CONTENTS

ANAND RAO AND MICHAEL WOOLDRIDGE
Foundations and Theories of Rational Agents 1

STUART RUSSELL
Rationality and Intelligence 11

AARON SLOMAN
What sort of architecture is required for a human-like agent? 35

JOHN L. POLLOCK
Planning Agents 53

PIOTR J. GMYTRASIEWICZ
Toward Rational Interactions in Multiagent Domains 83

MUNINDAR P. SINGH
Know-How 109

W. VAN DER HOEK, B. VAN LINDER AND J-J.CH. MEYER
An Integrated Modal Approach to Rational Agents 137

DAVID R. TRAUM
Speech Acts for Dialogue Agents 173

AFSANEH HADDADI
Communication among Rational Agents 207

BARBARA J. GROSZ AND SARIT KRAUS
The Evolution of SharedPlans 233

S. REBECCA THOMAS
A Survey of Agent-Oriented Programming 273

YVES LESPÉRANCE, HECTOR J. LEVESQUE, AND RAYMOND REITER
A Situation Calculus Approach to Modeling and Programming Agents 285

BARBARA J. GROSZ AND SARIT KRAUS

THE EVOLUTION OF SHAREDPLANS

1 INTRODUCTION

Rational agents often need to work together. There are jobs that cannot be done by one agent—for example, singing a duet or operating a computer network—and jobs that are more efficiently done by more than one agent—for example, hanging a door or searching the Internet. Collaborative behavior—coordinated activity in which the participants work jointly with each other to satisfy a shared goal—is more than the sum of individual acts [24, 8] and may be distinguished from both interaction and simple coordination in terms of the commitments agents make to each other [4, 10, 9]. A theory of collaboration must therefore treat not only the intentions, abilities, and knowledge about action of individual agents, but also their coordination in group planning and acting. It also must account for the ways in which plans are incrementally formed and executed by the participants.

Our long term goal is to provide a formalization of collaboration that supports the design and construction of collaborative computer systems, including systems that are able to collaborate with one another [13], systems that support groups of people working together, and collaborative systems for human-computer communication [16, 22]. The SharedPlan formalization of collaboration [10, 7] provides specifications that are normative, but it has been informed by an analysis of human collaborations [1]. The formalization is based on a mental-state view of plans [2, 21]: agents are said to have plans when they have a particular set of intentions and beliefs.

Agents may have different beliefs about the ways to do an action or to achieve some desired state. Pollack [21] argued that, as a result, a move away from a view of plans as primarily data-structure-like (i.e.,a plan as a set of [partially] ordered actions that when done in the appropriate conditions lead to a specified new state of the world) to a view of plans as mental state was necessary for plan inference.[1] Bratman [2, 3] also argues for a mental-state view of plans, emphasizing the importance of intentions to plans. Having a plan is not merely knowing how to do an action, but also having the intention to do the actions entailed. Bratman argues

[1]Pollack showed in particular that these differences among agent beliefs must be detected in plan recognition because they may affect the ways in which a question-answering system should respond to questions that are possibly incomplete or inaccurate.

that intentions to do an action play three roles in rational action: having an intention to do an action constrains the other intentions an agent may adopt, focuses means-ends reasoning, and guides replanning.

To adequately model collaboration it is necessary both to accommodate differences among the beliefs of individual participants and to distinguish between knowledge about action performance and intention to act. Agents may differ not only in their beliefs about the ways to perform an action and the state of the world, but also in their assessments of the ability and willingness of an individual to perform an action. The three roles Bratman describes for intentions may be even more important for collaborative activity than for individual plans. Hence, the adoption of a mental-state view of plans is essential to the formalization.

Following Pollack, we will use the terms "recipe" and "plan" to distinguish between knowing how to do an action and having a plan to do the action. When agents have a SharedPlan to do a group action, they have certain individual and mutual beliefs about how the action and its constituent subactions are to be done. We use the term *recipe* [21, 1, 17] to refer to a specification of a set of actions, which we will denote as β_i $(1 \leq i \leq n)$, the doing of which under appropriate *recipe-constraints*, denoted as ρ_j $(1 \leq j \leq m)$, constitutes performance of α.[2] We use the meta-language symbol R_α to denote a particular recipe. The β_i may be either single-agent or multi-agent actions; they may also be complex, i.e., may comprise constituent subactions. In some cases we may need to refer to a partial (possibly empty) recipe for α; we will use the meta-language symbol R^p_α to denote such a partial recipe.

The SharedPlan formalization provides mental-state specifications of both *SharedPlans* and *individual plans*. SharedPlans are constructed by groups of collaborating agents and include subsidiary SharedPlans [16] formed by subgroups as well as subsidiary individual plans formed by individual participants in the group activity. The formalization distinguishes between complete plans—those in which all the requisite beliefs and intentions have been established—and partial plans. In addition to the propositional attitude of *intending to* do an action, it introduces the attitude of *intending that* a proposition hold.

In an earlier paper [10] we provide definitions for individual plans and SharedPlans, both complete and partial, and briefly describe the planning actions that play a role in expanding partial plans to complete ones. The focus of that paper was on specifying the capabilities to act and mental attitudes (beliefs and intentions) required for the participants in a group activity to have a collaborative plan. This paper complements that one by examining in more detail the dynamics of SharedPlans and the role of intentions-that in their evolution.

The remainder of this paper is organized as follows. The next section provides an overview of the formalization of partial SharedPlans and presents an example

[2] The indices i and j are distinct; for simplicity of exposition, we omit the range specifications in the remainder of this document.

used throughout the paper to illustrate various features of the formalization and its use in the design of automated collaborative agents. Section 3 details extensions to the formalization of intentions-that. Section 4 presents an architecture for the design of collaborating agents based on the formalization. Section 5 briefly concludes with mention of some important open problems.

2 BACKGROUND

2.1 Basic Tenets of the SharedPlan Approach

Because the world is dynamic and uncertain and an agent's knowledge is typically incomplete, recent efforts in the areas of planning and rational action have focused on ways to coordinate planning and acting and on methods for dealing with resource limitations [14, 5, 6, 11, 14, 26, inter alia]. Change, uncertainty, and incomplete knowledge also affect agents when they work together and thus must be accommodated by any theory of collaboration. Plans—whether of an individual or of a group—evolve over time. Although completed plans are the ultimate aim of planning, partial rather than complete SharedPlans best characterize the mental state of participants in a group activity. As a result, the evolution of SharedPlans is, in reality, the evolution of partial SharedPlans, and this paper focuses largely on partial SharedPlans.

Partial individual plans get expanded to more complete plans through means-ends reasoning about intended actions. Collaborative plans have a correlate of this reasoning, or plan-elaboration, process, but their expansion entails communication and negotiation as well as means-ends reasoning about the way in which to do the group action. In our formalization, means-ends reasoning is represented by the complex planning action Elaborate_Individual and the group planning activity is represented by the complex action Elaborate_Group.[3]

A basic claim of the SharedPlans formalization is that collaborative activity is rooted in the individual mental actions and domain actions of individual agents. This constraint holds of complex group planning actions as well as of domain actions. Thus, Elaborate_Group comprises individual planning actions of the group members. We use the the complex action I_Elaborate_Group to represent these individual actions.

In multi-agent activities participants not only do means-ends reasoning about their own actions, they also reason about how to coordinate with and support the actions of others in the group. These activities require plan-based reasoning that arises from the participants' attitudes of intentions-that toward the actions of others and of the group as a whole. To handle this aspect of the dynamics of SharedPlans we introduce a new complex planning action, Cultivate.

[3] In the formalization [10], Elaborate_Individual appears in the definition of the modal operator for intentions-to act, Int.To, and Elaborate_Group appears in the definition of SharedPlan.

Thus, two different complex planning actions—Elaborate_Group and Cultivate—combine to provide the planning processes for group activities analogous to those that Elaborate_Individual provides for individual actions. Cultivate is the general process that is active whenever an agent has an intention-that, just as Elaborate_Individual is active whenever an agent has an intention-to. Elaborate_Group is activated whenever the intention-that leads to a SharedPlan; it serves to oversee the completion of the plan.

This paper extends the original treatment of the modal operator for intentions-that, Int.Th, by providing additional axioms, specifying the minimal conditions for an agent having an intention-that, and describing various ways in which intentions-that can lead to actions in the support of a collaborative plan. It also presents an architecture for an automated collaborative agent based on the specification provided by the SharedPlan definitions. This architecture includes I_ELABORATE_GROUP and CULTIVATE processes which realize the correspondingly named complex actions of the formalization. The paper uses a slightly revised definition of partial SharedPlans. This new definition covers a wider range of transitional states in partial plans and provides for more generality in the planning processes.

2.2 *A Collaborative Planning Scenario*

To help explain the definition of partial SharedPlan (PSP), we will examine a scenario based on the Truckworld simulated environment [12] which was the test environment for an initial implementation [20]. This scenario is portrayed in Figures 4 and 5. MT_1 and MT_2 are mobile truck agents; ST_q is a stationary truck agent with crane-arms located at Q, and ST_b is a stationary truck agent with crane-arms located at B. G_q and G_b are gas tanks. In the scenarios we will consider, the four agents MT_1, MT_2, ST_q, and ST_b, have formed a collaborative team, TG, to move certain building material from Q to B. We assume initially that the agents are partway in forming their plan for performing this group action, which we will refer to as MOVE39. They have identified a recipe for doing the action—i.e., they have figured out at a relatively high level of description how to do MOVE39—but they have not yet worked out the lower-level details. For instance, they may not have decided how to do some subactions in the recipe or who will do them.

Collaborative activity requires a commitment on the part of the participants to the joint endeavor [4, 10, 15]. Thus, each of the team members in TG must be committed to the group's moving material from Q to B. In the basic scenario, the group has chosen a recipe which comprises two shipment actions, each of which will move some of the material. The first shipment action, Shipm1, will be performed by the subgroup TG1, comprising MT_1, ST_q and ST_b. TG1 must form a SharedPlan for doing Shipm1. This SharedPlan is subsidiary to the SharedPlan for MOVE39; it may be either partial or complete. For illustrative purposes, we assume TG1 has decided to use a recipe with the following three subsidiary actions: a subgroup of TG1 loads a truck (this truck is one of the members of TG1), the truck drives to B, and a subgroup of TG1 unloads the truck.

In collaborative activity, plans for subactions also impose some constraints on the group doing the overall activity. In particular, the full group must agree on who will do the subsidiary action, must have confidence that the subgroup can and will do the action, and must be committed to the subgroup's success. However, it would be too burdensome (engendering unnecessary communication and computation) to require that agents outside the subgroup know all the details of how the subgroup will act. Thus, the formalization requires that the full group, TG, have mutual belief that the subgroup TG1 has formed a SharedPlan for Shipm1 and that TG1 can perform this action. It does not require though that MT_2, who is not in TG1, know the recipe TG1 has chosen; rather MT_2 must believe there is some recipe that the group TG1 can use. In addition MT_2 must be committed to TG1's being able to do the Shipm1 action.

