in GR, G \neq G_1') \\ \quad \mathit{declared}(G, G_1', \mathit{AcceptedCBP}(G, G_1', \texttt{BIND}(?\texttt{K}, \texttt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0)).\end{array}$$

Since these are authorized declarations, their propositional content is thereby established:

$$(\forall G \in GR, G \neq G_1')\ \mathit{AcceptedCBP}(G, G_1', \texttt{BIND}(?\texttt{K}, \texttt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0).$$

The Announcement of a Group Decision. As a result of the above, G_1' is authorized to announce the group decision:

$$\mathit{declared}(G_1', GR, \mathit{GrAccCBP}^{+}(G_1', \texttt{BIND}(?\texttt{K}, \texttt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0)).$$

Since this is an authorized declaration, its content

$$\mathit{GrAccCBP}^{+}(G_1', \texttt{BIND}(?\texttt{K}, \texttt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0),$$

which is an abbreviation for

$$\left\{\begin{array}{l}(\forall G \in GR)\ \mathcal{O}(G, GR, \mathit{Update}(I_{64}, \texttt{BIND}(?\texttt{K}, \texttt{Car}_{39}))) \\ \wedge\ \mathit{GroupAcceptedCBP}(G_1', \texttt{BIND}(?\texttt{K}, \texttt{Car}_{39}), P_{15}, G_1, GR, I_{64}, \alpha, \Upsilon_0),\end{array}\right.$$

is established. Hence, the agents in the group are obliged to update their intentions identified by I_{64} to reflect that the parameter $?\texttt{K}$ has been bound to the value $\texttt{Car}_{39}$.

2.3 A Formal Analysis of the Proposal-Based Mechanism

This section states and proves several important properties of the proposal-based mechanism. The main result says that, under the assumptions about agent beliefs given below, an announcement of a group decision happens only if duly authorized by the group.

2.3.1 Assumptions about Agent Beliefs

The analysis in this section assumes a standard KD45 modal belief operator (Chellas, 1980). All axioms presented in this section are assumed to be common knowledge.

Axiom 2.4 (Declarations and Mutual Belief) *The recipients of a declaration have correct and complete mutual belief about the occurrence (or non-occurrence) of that declaration:*

$$\begin{array}{l}\models \mathit{declared}(G, GR, \phi)\ \Leftrightarrow\ MB(GR, \mathit{declared}(G, GR, \phi)) \\ \models \neg\mathit{declared}(G, GR, \phi)\ \Leftrightarrow\ MB(GR, \neg\mathit{declared}(G, GR, \phi))\end{array}$$

In addition, their mutual beliefs are correct and complete concerning declarations made in the past:

$$\begin{array}{l}\models\ \Diamond^{\leftarrow}\mathit{declared}(G, GR, \phi)\ \Leftrightarrow\ MB(GR, \Diamond^{\leftarrow}\mathit{declared}(G, GR, \phi)) \\ \models\ \neg\Diamond^{\leftarrow}\mathit{declared}(G, GR, \phi)\ \Leftrightarrow\ MB(GR, \neg\Diamond^{\leftarrow}\mathit{declared}(G, GR, \phi))\end{array}$$

Recall that the speaker of a declaration is one of the recipients and, thus, participates in the mutual beliefs.

Axiom 2.5 (Sincerity) *Agents only make declarations they believe they are authorized to make.*

$$\models\ declared(G, GR, \phi)\ \Rightarrow\ \mathcal{P}Bel(G, auth(G, GR, \delta(G, GR, \phi)))$$

The purpose of the following axiom is to preclude the need for explicit sets of agents in the syntax. For example, GR is used instead of an explicit set such as: $\{G_1, G_2, \ldots, G_n\}$.

Axiom 2.6 (Agents, Groups, Identity) *Every agent G has correct and complete beliefs about whether any other agent G' is a member of the group GR, and about its own identity. Furthermore, these propositions are timeless.*

$$\begin{aligned}
&\models\ Bel(G, (G' \in GR)) \Leftrightarrow (G' \in GR)\\
&\models\ Bel(G, (G \neq G')) \Leftrightarrow (G \neq G')\\
&\models\ (G' \in GR) \Leftrightarrow \Diamond^{\leftarrow}(G' \in GR)\\
&\models\ (G \neq G') \Leftrightarrow \Diamond^{\leftarrow}(G \neq G')\\
&\models\ Bel(G, (\forall G' \in GR, G' \neq G'')\ \phi)\ \Leftrightarrow (\forall G' \in GR, G' \neq G'')\ Bel(G, \phi)
\end{aligned}$$

Axiom 2.7 (Beliefs about the Past) *If an agent G believes that at some point in the past ϕ held, then at some point in the past G believed that ϕ currently held:*

$$\begin{aligned}
&\models\ Bel(G, \Diamond^{\leftarrow}\phi) \Rightarrow \Diamond^{\leftarrow} Bel(G, \phi)\\
&\models\ Bel(G, \mathcal{P}\phi) \Rightarrow \mathcal{P}Bel(G, \phi).
\end{aligned}$$

Axiom 2.7 is a strong assumption that allows us to avoid complex issues of belief revision. It is equivalent to

$$\models\ \neg\Diamond^{\leftarrow} Bel(G, \phi) \Rightarrow \neg Bel(G, \Diamond^{\leftarrow}\phi),$$

which says that if at no point in the past did G believe that ϕ currently held, then G does not currently believe that ϕ ever held in the past.

2.3.2 An Analysis of Context-Free Proposals

This section presents a formal analysis of the sample, proposal-based group decision-making mechanism in the case of context-free proposals. The analysis culminates with Theorem 2.13 which says that if an agent G believes that some other agent G_1 declared that the group GR accepted a context-free proposal, then G_1 was, in fact, so authorized, and did make such a declaration. This result says that, given the assumptions in Section 2.3.1, agents can trust their beliefs about agent declarations in the proposal-based group decision-making mechanism.

To simplify the presentation, the proofs have been placed in Appendix A. Section 2.3.3 extends these results to context-bound proposals.

Theorem 2.8 *If an agent G believes that a possibly different agent G_1 made an authorized declaration of a context-free proposal at some point in the past, then G_1 did, in fact, do so:*

$$\begin{aligned} \models\ & \mathit{Bel}(G,\ \Diamond^{\leftarrow}\mathit{authDecl}(G_1, GR, \mathit{MadeCFP}(G_1, GR, I, \alpha, \Upsilon_0))) \\ & \Rightarrow\ \Diamond^{\leftarrow}\mathit{authDecl}(G_1, GR, \mathit{MadeCFP}(G_1, GR, I, \alpha, \Upsilon_0)). \end{aligned}$$

Theorem 2.9 *If an agent G believes it is authorized to vote either to accept or reject a context-free proposal, then G is, in fact, so authorized:*

$$\models\ \mathit{Bel}(G, \mathit{auth}(G, G_1,\ \delta(G, G_1, \xi)))\ \Rightarrow\ \mathit{auth}(G, G_1,\ \delta(G, G_1, \xi)),$$

where ξ is $\mathit{AcceptedCFP}(G, G_1, GR, I, \alpha, \Upsilon_0)$ or $\mathit{RejectedCFP}(G, G_1, GR, I, \alpha, \Upsilon_0)$. Similarly, if the originator G_1 of the proposal believes G is authorized to vote on that proposal, then G is, in fact, so authorized.

Theorem 2.10 *If an agent G_1 believes that another agent G accepted a context-free proposal originated by G_1, then G did, in fact, accept that proposal:*

$$\begin{aligned} \models\ & \mathit{Bel}(G_1, \mathit{AcceptedCFP}(G, G_1, GR, I, \alpha, \Upsilon_0)) \\ & \Rightarrow\ \mathit{AcceptedCFP}(G, G_1, GR, I, \alpha, \Upsilon_0). \end{aligned}$$

Theorem 2.11 *If an agent G_1 believes that it is authorized to declare, on behalf of the group, that the group has accepted a context-free proposal, then G_1 is, in fact, so authorized:*

$$\begin{aligned} \models\ & \mathit{Bel}(G_1, \mathit{auth}(G_1, GR,\ \delta(G_1, GR, CCR^+(GR, I, \alpha, \Upsilon_0)))) \\ & \Rightarrow\ \mathit{auth}(G_1, GR,\ \delta(G_1, GR, CCR^+(GR, I, \alpha, \Upsilon_0))). \end{aligned}$$

Theorem 2.12 *If an agent G_1 declares, on behalf of the group, that the group has accepted a context-free proposal, then that agent was, in fact, authorized to make that declaration:*

$$\begin{aligned} \models\ & \mathit{declared}(G_1, GR,\ CCR^+(GR, I, \alpha, \Upsilon_0)) \\ & \Rightarrow\ \mathit{auth}(G_1, GR,\ \delta(G_1, GR, CCR^+(GR, I, \alpha, \Upsilon_0))). \end{aligned}$$