For TG1 to complete their SharedPlan for Shipm1, they must decide who will do each subsidiary action in the recipe. For instance, they might decide, as in the basic scenario in Figure 5, that ST_q and MT_1 will load MT_1, MT_1 will drive to B, and then MT_1 and ST_b will unload MT_1. If so, MT_1 and ST_q must form a complete subsidiary SharedPlan to do the loading action; MT_1 must have an intention-to drive from Q to B and a plan for doing so; and MT_1 and ST_b must form a complete subsidiary SharedPlan to do the unloading action.

We assume initially that TG has not yet determined who will do the second shipment action, Shipm2. We refer to recipe subactions in this status as "unresolved actions."[4] Collaborative activity imposes two constraints on participants with respect to unresolved actions. All of the group members must believe that some subgroup can do the unresolved action, and the group must have a way to choose a subgroup to do it. Thus, to resolve Shipm2 (making it a "resolved action"), TG must agree on a subgroup, TG2, to do the action and form beliefs and intentions analogous to those described above for Shipm1. In choosing who will do Shipm2, the members of TG must take into account what they know about the plans (especially the commitments made by members of TG1) for Shipm1.

The scenario variations given in Figure 5 will be used later in the paper to illustrate various features of the formalization and architecture. For some of these examples, we will assume that the recipe for Shipm2 is identical to the one for Shipm1, but that MT_2 will drive the load of rocks from Q to B for Shipm2.

2.3 Partial Plans: Definition and Illustration Using Scenario

The formalization uses two meta-predicates to which later sections of this paper refer. CBA ("can bring about") and CBAG ("can bring about group") are used to

[4]In our earlier work, we referred to these actions as "unreconciled." We have changed the terminology to avoid conflict with the use of "reconcile" for intentions.

$\mathrm{SP}(P_\alpha, \mathrm{GR}, \alpha, T_p, T_\alpha, \Theta_\alpha, \mathrm{IC}_\alpha)$

1. The group has a full shared plan:
(1) $[(\exists R_\alpha)\mathrm{FSP}(P_\alpha, \mathrm{GR}, \alpha, T_p, T_\alpha, R_\alpha, \Theta_\alpha, \mathrm{IC}_\alpha)]$ $\otimes$

2. The group has a partial shared plan, and a full shared plan to complete it:
(2) $[(\exists P_{elab}, T_{elab}, R_{elab})$
$(2a)$ $\mathrm{PSP}(P_\alpha, \mathrm{GR}, \alpha, T_p, T_\alpha, \Theta_\alpha, \mathrm{IC}_\alpha) \wedge$
$(2b)$ $\mathrm{FSP}(P_{elab}, \mathrm{GR}, \mathrm{Elaborate\_Group}(P_\alpha, \mathrm{GR}, \alpha, T_p, T_\alpha), T_p, T_{elab}, R_{elab}, \Theta_\alpha, P_\alpha)]$

Figure 1. Definition of SP (SharedPlan).

represent the beliefs agents have about their own and their collaborators' abilities to perform actions. The plan definitions and axioms use these meta-predicates within an embedding belief context so that omniscience is not required; a more extensive discussion of the reasons for the particular choice of meta-predicates may be found in our earlier paper [10]. $\mathrm{CBA}(G, \alpha, R_\alpha, T_\alpha, \Theta_\alpha)$ represents that the agent G is able to do the action α using the recipe R_α at time T_α under constraints Θ_α. CBAG is defined analogously for groups. Formal definitions of these meta-predicates are in the earlier paper.

Figure 1 gives the definition of SharedPlans showing the role of the Elaborate_Group action. When the agents' plan is only partial [the second clause], they must also have a second plan, which is complete (an FSP) for extending the parent partial plan [Clause (2b)]; that is, they must have agreed on the procedures they will follow to establish the additional beliefs and intentions needed to complete the parent plan. To explain this elaboration process, we must look at the definition of partial SharedPlan in some detail.

Figure 2 gives an informal definition of partial SharedPlan that will be the basis for our discussion. Because the full definition is complex, Figure 3 gives the general form of the definition with the relevant quantifier scopings for each of the major clauses. The formal definition, provided in Appendix A, uses a first-order logic augmented with several modal operators and meta-predicates. To enable readers to connect the intention-that discussion and new axioms to the formal definition, clauses in the two definitions correspond exactly. Table 1 provides a summary of the notation used in the formalizations in this paper. Table 3 in Appendix A lists the various operators, predicates and functions used in the formal definition, provides an informal description of their meaning, and indicates a reference for further information (either a section of this paper or the paper in which the original formalization is provided).

SharedPlans include as constituents individual plans formed by participants to

Notation	Meaning	Comments
α	action	Also: β_i,β_r, β_k, δ, γ
IC$_\alpha$	intentional context of α	
P_α	plan for α	
R_α	recipe for α	Also: R^f_α
R^p_α	partial (possibly empty) recipe for α	
T_i	time of i	i: action, plan, proposition
G	agent	Also G_i, G'
GR	group	Also GR$_i$
Θ_α	constraints of α	Also, ρ_j
$prop$	proposition	
A_{prop}	atemporal propositional content	

Table 1. Summary of notation used for special variables and constants.

do actions needed by the group activity. To distinguish among the different types of plans, the formalization defines five meta-predicates: FIP for *full individual plans*; PIP for *partial individual plans*; FSP for *full SharedPlans*; PSP for *partial SharedPlans*; and SP for SharedPlans of indefinite completeness. The plan meta-predicates make claims about the mental states of agents, but are not new modal operators. Each is defined in terms of intentions and beliefs of the agents who have plans of the given type [10].

As shown in the informal definition given in Figure 2, PSP's contain four main elements. Clauses 0 and 1 specify, respectively, the commitments required of the agents in the group to the joint action and the need for them to establish mutual belief in the existence of a (partial) recipe for the action. In the early stages of working together, many of the subactions of the partial recipe may be unresolved. The beliefs and intentions required for unresolved actions are represented in Clause 4. Clauses 2 and 3 represent the two possibilities for actions that have been resolved: either some group member (or subgroup) will do the action or the group will contract out the action.

In the Truckworld scenario, each of the members of TG must have an intention-that the group move material from Q to B (Clause 0); here "move" refers to the full complex of loading the material, transporting it, and unloading it at the destination. They have chosen a recipe (Clause 1) which comprises two shipment actions, each of which will move some of the material. The group has decided that the first shipment action, Shipm1 will be performed by the subgroup TG1, comprising MT$_1$, ST$_q$ and ST$_b$. Thus, Shipm1 is a "resolved" action and falls under the "core case" of the definition (Clause 2). To meet the PSP specification, TG1 must have a SharedPlan for doing Shipm1 (Clause 2b1). This subsidiary SharedPlan may be either partial or complete; for illustrative purposes, we assume TG1 has decided to use the recipe described earlier. (A specific instantiation of this recipe is given in the basic scenario in Figure 5.) In addition, the full group, TG, must mutually believe TG1 has a SharedPlan for Shipm1 (Clause 2b2) and that they can (successfully) complete it (Clause 2b3); MT_2, who is not in TG1, may not know the recipe TG1 has chosen, but must believe there is some recipe that group can use.[5] In addition MT_2 must have an intention-that TG1 will be able to do the Shipm1 action (Clause 2b4).

The locus of activity in the evolution of SharedPlans is in the transition from the "unresolved case" (Clause 4 of the definition), to the "core case" (Clause 2) or the "contracting" case (Clause 3).[6] The action resolution process for the group requires that they identify the agent(s) who will do the constituent action and obtain adequate commitments from group members. The member (or subgroup) selected to do an action must commit to doing it and the other agents must commit to the member's (subgroup's) being able to do it. In each of these cases, the individual agents must reconcile the intentions-to and intentions-that entailed by these new commitments with the other commitments they hold.

Clause 4 is also the principal way in which the PSP definition given here differs from the original definition. When designing the architecture for an automated collaborating agent, we realized that the original definition could be generalized. The Select_Agent and Select_Subgroup complex actions in the new definition allow for a wider range of strategies than the original definition, both for agent selection and for commitment formation.

In the Truckworld example, Shipm2 is unresolved and so falls under Clause 4. The formalization requires that each member of TG believe that some subgroup of TG can do Shipm2 (Clause 4a2a). In addition, TG must have a full SharedPlan to choose a subgroup to do this action (Clause 4a2b); that is, TG must have determined the procedure they will use to decide who will do Shipm2. To move Shipm2 from the unresolved to the core case (the job of the Select_Subgroup action), TG must agree on a subgroup, TG2, to do the action and form beliefs and intentions analogous to those described above for Shipm1. The members of TG2 do not need to know the recipe they will use, but they must at least agree on how they will figure this out (Clause 1).

The SharedPlan definitions also associate an intentional context parameter with each plan meta-predicate and intention operator.[7] The intentional context constrains replanning in the case of execution problems and guides recipe choice or completion. From a more theoretical standpoint, it provides the reason for an agent undertaking a plan and thus can be used to distinguish between two different uses

[5]The definition requires only that each agent intend-that this mutual belief holds at the time of the shipment. We discuss the reasons for this weaker constraint later.

[6]In this paper we do not discuss the contracting case further.

[7]In the original definitions, the context parameter also encoded constraints on the action. In the current definitions (see Appendix A) we have two separate parameters, IC_α and Θ_α, for intentional context and constraints respectively. The motivation for each of these parameters was given previously [10].

of the same action.[8]

Table 2 lists the intentional context for the meta-predicates and operators relevant to this paper. To simplify the presentation, in this table and in the subsequent text, we elide those arguments of operators that are not germane to the discussion. The table shows, for example, that the SharedPlan of a group to do α and the partial recipe they have constitute the intentional context for the intention-to do subsidiary acts (first line) and the recipe selection plan (fourth line). The last entry in the table gives the intentional context for an action γ that indirectly brings about some intended proposition, $prop$. It covers situations like contracting in which $prop$ results from an action, δ, done by another agent G_j (or, alternatively, by a group GR_j) and γ functions to get G_j (or GR_j) to do δ. In this case, both the intention-that for $prop$ and the intention to do δ (or the SharedPlan for δ if it is a group activity) are in the intentional context for γ.

The plan meta-predicates also have a constraints parameter that encompasses three types of constraints: execution preconditions, recipe-constraints, and constraints imposed by intention-conflict resolution. Typically, recipe-constraints concern time, location or other resources. For example, there is a time constraint between making dough and baking bread: the dough-making must precede baking. When reconciling intentions, an agent may discover it needs to further restrict the time, location or choice of other parameters of an action (e.g., only one of the family cars may be available for a trip to the movies). Constraints affect the selection of recipes for an action and may limit the resources available for doing subactions. Even though in general an agent may be able to perform an action, constraints may preclude its being able to do so in a particular situation.

Constraints of each type may derive from properties of a subaction β_i itself or from the embedding action α of which β_i is a constituent. However, not all the constraints on α will be constraints on β_i. Furthermore, there may be constraints on α that are not inherited by any of the actions it comprises. For instance, if an open-door action has the precondition *unlocked(door)* and two constituent subactions, *turn-handle* and *push*, then assuming doors have bolt locks that are separated from the door-handle mechanism, the precondition $unlocked$ is not a precondition of either *turn-handle* or *push*. The formalization uses the function $inherit$ to refer to the constraints that are inherited, the function $prec$ to refer to an action's preconditions, and the function con to refer to the constraints associated with an action.

3 INTENDING-THAT

The intentional attitude, intending-that, was introduced into the SharedPlan formalization to account for the commitment participants in a group activity make to

[8] For example, the intentional context would distinguish between Searle's two types of business school graduates [24, pp. 404-405].

Intention/SP	IC
Int.To$(G_k, \beta_i, ...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
Int.Th$(G_k, \text{Do}(\text{GR}_k, \beta_i, ...), ...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
PSP$(P_{\beta_i}, \text{GR}_k, \beta_i, ...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
FSP$(P_{select.rec.g}, \text{GR}, \text{Select_Rec_GR}(\text{GR}, \alpha, ...)...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
FSP$(P_{elab}, \text{GR}, \text{Elaborate_Group}(P, \text{GR}, \alpha, ...)...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
FSP$(P_{select.subgroup}, \text{GR}, \text{Select_Subgroup}(\text{GR}, \beta_k, ...)...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
FSP$(P_{select.agent}, \text{GR}, \text{Select_Agent}(\text{GR}, \beta_k, ...)...)$	SP$(\text{GR}, \alpha, ...)$, R^p_α
Int.Th$(G_j, \text{id.params}(\text{GR}, \alpha, T_\alpha), ...)$	SP$(\text{GR}, \alpha, ...)$
Int.Th$(G_i, \Theta_\alpha, ...)$	SP$(\text{GR}, \alpha, ...)$
Int.To$(G_i, \gamma, ...)$ where $prop$ is a direct effect of γ	Int.Th$(G_i, prop, ...)$
Int.To$(G_i, \gamma, ...)$ where $prop$ is an indirect effect of γ via δ	Int.Th$(G_i, prop, ...)$, Int.To$(G_j, \delta, ...)$/SP$(P_\delta, \text{GR}_j, \delta, ...)$

Table 2. Intentional Context for elements of SP$(P_\alpha, \text{GR}, \alpha, T_p, T_\alpha, \Theta_\alpha, \text{IC}_\alpha)$ with β_i in R^p_α.

one another's actions and to their joint activity. Intentions-that, like intentions-to, serve both to constrain the intentions an agent adopts and to affect its plan-based reasoning. In this section, we first review the basic properties of intending-that from the initial formalization. We then extend the treatment by specifying the minimal conditions for an agent to have an intention-that and providing additional axioms for the intention-that modal operator, Int.Th. Finally, we enumerate the ways in which an intention-that may lead an agent to adopt an intention-to do some action in service of that intention-that. The extensions presented here were motivated by the ways in which intentions-that lead agents to undertake actions that are required by their group activity, restrict the actions rational agents consider performing, and cause agents to adopt beliefs needed for collaboration. These extensions were constrained by our use of the formalization in the design of an architecture for automated collaborative agents.

3.1 An Overview of Intentions-That in SharedPlans

Intending-that is an attitude that holds between an agent and a proposition. Int.Th$(G, prop, T_i, T_{prop}, \text{IC}_{prop})$ represents an agent G's intention at time T_i that a certain proposition *prop* hold in the intentional context IC_{prop}.[9] In our formalization, propositions have a time [interval] associated with them. We denote the time of proposition $prop$ by T_{prop} and the atemporal propositional con-

[9] As discussed in our earlier paper [10], the significant distinction between intentions-to and intentions-that is "not in the types of objects each relates, but in their connection to means-ends reasoning and in their different presumptions about an agent's ability to act in service of the intention." [p. 281]. The reader is referred to that paper for a more extensive discussion of the differences.

tent by A_{prop}. For example, if it snows in Boston on Tuesday then $prop$ is $snow(Boston, Tues)$, T_{prop} is Tuesday, and A_{prop} is the atemporal fact of snowing in Boston. In most of the cases we are concerned with, T_{prop} is a time that is in, or extends into, the future. One class of propositions that play central roles within the context of SharedPlans are propositions that represent agents' abilities to perform actions (using CBA and CBAG). For the CBA and CBAG propositions, T_{prop} (i.e., the time at which the beliefs about ability must hold) is the time of the action that the ability proposition concerns.

Intentions-that play a central role in a collaborative activity from its very inception. A collaboration is initiated when an agent, G, moves from simply desiring that some action occur to intending that some group of agents, GR, carry out that action, i.e., to Int.Th$(G, \text{Do}(\text{GR}, \alpha, T_\alpha, \Theta_\alpha), T_p, T_\alpha, \text{IC}_\alpha)$. The intention-that leads the agent G to take action, as described below, to get the group GR to form a SharedPlan for α. If the other members $G_i \in$ GR agree, then they too will have an intention-that of the form Int.Th$(G_i, \text{Do}(\text{GR}, \alpha ...)...)$. Because each G_i may have different reasons for agreeing to participate in the group activity and thus in forming the SharedPlan, the intentional contexts for the different agents' intentions-that may be different.

In the Truckworld scenario, MT_1 might have been scheduled (e.g., by the plant manager) to do the job of moving material from Q to B. MT_1 might then have recognized that it would need help to do this job. MT_1 might have subsequently formed, on its own, an intention-that the group TG, comprising the set of agents MT_1, MT_2, ST_q, and ST_b, perform the action MOVE39. MT_1 would then contact the other three agents and convince them to form a collaboration. As a result, each of MT_1, MT_2, ST_q, and ST_b would then have an individual intention-that TG moves the materials.

The SharedPlan formalization employs Int.Th in five settings, listed in Figure 6. The intentions-that have the following three major functions:

Decision-making influence: all intentions-that influence which other intentions an agent will adopt.

Planning instigation: some intentions-that cause agents to complete their plans by changing either their mental state or the state of the world; as a result, commitment to the collaborative activity may lead to planning.

Collaborative acting assistance: some intentions-that lead agents to assist others in the group in performing actions relevant to the group activity or to alert them when things go wrong.

To demonstrate these functions, we will examine each of the five settings given in Figure 6 in turn.

Group activity commitment: Group activity commitment intentions-that (AC-IT's) exert decision-making influence by constraining the adoption of new intentions to ones that are compatible with the joint activity. The conflict-avoidance axiom in our previous paper [10] represents one way in which this occurs. For instance, once truck MT_2 joins the MOVE39 collaboration, it will ordinarily not take on a job that would hinder completion of MOVE39.[10]

AC-IT's may instigate a range of planning activities, because one way in which to satisfy an intention of the form Int.Th$(G_j, \text{Do}(\text{GR}, \alpha, ...))$ is to form a SharedPlan for doing the action α.[11] To form this SharedPlan, the agent and other members of the collaborating team must do an Elaborate_Group (see the SP definition in Figure 1). Each agent thus will undertake planning activities as part of its associated I_ELABORATE_GROUP. For example, ST_q's AC-IT for the MOVE39 action may lead it to various activities in pursuit of determining, with other group members, a recipe for doing MOVE39.

AC-IT's may lead agents to assist one another in their collaborative activity in two ways, by prompting helpful behavior and through cooperative replanning. Helpful behavior in the Truckworld domain is exemplified in Variation 1 of Figure 5: MT_1 recognizes that ST_q cannot work on MOVE39 for several hours because it is busy unloading another truck. MT_1 is not occupied; by doing some unloading it can lessen the wait for ST_q to begin participating in MOVE39. Because this will lower the overall cost of MOVE39 without otherwise affecting MT_1's performance, MT_1 helps out.

Cooperative replanning is exemplified in Variation 2: MT_1 successfully delivers the first shipment of rocks and heads back to home base; MT_2 breaks down and cannot complete delivery of the second shipment; Shipm2 thus reverts to being unresolved; MT_1 takes over and completes the delivery.[12] MT_2's communication to MT_1 is motivated by its AC-IT: because it cannot complete the drive subaction of Shipm2, it knows that Shipm2 and thus MOVE39 cannot be done according to the original recipe. Thus, it no longer believes that the Do-proposition embedded within its AC-IT holds. However, MT_2 can do something that will re-establish its belief in this Do-proposition, namely recruit someone else to complete the drive action. It does so, based on reasoning that follows the specification of Axiom 2 presented later in this paper. Because MT_2 cannot do the action, MT_1 is not assisting MT_2 but replacing it as the agent of the drive subaction of Shipm2. Thus, MT_1's assistance is motivated by its AC-IT commitment to MOVE39, not by a core-case intention that.

[10] That is, it will not do so unless it decides to opt-out of the collaboration [10].

[11] An agent might decide to engage in some other form of group acting.

[12] MT_1 and MT_2 may be able to do the replanning required without consulting ST_q and ST_b. However, ST_q and ST_b must be informed of the agent change.

Parameter identification: Parameter-identification intentions-that (PI-IT's) lead agents to perform actions that establish knowledge preconditions of the group activity [18].[13] The formalization specifies that all the agents must intend-that the parameters of the group action be identified [Int.Th(id.params)], not that each agent must intend to do some action that leads to the parameter being identified. Typically only some of the participants in a group activity will be involved in identifying the parameters; however all of them must be committed to the group knowing what they are. For example, ST_q might determine the time or place of the loading for Shipm1 and announce this to MT_1; ST_q thus does an action to determine the identity of this parameter, but MT_1 must also intend-that a time and place be determined and ST_q and MT_1 must, by communicating and negotiating as necessary, agree to the choices. Thus, all three of the functions of intentions-that may be invoked by PI-IT's.

Constraint satisfaction: Constraint-satisfaction intentions-that (CS-IT's) may engender planning actions to establish any of the three types of constraints—preconditions, recipe-constraints, or constraints imposed by intention-conflict resolution—or to maintain them during the construction and execution of a SharedPlan. The planning instigated by precondition constraints for SharedPlans is similar to that done in single-agent planning. However, the collaborative setting introduces the possibility that different agents or subgroups may handle different preconditions. It concomitantly requires that the agents coordinate and communicate about the assignment of responsibility for establishing preconditions. CS-IT's also exert decision-making influence: the need to have certain conditions hold for one action (e.g., the constraint that MT_1 be at the quarry at 10 AM for Shipm1) may preclude other actions (e.g., MT_1 working on a job in a different location at that time).