Hence,

$$\begin{aligned} \models\ & \mathit{declared}(G_1, GR,\ CCR^+(GR, I, \alpha, \Upsilon_0)) \\ & \Rightarrow\ \mathit{authDecl}(G_1, GR,\ CCR^+(GR, I, \alpha, \Upsilon_0)). \end{aligned}$$

Theorem 2.13 *If an agent G believes that a possibly different agent G_1 declared that the group GR accepted a context-free proposal, then G_1 was, in fact, authorized to make such a declaration and did so:*

$$\begin{aligned} \models\ & \mathit{Bel}(G, \Diamond^{\leftarrow}\mathit{declared}(G_1, GR,\ CCR^+(GR, I, \alpha, \Upsilon_0))) \\ & \Rightarrow\ \Diamond^{\leftarrow}\mathit{authDecl}(G_1, GR,\ CCR^+(GR, I, \alpha, \Upsilon_0)). \end{aligned}$$

2.3.3 Extending the Analysis to Context-Bound Proposals

The difference between group decisions based on context-free proposals and those based on context-bound proposals is primarily in their content. The authorization conditions for the various declarations (making a proposal, voting on a proposal, announcing that the group has rejected or accepted a proposal) are very similar. As a result, the sequence of theorems presented in the formal analysis of context-free proposals (i.e., Theorems 2.8–2.13) may, with some minor adjustments, be carried over to a formal analysis of context-bound proposals.

The primary difference between the authorization conditions for declarations concerning context-free and context-bound proposals is in the authorization conditions for *initiating* a proposal. In particular, the authorization conditions for a context-bound proposal concerning some group activity make reference to the authorized declaration that established the coordinated-cultivation requirement for that activity. Therefore, the primary modifications to the sequence of theorems for the analysis of context-free proposals are restricted to the theorem that addresses the authorization conditions for making a proposal (i.e., Theorem 2.8). The rest of the theorems carry over with a straightforward translation of declarations. For example, the declaration of a vote to accept a context-free proposal is translated into a declaration of a vote to accept a context-bound proposal.

Since the first theorem in the sequence is the only one requiring non-trivial modifications, it is the only one addressed in this chapter. Theorem 2.14 is the modified version of Theorem 2.8. Its proof is given in Appendix A.

Theorem 2.14 *If an agent G believes that a possibly different agent G_1' made an authorized declaration of a context-bound proposal at some point in the past, then G_1' did, in fact, do so:*

$$\begin{aligned}\models\ & Bel(G, \Diamond^{\leftarrow} authDecl(G_1', GR, MadeCBP(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0))) \\ & \Rightarrow\ \Diamond^{\leftarrow} authDecl(G_1', GR, MadeCBP(G_1', \Upsilon, P, G_1, GR, I, \alpha, \Upsilon_0)).\end{aligned}$$

Given Theorem 2.14 as a starting point, modifying the rest of the theorems (and their proofs) to make them applicable to the context-bound case is trivial. The final theorem in the sequence states that an agent's beliefs about a declaration of another agent announcing a group's acceptance of a context-bound proposal are reliable. Thus, as in the context-free case, agents may trust their beliefs about announced group decisions concerning context-bound proposals.

2.4 The Coordinated-Cultivation Requirement

This section formally defines the coordinated-cultivation requirement (CCR) that prohibits collaborating agents from making unilateral intention-update decisions for their group-related intentions. The definition of the CCR is based on the treatment of conditional obligations, general prohibitions and explicit permissions by Brown (2000), illustrated by the following example: "You ought not to drive through an intersection while the traffic light is red unless explicitly permitted to do so (by a duly constituted authority)." The CCR stipulates that an agent ought not to update its shared intention unless explicitly permitted to do so by an authorized declaration of a group decision. Unlike Brown, the obligation operator $\mathcal{O}$ used in this thesis represents a *social obligation* that includes an argument specifying the group to which an agent is obliged.