Core Case: Core-case intentions-that (CC-IT's) exert decision-making influence by restricting collaborating agents' choices of the ways to do constituent actions and the resources that are used so that plans for these mesh with each other and with the SharedPlan overall [4, 10]. TG2 will not plan to use ST_q to load MT_2 for Shipm2 until after ST_q has finished loading MT_1 for Shipm1. A second example is provided by Variation 3 in Figure 5. MT_1 has unloaded and is ready to refuel. It can stay where it is to refuel, but knows that if it does MT_2 will not be able to load until the refueling is done. Alternatively, it may move a short distance, thereby enabling MT_2 to begin unloading immediately.

CC-IT's are a major locus of motivation for agents to assist one another. If, as in Variation 4, MT_1 realizes that MT_2 is stuck enroute from Q to B and cannot complete its delivery without getting more fuel, and MT_1 believes it

[13] Lochbaum also argues for another knowledge precondition, $has.recipe$, but we treat $has.recipe$ directly in the plan definitions.

can provide that fuel on its way back to Q, a CC-IT will cause it to consider doing so.[14] Another form of collaborative action assistance is communicating to avoid conflicts in resource use (and hence conflicting intentions). For instance, if MT_1 believes MT_2 may need to refuel and there is only a limited amount of gas in the pump, then MT_1 may ask MT_2 about its needs, and, to the extent possible, adjust the amount of gas it uses to leave sufficient gas for MT_2.

Mutual belief satisfaction: MB satisfaction intentions-that (MB-IT's) serve a different type of planning function. They cause agents to communicate about their abilities to do actions and their beliefs about other agent's capabilities so that the mutual beliefs required for complete SharedPlans can be established. By requiring only an intention-that these mutual beliefs be established rather than the existence of the mutual beliefs themselves, the partial SharedPlan definition [Clauses (2a2 and 2b3)] allows for a more smooth evolution of partial SharedPlans. In particular, so long as no one in the group believes it is impossible for G_k to perform the action, and everyone believes G_k is committed to doing it, the definition classifies the action as resolved. If there ceases to be mutual belief of an individual's or subgroup's ability to perform an action, it is more effective to attempt to re-establish this mutual belief before seeking a new agent, because the agent selection process is more complex than the process of establishing mutual belief of ability—in particular in its need for negotiation and intention-conflict resolution.

The three functions of intentions-that—decision-making influence, planning instigation, and collaborative acting assistance—are central to efficient collaborative activity. The decision-making influence of intentions-that is to prevent agents from making commitments that would preclude their being able to do the group action. The other two functions of intentions-that cause agents to take action in service of their SharedPlans. These functions raise a formal question: what beliefs and intentions must an agent have when it has an intention-that so that only appropriate and useful intentions-to are generated? The next two sections address this question.

3.2 *Minimal Conditions for Intending-That*

Axioms in the original SharedPlan formalization [10] constrain an agent's intentions by requiring that the agent not hold conflicting intentions. The formalization imposes additional minimal constraints on the attitude of intending-to: for an agent to intend to do a complex action it must have some knowledge about determining a way to do the action, it must believe there is some recipe for the action that it can find and carry out, and it must be committed to expanding its partial plan for the

[14] Whether MT_1 will do so or not depends on its other obligations and the cost, as discussed elsewhere [10].

action to a complete plan. These requirements follow Bratman [2] in connecting intentions to means-ends reasoning; they also ensure that intending-to is closely related to acting.

This section provides an analogous constraint for intending-that: it specifies the minimal conditions on an agent's beliefs and intentions needed for it to hold an intention-that. To explain these constraints, we need to distinguish among three possible belief relations between an agent and a proposition $prop$: (a) at time T_{bel} the agent believes that *prop* holds: $\text{Bel}(G, prop, T_{bel})$; (b) at time T_{bel} the agent believes *prop* does not hold: $\text{Bel}(G, \neg prop, T_{bel})$; (c) at time T_{bel} the agent believes neither that $prop$ holds nor that it does not: $\neg\text{Bel}(G, \neg prop, T_{bel}) \wedge \neg\text{Bel}(G, prop, T_{bel})$.

Because propositions include time, case (b) represents the situation in which an agent holds the strong belief that it is impossible for the atemporal fact A_{prop} to hold at time T_{prop}. Cases (a) and (c) thus represent the only situations in which, according to the agent's beliefs, $prop$ might hold: in case (a) the agent believes it does hold and in case (c) there are some possible futures in which it might. The distinction of importance to SharedPlans is that between case (b) on the one hand and cases (a) and (c) on the other. We will say that an agent believes $prop$ is *impossible* in case (b) and *possible* otherwise.[15]

We assume that beliefs persist by default, but do not formalize this here (cf., [25]). However, we note that an agent's beliefs about $prop$ are not fixed. External events, in particular actions by another agent, may cause an agent to change its beliefs. Furthermore, an agent who does not have definite beliefs about $prop$ (case (c)), may seek information or take actions that will change its belief about $prop$.

We impose two constraints on an agent's holding an intention-that toward a proposition. First, the agent can only have an intention-that toward propositions it believes are possible. Our axiomatization does not allow an agent to intend-that an impossible proposition hold. Second, if the agent believes that the intended proposition *prop* is possible but does not currently believe it holds and knows some actions α_j it can take to cause proposition $prop$ to hold, then the agent must consider doing at least one of these actions.

Axiom 1: Restriction to Possible Propositions

The formalization constrains an agent's intentions-to do an action by requiring that the agent must be knowledgeable enough to do means-ends reasoning about the intended action. The first constraint on intending-that is analogous: an agent must believe it is possible for the intended proposition to hold.

[15] The incorporation of time in propositions and tying the possible/impossible classification to time is important for the formalization. For instance, what matters is that MT_1 will be able to carry rocks to the building site at a particular time. If the rocks must be transported today and MT_1 is only available tomorrow, then the action that is needed cannot be done by MT_1.

Axiom 1, given in Figure 7, represents this constraint; it stipulates that an agent cannot intend-that $prop$ if the agent believes $prop$ is impossible. There are two main reasons for this constraint. First, if an agent intends-that $prop$, it will consider this intention before adopting any additional intentions. If the agent believes that $prop$ is impossible, there is no benefit to avoiding actions or commitments that might make $prop$ impossible. Second, because having an intention-that toward $prop$ may, following Axiom 2, lead an agent to take actions that can directly or indirectly help to bring about $prop$'s holding, it is detrimental to have such an intention when $prop$ is impossible; there is no point in wasting energy on lost causes. AC-IT and CC-IT intentions-that provide examples of cases in which the restriction that Axiom 1 imposes functions to limit unproductive collaborative planning actions.

AC-IT's represent an agent's commitment toward the joint activity; it is detrimental to commit to impossible plans. More specifically, the commitment to the joint activity will lead an agent to undertake planning actions in service of this joint activity (from Axiom 2), including attempting to establish the minimal conditions for a SharedPlan (i.e., having collaborators and a plan for finding a recipe); there is no point in an agent's planning for an action if it believes the performance of that action is impossible (i.e., if it believes the Do proposition in the AC-IT is impossible). For example, if time constraints preclude MT_2's doing the driving for Shipm1 in the Truckworld scenario, then there is no point in MT_2 attempting to be part of a group that forms a SharedPlan to do Shipm1. Axiom 1 would prevent MT_2 from doing so because MT_2 does not believe that $\mathrm{Do}(\{\mathrm{MT}_2, \mathrm{ST}_q, \mathrm{ST}_b\}, \mathrm{Shipm1}, ...)$ is possible.

CC-IT's represent agents' commitments to each other's actions. If an agent G_j believes of one of its collaborators, G_k, that $\neg\mathrm{CBA}(G_k, \beta_i, T_p, T_{\beta_i}, \Theta_{\beta_i})$, then G_j believes it is impossible for G_k to perform β_i. G_j's belief may be based on G_k's general abilities to do an action of the type of β_i, or may be a result of the constraints on this performance of β_i, i.e., the Θ_{β_i}. As a result, G_j believes that the group activity based on a SharedPlan with the current recipe and agent choices cannot succeed. It only makes sense to continue to collaborate in doing α, if G_j and G_k and perhaps other collaborators revise their plan. Because a SharedPlan in which G_k performs β_i cannot succeed, there can be no such plan. In this situation, G_j should drop its intention-that G_k be able to do the action (the CC-IT). Axiom 1 stipulates that an agent in such a situation will not have an intention-that. For instance, given the time constraints above, there is no point in MT_1 forming an intention-that of the form $\mathrm{Int.Th}(\mathrm{MT}_1, \mathrm{CBA}(\mathrm{MT}_2, drive...)...)$ if MT_1, like MT_2, believes that $\mathrm{CBA}(\mathrm{MT}_2, drive...)$ is impossible. However, the intention-that the group do the higher level action of which β_i is a constituent is still valid, and in fact should lead G_j to inform its partners of the problem.

Axiom 2: Actions Prompted by Intentions-that

It is common to distinguish between desiring an action and intending to do an action [3]. Axiom 2 (Figure 7) enforces a similar distinction between desiring that a proposition hold and intending that the proposition hold. It ties an agent's intending-that $prop$ to actions that contribute to $prop$'s holding in a way similar to tying an agent's means-ends reasoning and intentions to do subactions to an agent's intending to do an action. In particular, if the agent intends-that $prop$, it must be willing to consider undertaking actions that will help bring $prop$ about.

Axiom 2 applies in the situation in which an agent is uncertain about whether proposition $prop$ holds (i.e., case (c) described above, $\neg$Bel$(G, prop, T_{bel}) \wedge \neg$Bel$(G, \neg prop, T_{bel})$),[16] either from a lack of knowledge alone (the agent just does not know whether $prop$) or, in the more interesting case, from circumstances in which the future is indeterminate with respect to $prop$. The agent may be in this state either initially when adopting the intention-that or later because some monitoring action indicates that $prop$ may no longer hold. If the future is indeterminate, then the agent believes that should certain events, $e_1, ..., e_k$, occur then $prop$ will hold, but other events may lead to $\neg prop$.

Axiom 2 stipulates that if an agent is uncertain about whether proposition $prop$ holds and believes there are some actions $\alpha_1, ..., \alpha_m$ any of which might lead to $prop$ holding (i.e., the α's each could cause one of $e_1, ..., e_k$ to occur), then it must either [Clause 3] intend to do, or be actively considering doing, one of these α's[17] or [Clause (4)] it must have considered all of the α's and determined that it could not do any of them.

Axiom 2 is of the form Clause 1 $\rightarrow$ [Clause 2 $\rightarrow$ (Clause 3 $\vee$ Clause 4)]. It uses the meta-predicate CONF which represents situations in which (a) the performance of an action conflicts with the performance of another action, or (b) the performance of an action conflicts with a proposition continuing to hold, or (c) two propositions cannot hold simultaneously. To simplify the statement of Axiom 2 we introduce the meta-predicate LEAD. LEAD$(G, \alpha, T_\alpha, prop, \Theta, T_i)$ summarizes the conditions under which, according to the agent's beliefs, an action α directly (Clause 2) or indirectly (Clause 3) leads to $prop$ holding. The parameter Θ encodes constraints under which the action α is done. For example, in Variation 4 of the Truckworld scenario (Figure 5), if MT_1 believes it can deliver fuel to MT_2 and that this delivery will enable MT_2 to continue driving to B, then MT_1's delivery-of-fuel action "leads to" TG2's being able to do (CBAG) Shipm2. One constraint on the delivery-of-fuel action is the time period in which it must be completed.

¿From the definition of the LEAD macro, Axiom 2 allows both for cases of α directly bringing about $prop$, and for cases in which α indirectly leads to $prop$

[16] The axiom does not include $\neg$Bel$(G, \neg prop, T_{bel})$ in the antecedent because the intention-that and Axiom 1 guarantee this.

[17] Clause (3a) repeats the antecedent of Clause 2 because the existential does not distribute over implication.

through performance of an intermediary action β by another agent or group of agents. In the indirect case, α ensures that the other agent (or group of agents) can do (CBA[G]) β and motivates it (them) to consider doing it (by adopting potential intentions-to or potential intentions-that).

Axiom 2 uses the attitude of potentially intending to do an action, Pot.Int.To, in Clause (3). Potential intentions represent an agent's mental state when it has not yet weighed a possible intention against existing commitments and alternative courses of action [10]. The axiom uses a disjunction of Int.To and Pot.Int.To (in Clause 3b) because we require agents to reconcile a potential new intention with existing intentions before adopting the new intention.[18] When the agent initially considers α, it must weigh doing α against its existing commitments. If it reconciles doing α successfully, then it will have an intention-to do α. The LEAD meta-predicate uses potential intentions-to and potential intentions-that in Clause (3) for analogous reasons.

To accommodate the need for agents to consider the costs of actions and other factors (e.g., an agent might be able to do something but only at the cost of losing its job), we use constraints to restrict the range of actions an agent must consider to ones that are "reasonable". In particular, limitations on cost or time may be encoded as constraints. For example, if MT_1 is committed to another, high-priority job immediately following delivery of the rocks to Q, then it may not be able to deliver fuel to MT_2.

Even after adopting an intention-to do α, the agent may not believe $prop$ will hold, because it may be unsure of its success in doing α.[19] If the agent drops its intention-to do α or if it fails in doing α, then to keep its intention-that $prop$, it must choose another action that could help, if there are any.

The final clause (Clause 4) of Axiom 2 is needed to avoid requiring an agent to consider repeatedly the same action.[20] From the disjunction, if Clause (4) holds, Clause (3) does not need to. Clause (4) states that the agent has reconciled all possible actions it could take and determined they each conflict in some way with other intentions. Thus, an agent needs to reconcile doing α only once; if it reconciles the potential intention-to do α, discovers a conflict, and decides to keep its current obligations and not to intend-to do α, then it does not need to reconsider α. For example, if MT_1 reconciles the delivery-of-fuel action, determines it conflicts with an obligation to be back at the quarry for another job, and decides it cannot do the delivery-of-fuel, then MT_1 should not consider this action further. In such

[18] In addition, turning a potential intention into a full fledged intention requires some preliminary means-ends reasoning [10].

[19] To have the intention-to, the agent must believe it will be able to perform the action; however, the success of its execution is a different matter.

[20] The last part of this clause is second-order: it quantifies over $prop$, because the axiom needs to cover conflicts doing α may have with an existing intention-that as well as with an existing intention-to. From the implementation perspective, this second-order characteristic does not necessarily present a problem; the formalization is a specification, not input to a theorem prover. One possible implementation strategy is to maintain a database of intentions-that and use it to check explicitly for conflicts.

cases, the agent will have neither an intention-to do α nor a potential intention-to do α, but it may still intend-that $prop$.[21]

This last clause is needed only when the agent has reconciled all the actions it possibly might do to help achieve $prop$ and has been unable to adopt an intention to do any of them. In particular, it is not needed if there is another α_i that the agent believes it can do and that "leads to" $prop$, because the existential may be satisfied with any action. For example, if MT_1 can pay someone to take fuel to MT_2 then the payment action satisfies the existential.

In situations that match Clause 4 of this axiom, the intention-that functions passively to keep the agent from doing anything that would conflict with $prop$, rather than actively leading the agent to do something. For example, MT_1 would not adopt an intention to do some action that interfered with another truck delivering fuel to MT_2.

Axiom 2 applies to planning actions as well as to domain actions; planning actions can provide an indirect way to make a proposition true. Thus, this axiom is a mechanism for realizing the planning-instigation function of intentions-that.

3.3 Getting from Intentions-that to Intentions-to

Intentions-that play both static and active roles in constraining rational behavior. The static role, realized in intention conflict avoidance axioms [10], serves to prohibit an agent from committing to contradictory states of affairs. The active role, which we consider in this section, serves to generate actions that help bring about the intended proposition. The active role of intentions-that requires means-ends reasoning analogous to that for intentions-to. We use the term Cultivate to refer to the complex activity of doing this reasoning.

There are three different sources of an agent adopting an intention-to in service of an intention-that. First, as represented by Axiom 2, the agent may intend-to do an action that will bring about, or help bring about, the intended proposition. Second, the agent may adopt an intention to do an action that will help another agent when the intended proposition concerns an action by that second agent.[22] For example, MT_1's intention-that TG2 be able to deliver the second shipment of rocks to B may cause MT_1 to radio road-condition information to TG2 (or, if it has sufficient information to know that MT_2 is doing the driving, then just to MT_2) to make the trip shorter and the Shipm2 action overall more efficient (Variation 5 in Figure 5). Third, even when an agent believes that $prop$ holds, if the time T_{prop} has not passed yet, the intention-that $prop$ may generate actions to ensure that $prop$ continues to hold. Monitoring actions are instances of such behavior. Variation 6 provides an example: if ST_q has access to information sources unavailable

[21] More specifically, it can keep the intention-that so long as the possibility of doing α was not the sole reason it believed $prop$ was possible. The agent must still believe that $prop$ is possible.

[22] Axioms that directly tie helpful behavior to a SharedPlan or contracting situation appear in our previous paper [10].

to MT_1 once enroute, then ST_q might monitor road conditions and radio important information to MT_1.

Helpful behavior, the second way in which an agent's intention-that may lead to an intention-to, arises when an agent G_1 intends-that a proposition $prop$ holds, $prop$ relates in some way to the performance of an action β by another agent (G_2), and G_1 believes that it will be easier for G_2 to do β if G_1 does α. In the Truckworld example of the preceding paragraph, $prop$=CBA$(MT_2, \text{drive_to_B}(MT_2)...)$, $\beta =$ drive_to_B(MT_2), and α=radio_road_info(MT_1). In general, we consider $prop$ relevant to G_2's doing β, if $prop$ refers to G_2's performance of β or ability to do β. The two situations we model formally are $Do(G_2, \beta...)$ and CBA$(G_2, \beta...)$ (or CBAG$(G_2, \beta...)$).[23]

An agent may monitor a proposition actively or passively. ST_q's monitoring of road conditions exemplifies active monitoring; the agent actively does some action to track whether the proposition holds. In some situations, an agent may be able to rely on someone else to alert it if a proposition changes. For instance, ST_b may assume that MT_2 will let it know if something goes wrong and MT_2 cannot complete the driving subaction of Shipm2. Passive monitoring is daemon-like: notification triggers an adjustment in the SharedPlan. Passive monitoring is less costly than active monitoring (because the agent does not need to do anything unless a condition changes), but cannot always be relied upon. However, a participant in a SharedPlan can assume its collaborators will let it know if a proposition relevant to the SharedPlan (e.g., a CBA or an Int.To) ceases to hold. As a result, passive monitoring can be used for these intentions-that.

4 SHAREDPLAN-BASED ARCHITECTURE FOR COLLABORATIVE-AGENT DESIGN

In this section, we present an architecture for collaborative-agent design based on the SharedPlan formalization. This architecture presumes as a base an architecture for an individual, resource-bounded rational agent that has capabilities for scheduling and executing intended actions and for monitoring their effects. A fundamental characteristic of the design of resource-bounded rational agents is that mental actions and domain actions must be treated in a coordinated manner. The agent must be able to interleave thinking (in particular, planning) and acting. At some level of abstraction a single scheduler handles an agenda including both mental and domain actions. Furthermore, the agent architecture must include some basic mechanisms for managing pending and adopted intentions. A variety of designs for such agent architectures have been proposed in current planning architectures and formalizations [23, 5, 25, inter alia].

In this section we address the problem of how to go from the SharedPlan for-

[23] There are additional motivations for helpful behavior, but those are beyond the scope of this paper.

malization to an implementation that embodies it in an automated rational agent. The specification of collaboration provided by SharedPlans could be used in one of three ways to produce a system: (a) as input for a theorem-proving system; (b) for developing a system tailored to a specific domain in which the system designer decides which components to reason explicitly about and which to hardwire in; or, (c) to develop general algorithms that realize the specifications. We have taken the last approach which we believe to be the most promising. In this section, we describe the most distinctive features of the architecture. We have used this architecture in the implementation of a simple multi-agent system [20] in the Truckworld environment [19] and are currently exploring its use in the implementation of a commercial-exchange system and a collaborative human-computer interface system.

Two processes naturally emerge from the SharedPlan formalization as central to controlling an agent's behavior in accordance with its intentions. The first, ELABORATE_INDIVIDUAL, is responsible for the means-ends reasoning that derives from an intention-to. This process and its relationships to intention-to have been described previously [10]. The second, CULTIVATE, plays an analogous role for intention-that. It both monitors for intention conflicts and initiates an agent's I_ELABORATE_GROUP process. Because SharedPlans are driven off of intentions-that, CULTIVATE plays a central role in the architecture and we describe it before looking at the SharedPlan subsystem and the I_ELABORATE_GROUP processes that CULTIVATE spawns for some intentions-that within the SharedPlan context.

4.1 The cultivate process

CULTIVATE determines the actions that an agent might take as a result of having an intention-that and launches the processes needed for the agent to take (consider taking) them (i.e., creates potential-intentions to do them). It embodies the axioms for intending-that given in our earlier paper. The main job of CULTIVATE for Int.Th$(G, prop, ...)$ is to identify actions an agent can take to help establish or maintain the proposition $prop$.