Definition 2.15 (Coordinated-Cultivation Requirement) *The coordinated-cultivation requirement for a group GR of agents and an intention identifier I is represented by the formula $CCR(GR, I)$ which abbreviates the following:*

$$(\forall\Upsilon)\begin{cases}(\exists G_1 \in GR)\ \Diamond^{\leftarrow} authDecl(G_1, GR, (\forall G \in GR)\ \mathcal{O}(G, GR, Update(I, \Upsilon)))\\ \vee \\ (\forall G \in GR)\ \mathcal{O}(G, GR, \neg Update(I, \Upsilon)).\end{cases}$$

This formula, in which the universally quantified variable Υ represents an arbitrary set of intention-update operations, says that unless some member of the group made an authorized declaration establishing that each member G of the group GR is explicitly obliged to do the set Υ of intention-update operations, then each member of the group is obliged *not* to do them

This definition of the coordinated-cultivation requirement employs universal and existential quantification, which are not provided for in the current formulation of the logic DDLTLB used in this thesis. Extending the semantics of DDLTLB to provide for universal and existential quantification is beyond the scope of this thesis. However, the formal analysis of the sample, proposal-based mechanism presented in Section 2.3 is independent of the definition of the coordinated-cultivation requirement.

In practice, an agent can maintain a list of explicitly authorized declarations relevant to the CCR. For its intention identified by I, the agent is obliged to do all of the intention-update operations authorized and obliged by such explicit declarations and no others.

2.5 Enabling an Agent's Participation

The definition of any group decision-making mechanism must specify:

- the legal inputs agents can make; and
- the ways of combining inputs from multiple agents to generate authorizations for various group decisions.

To participate in such a mechanism, an agent must be able to:

- generate candidates for its input to the mechanism; and
- determine whether a given candidate might lead to obligations that might conflict with its current commitments.

The candidate-testing problem arises because any input an agent makes to a mechanism might combine with the inputs of other agents to authorize a group decision and, hence, subject the agent to various obligations. An agent typically honors such obligations by modifying an existing intention or creating a new intention. However, a rational agent does not knowingly create conflicts among its intentions. Thus, an agent must be able to reason about the interaction between its set of pre-existing commitments and any obligations that might ensue from its participation in a group decision-making mechanism which, of course, leads to the well-known *ramification problem* (Russell and Norvig, 1995).

One way an agent might be able to ease its computational burden when reasoning about its participation in a group decision-making mechanism would be to limit the portion of its private knowledge base that such reasoning takes into consideration. For example, when considering whether to support a proposal that our group rent a '62 Chevy or a proposal that we rent an '86 Ford Bronco, I might only need to consider things like gas mileage and cargo space. This thesis refers to the portion of an agent's knowledge base used by that agent when reasoning about its participation in a group decision-making mechanism as the agent's *mechanism-participation context* (MPC). This thesis refers to the problem of generating a suitable mechanism-participation context as the *MPC-Generation Problem*.

When generating a mechanism-participation context, an agent must ensure (as much as possible) that the generated MPC is in some sense *decoupled* from the rest of its knowledge base, relative to the reasoning problems associated with its participation in the given mechanism. For example, if I limit my reasoning about the '62 Chevy and the '86 Bronco to issues of gas mileage and cargo space, I'd better be sure that my knowledge of off-road routes is not relevant. One way to make sure of this would be to decide that I will not go off road between here and New York City; in fact, I might decide to stay on the Interstate the whole way. In general, the application of additional constraints is necessary to decouple an MPC from the rest of an agent's private knowledge base.

When a group decision is reached, implementing the decision may give rise to various coordination issues. For example, suppose we have decided that I shall buy the food for dinner and you shall buy the wine. Since certain combinations of food and wine are more pleasing than others, my choice of food and your choice of wine are not independent. We could coordinate our choices in many ways. For example, we might negotiate by cell phone as we head to our different stores. Alternatively, we might decide to act independently, each making the choice that is least likely to offend the palette given our uncertainty about what the other will buy. We refer to the problem of coordinating agent activity after a group decision has been implemented as the *post-decision coordination problem*.

In summary, to facilitate their participation in group decision-making mechanisms, agents must be able to deal with the following problems:

- The candidate-generation problem
- The candidate-testing problem
- The MPC-generation problem

- The post-decision coordination problem

The first three of these problems are single-agent problems; the last is a multi-agent coordination problem.

The next chapter focuses on the problem of how agents can determine, both individually and collectively, whether to commit to some proposed group activity, that is, the Initial-Commitment Decision Problem (ICDP). It introduces a group decision-making mechanism based on a combinatorial auction that agents can use to solve instances of the ICDP. Each of the problems listed above has a more specific counterpart for the ICDP mechanism. Subsequent chapters provide algorithms that agents can use to solve such problems and, thus, to participate effectively in the auction-based ICDP mechanism.