Figure 8 shows the main constituent steps of the CULTIVATE process. The *Cultivate-Act* branch applies when the agent [currently] believes that the intended proposition, $prop$, does not hold; the actions an agent takes depend on the particular content of $prop$, and possibly on the domain. The *Cultivate-Help* branch applies for propositions that contain some element referring to an action α by another agent or subgroup, e.g., Do$(G_2, \alpha ...)$; in these cases, even if G_1 believes $prop$ holds, it may be able to assist G_2 and in some way make it easier or less costly to do α. Again, the actions depend on $prop$ and α. The *Cultivate-Monitor* branches are motivated by situations like the road-condition monitoring examples described earlier.

In the remainder of this section we discuss the Cultivate-Act branch of the algorithm as it applies for the cultivate processes associated with the intentions-that in the SharedPlans definitions. We leave the discussion of the Cultivate-Help and Cultivate-Monitor branches to future work.[24]

The five settings for intentions-that in SharedPlans listed in Figure 6 engender different kinds of CULTIVATE behavior. We discuss each type in turn.

Group activity commitment: If the intention-that is of the AC-IT type, Int.Th$(G, \text{Do}(\text{GR}, \alpha, T_\alpha, \Theta_\alpha), T_p, T_\alpha, \text{IC}_\alpha)$, then the main action an agent can take is to recruit the other members of the group to work on doing α and, in particular, to begin the formation of a SharedPlan. Hence, the main subprocesses the CULTIVATE process spawns are ones to establish that all other members, G_j, of GR also hold this intention [i.e., $(\forall G_j \in \text{GR})$Int.Th$(G, \text{Do}(\text{GR}, \alpha, T_\alpha, \Theta_\alpha), T_p, T_\alpha, \text{IC}_\alpha)$] and to establish mutual belief among members of GR of everyone's doing so. Because to form a SharedPlan (for doing α) group members must form a full SharedPlan to do the Elaborate_Group action, the CULTIVATE process must also initiate the agent's adoption of an intention-that the group will undertake this planning activity. Thus, the CULTIVATE process for an intention-that related to a domain action spawns an intention-that (and hence a CULTIVATE process) for a mental (planning) action. This second CULTIVATE process spawns an I_ELABORATE_GROUP process. A collaborating group's Elaborate_Group is realized by the sum total of the participating agents' I_ELABORATE_GROUP's spawned in this way.

In our architecture, we treat three instances of AC-IT specially, those associated with the full SharedPlan for selecting a recipe, selecting an agent(s) and elaborating a SharedPlan. These cases correspond to the I_ELABORATE_GROUP, I_SELECT_REC_GR, I_SELECT_AGENT, I_SELECT_SUBGROUP processes described in Section 4.2. For each of these cases, CULTIVATE directly initiates the appropriate special purpose process. Treating these AC-IT cases specially enables automated collaborative agents to save on communication and limits the level of recursion in SharedPlans. The advantage of otherwise treating them uniformly with domain actions is that doing so enables a single locus for scheduling and dealing with conflict avoidance.

Core Case: In the Core Case intentions-that (CC-IT), the proposition concerns the ability of another agent(s) to do an action; i.e., it is of the form Int.Th$(G, \text{CBA}(G_k, \beta_k, ...), ...)$ or Int.Th$(G, \text{CBAG}(\text{GR}_k, \beta_k, ...), ...)$. In these cases, CULTIVATE first determines whether there is some γ that

[24] As the axioms for helpful behavior in an earlier paper illustrate [10], an agent's decisions about what actions to take to help its collaborators not only refer to domain specific knowledge of action relationships, but also must take into account the cost of actions. The development of cost functions of actions and hence Cultivate-Help are beyond the scope of this paper, as is a discussion of monitoring issues and processes.

can lead (directly or indirectly) to CBA (CBAG). If it finds such a γ, it then begins the process for establishing an intention-to do γ.[25] The identification of a γ depends on the domain and the agent's reasons for currently believing that G_k cannot bring about β_k. The example of MT_1 radioing back to TG2 information about road conditions is an instance of the result of this kind of deliberation by a CULTIVATE process.

Parameter identification A variety of strategies can be adopted for parameter identification in SharedPlans,[26] depending on the relationships among group members. Thus we do not give a full specification here, but some possibilities are listed in Figure 9. Two general constraints apply to any strategy. First, the choice of the entity that fills a particular role in an action may depend on the subactions in which the parameter also participates. For example, the time of Shipm1 cannot be finally set until the agents know when ST_q and ST_b can do their parts of the loading and unloading actions. As a result, parameter identification strategies benefit from delaying commitment until as much recipe and agent selection as possible has been completed. This benefit must be balanced against the need to act before the planning process is completed. Second, the agents must have a mechanism for coordinating the choice of parameters for an action with the choice for any subsidiary actions in the recipe (whether single- or multi-agent) that use that parameter. Because handling the general problem of distributed parameter binding is an open issue in Computer Science, our implementations impose a strict ordering and communication regime as a short-term solution.

Constraint satisfaction: The CULTIVATE process associated with an intention-that constraints will be satisfied, (Int.Th$(G_j, \Theta_\alpha, T_p, T_{\Theta_\alpha}, P_\alpha)$), needs to handle the three types of constraints described earlier: execution preconditions, recipe-constraints, and constraints imposed by intention-conflict resolution. For precondition constraints CULTIVATE needs to start processes that coordinate with other agents; if the agent commits to satisfying a precondition, CULTIVATE spawns the corresponding intentions-to. For recipe constraints, CULTIVATE monitors the satisfaction of the constraints and is responsible for coordination with other agents. Handling the general problem of distributed constraint satisfaction is an open issue in Computer Science and beyond the scope of this paper.

Mutual belief satisfaction: The CULTIVATE process for these intentions-that over-

[25] This process begins with a reconciliation to determine whether the intention is compatible with other commitments.

[26] Individual plans also require parameter identification which is realized in our formalization through an intention-that. However, in this case, the agent acts independently and the CULTIVATE process need only spawn subsidiary elaboration processes for intentions-to do actions that will identify each of the parameters.

sees the agent's adoption of the belief and causes communication sufficient for the establishment of mutual belief. If an agent decides not to adopt the belief, then the CULTIVATE process must communicate this to the other agents.

4.2 SP Subsystem

Figure 10 shows the basic architecture of the SP subsystem (adapted from [20, Figure 8]). Each SharedPlan that an agent participates in will yield an instantiation of the processes portrayed in this figure. The major constituent processes in the subsystem are[27]

- Plan-completion process: I_ELABORATE_GROUP.
- Selection processes: I_SELECT_REC_GR, I_SELECT_AGENT and I_SELECT_SUBGROUP.
- Intention-management processes: CULTIVATE, ELABORATE_INDIVIDUAL, RECONCILE.

I_ELABORATE_GROUP is responsible for a participating agent's work in helping to create and carry out a group activity. Two major constituents of any Elaborate_Group action are those for selection of a recipe for doing the group action, which is realized by the I_SELECT_REC_GR processes of the collaborating agents, and for determining which agent(s) will do the constituent actions of the chosen recipe, which is realized by the I_SELECT_AGENT and I_SELECT_SUBGROUP processes of the collaborating agents.

The I_ELABORATE_GROUP, I_SELECT_AGENT, and I_SELECT_SUBGROUP processes each correspond to a requirement in the formalization that the group have a full (i.e., complete) SharedPlan for some complex planning action. I_ELABORATE_GROUP corresponds to the requirement in the SP definition (Figure 1) that the group have a full SharedPlan to do an Elaborate_Group; this process is the way in which the agent works toward helping to complete the partial SharedPlan. I_SELECT_REC_GR corresponds to the requirement in the PSP definition (Figure 2) that the group have a full SharedPlan to do a Select_Rec_GR; and I_SELECT_AGENT and I_SELECT_SUBGROUP correspond to the requirement in the PSP definition that the group have a full SharedPlan to do a Select_Agent or Select_Subgroup as appropriate. For each of these full SharedPlans the agent must have adopted an AC-IT intention-that which has an associated CULTIVATE process. As described above (Section 4.1), the CULTIVATE spawns the corresponding

[27] We have omitted the CULTIVATE and RECONCILE processes in this figure to avoid cluttering the diagram.

I_ELABORATE_GROUP, I_SELECT_AGENT, and I_SELECT_SUBGROUP. We discuss some details of the other processes in the next section.

4.3 Plan Completion Processes

I_ELABORATE_GROUP is active throughout the time an agent participates in the performance of a group action. If the group succeeds then the I_ELABORATE_GROUP will have taken the agent from its initial intention-that to having the full set of beliefs and intentions needed for it to carry out its part in the group activity. Furthermore, because plans may become more partial (e.g., if some agent is unable to do an action it committed to or a recipe does not work out), I_ELABORATE_GROUP remains active until the group successfully completes the action. In our implementations, the associated CULTIVATE process also tracks success or failure. I_ELABORATE_GROUP can be terminated prematurely either if the agent decides to drop out of doing the multi-agent action α (i.e., drops its AC-IT intention-that) or the group elaboration fails for some reason.

I_ELABORATE_GROUP initiates the following processes or monitors them for success or failure:

- Working with the group to determine a way to do the action I_SELECT_REC_GR, either by choosing a known recipe or by constructing a recipe.
- For each subaction in the recipe working with the group to identify agent(s) to do that action (I_SELECT_AGENT or I_SELECT_SUBGROUP)
- Doing its part to identify parameters of the action and meet constraints on doing the action (e.g., satisfying preconditions)

In addition, I_ELABORATE_GROUP is responsible for establishing and maintaining the mutual beliefs required by the FSP definition. In most cases I_ELABORATE_GROUP will cause some kind of communication among agents. However, in those cases in which the definition of PSP requires only an Int.Th$(G, \text{MB}...)$, I_ELABORATE_GROUP will spawn cultivate processes for these intentions.

4.4 Selection processes

The recipe-selection and agent-selection processes I_SELECT_REC_GR, I_SELECT_AGENT, and I_SELECT_SUBGROUP are more complicated than their individual-agent plan counterparts because of the need to coordinate with other agents and reach agreement. The flexibility that characterizes the SharedPlan formalization adds complexity both by not requiring a central manager and by allowing for plans to be partial, including allowing for plans to become more partial because of execution failures or resource limitations that lead agents to change their intentions.

Selecting Recipes

When an agent acts alone, it figures out how to do an action (i.e., finds a recipe) and then carries out the necessary subacts. When agents work together, both the formation or selection of the recipe and the carrying out of the subactions is distributed. The group must reach consensus on how they are going to perform the action (i.e., on the recipe they will use) and on who is going to do the subactions. A major advantage of group action—that an agent does not have to do everything itself—is counterbalanced by the necessity of agents agreeing on their plan and coordinating their activities. Thus, recipe and agent selection in the collaborative setting require interagent negotiation and communication as well as individual agent reconciliation.

The input to an I_SELECT_REC_GR process is the multi-agent action to be done by the group. The I_SELECT_REC_GR process terminates successfully when the group has agreed on a recipe for this action. An agent's basic planning and reasoning processes are important constituents that are used to figure out what the agent knows of how to do the multi-agent action α. In addition, the I_SELECT_REC_GR process requires capabilities for participating in multi-agent decision making.

In the simplest cases, each agent might put forward its favorite recipe; then the agents might vote to choose one of them. (Our implementations so far have taken this approach.) A more complex decision-making scenario is one in which no single agent has a complete recipe. The agents work together to combine pieces of recipes that they individually discover.

Because agents only know that a recipe has worked once they complete performing an action, the I_SELECT_REC_GR process must save enough of its decision making information to allow a new recipe to be selected, should the one selected by it fail. When the I_SELECT_REC_GR terminates it passes back to the spawning I_ELABORATE_GROUP either success and this decision-making record, or a failure and perhaps the reason for failure.

Selecting Agents

The I_SELECT_AGENT and I_SELECT_SUBGROUP processes are similar to I_SELECT_REC_GR in incorporating decision-making and negotiation strategies that enable an agent to form a consensus with others in the group. They are more complex because they also include processes for considering whether the agent itself can do (or participate in doing) an action, forming commitments to other agents' actions, and forming beliefs about other agents' abilities to do actions.

If the action about which the group is deliberating is a single-agent action, then each participating agent's I_SELECT_AGENT must first consider whether the agent can do the action by determining whether it has the capabilities to do it and by reconciling a potential intention to do it. If the agent is selected, its I_SELECT_AGENT will have to spawn the processes needed to form an intention to do the action.

These subproceses include those for adopting a commitment to doing the action and for forming a partial individual plan for the action. To do the latter, the agent must have sufficient knowledge to determine a way to do the action. If the agent is not selected, then its I_SELECT_AGENT must spawn processes for adopting intentions that the selected agent can do the action. In both cases it must spawn processes for establishing that the group mutually believe the agent is capable of doing it.

The case is similar if the action about which the group is deliberating is a multi-agent action. The main difference is that the I_SELECT_SUBGROUP must handle the problems of subgroup formation. In particular, it must incorporate processes for deciding whether to participate in a group and processes for initiating a subsidiary SharedPlan.

A further complication arises for both I_SELECT_AGENT and I_SELECT_SUBGROUP because rational-agent architectures require that planning and acting be interleaved. As a result, the formalization provides for groups to act on partial recipes. In particular, they may choose agents for a particular action who then might start performing the action before the construction of the full recipe has been completed. An action may regress to the unresolved case if the selected agent or subgroup is unable to perform it successfully. Thus, both I_SELECT_AGENT and I_SELECT_SUBGROUP must be able to restart with information about past attempts. To provide for this possibility in the architecture, we make the agent's I_SELECT_AGENT and I_SELECT_SUBGROUP processes children of both the I_SELECT_REC_GR and I_ELABORATE_GROUP processes. They are spawned by I_SELECT_REC_GR and report success or failure to the I_ELABORATE_GROUP process. Furthermore, if no other agent or group can take over, the agents may need to find another recipe; thus I_SELECT_REC_GR must also be reentrant.

5 CONCLUSIONS

This paper expands the SharedPlan formalization presented earlier [10] by providing a specification of the minimal conditions for an agent to have an intention-that, describing the ways in which having an intention-that can lead to taking action, and describing the basic processes in the architecture of a rational, resource-bounded agent required for that agent to participate in group planning. Two areas in which further research is required are the specification of group decision making processes (e.g., for agent and recipe selection) and the development of intention reconciliation processes that can treat a combination of group-directed and individual-directed intentions.

6 ACKNOWLEDGEMENTS

Support for the first author was provided by Grant No. IRI-9525915 and Grant No. CDA 94-01024 from the National Science Foundation. Support for the second author was provided by NSF Grant No. IRI-9423967. Hanna Pasula did the initial implementation of SharedPlans in the Truckworld domain. We acknowledge also her contribution to the research: many of her questions about the implementation led us to revise the formalization and modify the architecture. We also thank Luke Hunsberger and Charles Ortiz for comments on earlier versions of the paper.

REFERENCES

[1] Cecile T. Balkanski. Modelling act-type relations in collaborative activity. Technical Report 23-90, Harvard University, 1990.

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

[3] Michael E. Bratman. What is intention? In P.R. Cohen, J. Morgan, and M.E. Pollack, editors, *Intentions in Communication*, chapter 2, pages 15–31. MIT Press, Cambridge, MA, 1990.

[4] Michael E. Bratman. Shared cooperative activity. *The Philosophical Review*, 101(2):327–341, 1992.

[5] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(4):349–355, 1988.

[6] T.L. Dean and M. P. Wellman. *Planning and Control*. Morgan Kaufman, Publishers, California, 1991.

[7] Barbara Grosz and Sarit Kraus. Collaborative plans for group activities. In Ruzena Bajcsy, editor, *Proceedings of the 1993 International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 367–373, San Mateo, CA, 1993. Morgan Kaufmann Publishers, Inc.

[8] Barbara Grosz and Candace Sidner. Plans for discourse. In P. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*, pages 417–444. Bradford Books/MIT Press, Cambridge, MA, 1990.

[9] Barbara J. Grosz. Collaborative systems: 1994 aaai presidential address. *AI Magazine*, 2(17), Summer 1996.

[10] Barbara J. Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[11] P. Haddawy. *Representing Plans under Uncertainty: A Logic of Time, Chance and Action*. PhD thesis, University of Illinois, 1991. University of Illinois Tech. Report UIUCDCS-R-91-1719.

[12] Steve Hanks, Dat Nguyen, and Chris Thomas. A beginner's guide to the truckworld simulator. Technical Report 93-06-09, Department of Computer Science and Engineering, University of Washington, 1993.

[13] Nick R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence Journal*, 75(2):1–46, 1995.

[14] L. Kaelbling. An architecture for intelligence reactive systems. In M. Georgeff and A. Lansky, editors, *Reasoning about Actions and Plans*. Morgan-Kaufmann, Los Altos, California, 1987.

[15] H. Levesque, P. Cohen, and J. Nunes. On acting together. In *Proceedings of the Annual Conference of the American Association for Artificial Intelligence (AAAI-90)*, pages 94–99, 1990.

[16] Karen Lochbaum. *Using Collaborative Plans to Model the Intentional Structure of Discourse*. PhD thesis, Harvard University, 1994. Available as Tech Report TR-25-94.

[17] Karen Lochbaum, Barbara Grosz, and Candace Sidner. Models of plans to support communication: An initial report. In *Proceedings of AAAI-90*, pages 485–490, Boston, MA, 1990.

[18] Karen E. Lochbaum. The use of knowledge preconditions in language processing. In Chris S. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*, volume 2, pages 1260–1266, San Mateo, CA, 1995. Morgan Kaufmann Publishers, Inc.

[19] D. Nguyen, S. Hanks, and C. Thomas. The TRUCKWORLD manual. Technical Report TR 93-09-08, Dept. of Computer Science and Engineering, Univ. of Washington, 1993.
[20] Hanna Pasula. Design of a collaborative planning system. Harvard University, Senior Honors Thesis, 1996.
[21] Martha E. Pollack. Plans as complex mental attitudes. In P.N. Cohen, J.L. Morgan, and M.E. Pollack, editors, *Intentions in Communication*. Bradford Books, MIT Press, 1990.
[22] C. Rich and C. L. Sidner. Adding a collaborative agent to direct-manipulation interfaces. Technical Report 96-11, Mitsubishi Electric Research Laboratories, Cambridge, MA, May 1996.
[23] S. J. Russell and E. H. Wefald. Principles of meta-reasoning. In *Proceedings of the First International Conference of Knowledge Representation and Reasoning*, pages 400–411, 1989.
[24] John R. Searle. Collective intentions and actions. In *Intentions in Communication*, chapter 19. The MIT Press, 1990.
[25] Y. Shoham. Agent oriented programing. *Artificial Intelligence*, 1(60):51–92, 1993.
[26] S. Zilberstein and S.J. Russell. Optimal composition of real-time systems. *Artificial Intelligence*, 82 (1-2):181–213, 1996.

A FORMAL DEFINITION OF PARTIAL SHAREDPLAN

PSP(P_α, GR, α, T_p, T_α, Θ_α, IC_α)

0. The group GR has mutual belief that all members of the group are committed to the group doing α, i.e., that they each **intend-that** the group do α.

0a. Parameter identification: The group GR has mutual belief that all members of the group are committed (have an **intend-that**) to the identification of parameters necessary to α.

0b. Constraint satisfaction: The group GR has mutual belief that all members of the group are committed (have an **intend-that**) to making sure that the constraints for doing α (Θ_α) will hold.

1. The group GR mutually believe that there is a **recipe** for α, but their recipe for doing α may be partial; e.g., they may only have identified *some* of the subactions that need to be performed. If their recipe is only partial, they have a FSP to complete their partial recipe (using Select_Rec_GR).

For each subaction β_i in the partial recipe, one of (2) through (4) holds:

2. Core case (resolved actions β_r):

2a. Single-agent subaction: A member, G_k, of the group intends to do the subaction β_r; G_k may have either a complete or partial plan for doing it.

2b. Multi-agent subaction: A subgroup GR_k has a shared plan (SP) to do the subaction, but this plan may be only partial.

2b1. The subgroup GR_k has a SharedPlan for the subaction.

2b2. The group GR mutually believe that the subgroup GR_k has a SharedPlan for the subaction.

2b3. Every member of GR **intends-that** the group GR mutually believe that the subgroup GR_k is able to do the subaction.

2b4. The group GR mutually believe that all of its members are committed to the subgroup's success, i.e., **intend-that** the subgroup GR_k can do the subaction.

3. Contracting case: (omitted here; see [10])

4. Unresolved case (actions β_k): GR has not fully deliberated about the subaction β_k; no decision has been made about which agent(s) will do it.

4a1. Single-agent subaction:

4a1a The group GR mutually believe that some (perhaps as yet unidentified) group member G_k can do the subaction.

4a1b. The group GR has a full SharedPlan to select an agent to do the subaction (Select_Agent).

4a2. Multi-agent subaction:

4a2a. The group GR mutually believe that some (perhaps as yet unidentified) subgroup, GR_k, can do the subaction.

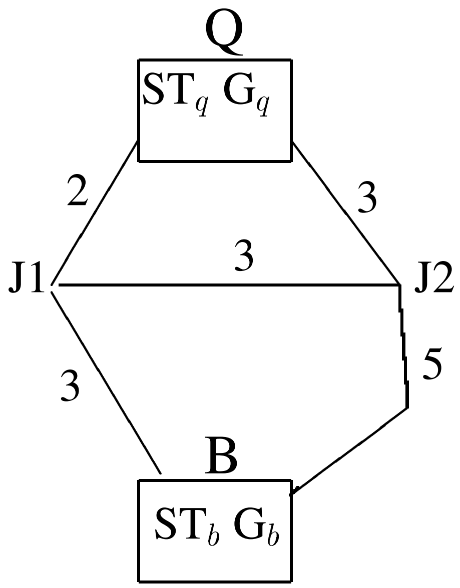
4a2b. The group GR has a full SharedPlan to select a subgroup to do the subaction (Select_Subgroup).

Figure 2. English description of the PSP (partial SharedPlan) definition.

General form:
Clause 0 $\wedge$ Clause 1 $\wedge$
[Clause 2 $\otimes$ Clause 3] $\wedge$
Clause 4

Detailed form:
Clause 0 $\wedge$
$(\exists R_\alpha, \mathrm{B}, \mathrm{B}_{res}, \mathrm{B}_{unres}, \mathrm{C})[$
Clause 1 $\wedge$
$(\forall \beta_r \in \mathrm{B}_{res})$[Clause 2 $\otimes$ Clause 3] $\wedge$
$(\forall \beta_k \in \mathrm{B}_{unres})$Clause 4
]

Figure 3. The form of the PSP definition



Two types of trucks agents:

ST: stationary truck with crane arm for [un]loading only.
MT: mobile truck with arms that can [un]load

Two sites:

Q: quarry with raw materials [e.g., big rocks], a stationary truck, ST_q, and a gas pump, G_q.
B: building construction site, a stationary truck, ST_b, and a gas pump G_b.

Routes: The layout of the roads between Q and B is shown above. Route distances are given in the rightmost column.

R1:	Q → J1 → B	5 units
R2:	Q → J2 → B	8 units
R3:	Q → J1 → J2 → B	10 units

The road between J1 and B is sometimes closed. If so, the shortest route from Q to B (R1) is unavailable.

Actions:

Multi-agent actions: Load truck, Unload truck
Load truck requires one ST and one or more MTs. Unload truck requires two trucks (either two MTs or one MT and one ST). Loading is slower than unloading.

Single-agent actions: All Truckworld basic-level actions (e.g., drive a route, send-info by radio, receive-info by radio; also [un]loading related actions: pick-up, put-down, move).

Figure 4. The Truckworld Domain.

Basic scenario: Mobile truck MT_1 moves the first shipment of rocks from Q to B along R1 if possible; ST_q helps load, ST_b helps unload; MT_2 does the transport for the second shipment.

Variation 1: Helpful behavior. MT_1 helps ST_q unloading another truck to lessen the wait for ST_q to begin participating in MOVE39.

Variation 2: Cooperative replanning. MT_2 breaks down doing Shipm2; MT_1 takes over so that Shipm2 and hence MOVE39 can be done successfully.

Variation 3: Intention-conflict avoidance. MT_1 moves from ST_b's unloading area while getting gas so that MT_2 can unload.

Variation 4: CBA-enable. MT_2 runs out of gas; MT_1 can help by bringing gas.

Variation 5: CBA-helpful behavior. Truck MT_1 senses road closure at J1 and radios this back to subgroup TG2 (or directly to truck MT_2) which is at Q so that MT_2 will take R2 and not waste time.

Variation 6: Monitoring. ST_q monitors road conditions and forwards any important information to MT_1 or MT_2 while they are enroute and unable to obtain this information independently.

Figure 5. Scenarios for the Truckworld domain.

Group activity commitment (AC-IT): (Clause (0))

$$\text{Int.Th}(G_j, \text{Do}(\text{GR}, \alpha, T_\alpha, \Theta_\alpha), T_p, T_\alpha, \text{IC}_\alpha)$$

Parameter identification (PI-IT): (Clause (0a))

$$\text{Int.Th}(G_j, \text{id.params}(\text{GR}, \alpha, T_\alpha), T_p, T_\alpha, P_\alpha)$$

Constraints satisfaction (CS-IT): (Clause (0b))

$$\text{Int.Th}(G_j, \Theta_\alpha, T_p, T_{\Theta_\alpha}, P_\alpha)$$

Core Case (CC-IT): (Clauses (2a3) and (2b4))

$$\text{Int.Th}(G_j, (\exists R_{\beta_r})\text{CBA}(G_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \text{inherit}(\Theta_\alpha, \beta_r) \wedge \bigwedge \rho_j \wedge \text{prec}(\beta_r)), T_p, T_{\beta_r}, P_\alpha)$$

$$\text{Int.Th}(G_j, (\exists R_{\beta_r})\text{CBAG}(\text{GR}_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \text{inherit}(\Theta_\alpha, \beta_r) \wedge \bigwedge \rho_j \wedge \text{prec}(\beta_r)), T_p, T_{\beta_r}, P_\alpha)$$

MB **satisfaction (MB-IT):** (Clauses (2a2) and (2b3))

$$\text{Int.Th}(G_i, \text{MB}(\text{GR}, (\exists R_{\beta_r})\text{CBA}(G_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \text{inherit}(\Theta_\alpha, \beta_r) \wedge \bigwedge \rho_j \wedge \text{prec}(\beta_r)), T_{\beta_r}), T_p, T_{\beta_r}, P_\alpha)$$

$$\text{Int.Th}(G_i, \text{MB}(\text{GR}, (\exists R_{\beta_r})\text{CBAG}(\text{GR}_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \text{inherit}(\Theta_\alpha, \beta_r) \wedge \bigwedge \rho_j \wedge \text{prec}(\beta_r)), T_{\beta_r}), T_p, T_{\beta_r}, P_\alpha)$$

Figure 6. Intentions-that in SharedPlans

Axiom 1 :
$(\forall G, T_i)$ Int.Th$(G, prop, T_i, T_{prop}, \text{IC}_{prop}) \Rightarrow \neg$Bel$(G, \neg prop, T_i)$

Axiom 2 :
If an agent intends-that $prop$ and does not believe $prop$
$(\forall G, T_i)$
(1) [Int.Th$(G, prop, T_i, T_{prop}, \text{IC}_{prop})$ $\wedge$ $\neg$Bel$(G, prop, T_i)] \Rightarrow$ [
and there is some action the agent can take that would lead to $prop$ holding then
(2) $[(\exists\alpha, T_\alpha)$LEAD$(G, \alpha, T_\alpha, prop, \text{con}(\alpha), T_i)] \Rightarrow$ [
either there is some such action that the agent (potentially) intends to do
$(3a)$ $[(\exists\alpha, T_\alpha)$LEAD$(G, \alpha, T_\alpha, prop, \text{con}(\alpha), T_i) \wedge$
$(3b)$ [Int.To$(G, \alpha, T_\alpha, T_i, \text{con}(\alpha), prop) \vee$ Pot.Int.To$(G, \alpha, T_\alpha, T_i, \text{con}(\alpha), prop)$]] $\bigvee$
or for every such action there is some conflict with the agent's other intentions
(4) $[(\forall\alpha, T_\alpha)$LEAD$(G, \alpha, T_\alpha, prop, \text{con}(\alpha), T_i) \Rightarrow$
$[[(\exists\beta, T_\beta, \text{IC}_\beta)$Int.To$(G, \beta, T_i, T_\beta, \text{con}(\beta), \text{IC}_\beta) \wedge$
Bel$(G, \text{CONF}(\alpha, \beta, T_\alpha, T_\beta, \text{con}(\alpha), \text{con}(\beta)), T_i)] \vee$
$[(\exists prop, T_{prop}, \text{IC}_{prop})$Int.Th$(G, prop, T_i, T_{prop}, \text{IC}_{prop}) \wedge$
Bel$(G, \text{CONF}(\beta, prop, T_\beta, T_{prop}, \text{con}(\beta)), T_i)]]]]]$

LEAD$(G, \alpha, T_\alpha, prop, \Theta, T_i)$
(1) Bel$(G, (\exists R_\alpha, T_\alpha)$ CBA$(G, \alpha, R_\alpha, T_\alpha, \Theta), T_i) \bigwedge$
(2) [[Bel$(G, \text{Do}(G, \alpha, T_\alpha, \Theta) \Rightarrow prop, T_i)] \bigvee$
(3) [Bel$(G_1, \text{Do}(G_1, \alpha, T_\alpha, \Theta) \Rightarrow$
$[[(\exists\beta, G_2, R_\beta, T_\beta, T_i')$
$[\,T_\alpha \leq T_\beta \wedge$ single.agent(β) $\wedge$ CBA$(G_2, \beta, R_\beta, T_\beta, \text{con}(\beta))$ $\wedge$
Pot.Int.To$(G_2, \beta, T_\beta, T_i', \text{con}(\beta), \text{IC}_\beta) \wedge T_i' < T_\beta \wedge$
$[\text{Do}(G_2, \beta, T_\beta, \text{con}(\beta)) \Rightarrow prop]]]$ $\vee$
$[(\exists\beta, \text{GR}_2, R_\beta, T_\beta, T_i')$
$[T_\alpha \leq T_\beta$ $\wedge$ multi.agent$(\beta) \wedge$ CBAG$(\text{GR}_2, \beta, R_\beta, T_\beta, \Theta) \wedge$
$(\forall G_j \in \text{GR}_2)$Pot.Int.Th$(G_j, \text{Do}(\text{GR}_2, \beta, T_\beta, \text{con}(\beta)), T_i', T_\beta, \text{IC}_\beta) \wedge T_i' < T_\beta$ $\wedge$
$[\text{Do}(\text{GR}_2, \beta, T_\beta, \text{con}(\beta))) \Rightarrow prop]]], T_i)]]$

Figure 7. Axiom schema encoding minimal conditions for intending-that

CULTIVATE$(Int.Th(G, prop, T_i, T_{prop}, \mathrm{IC}_{prop}))$, where $T_i \leq T_{prop}$

RECONCILE(prop,intentions-to $\wedge$ intentions-that)
Loop:
  if $Bel(G, \neg prop, T_i)$ then fail
  else if $\neg$Bel$(G, prop, T_i)$ then
    if know action that may lead to Bel$(G, prop, T_{i+1})$, then act (Cultivate-Act)
    else set up monitoring triggers (Cultivate-Monitor).
  else if [Bel$(G, prop, T_i)$]
    set up monitoring triggers (Cultivate-Monitor);
    if prop relevant to action by another agent, then consider helping (Cultivate-Help).
  sleep;
end loop;

Figure 8. Pseudocode for CULTIVATE process.

1. The first agent that needs a parameter chooses an appropriate entity for the parameter and broadcasts this choice to other agents; this strategy requires some conflict resolution mechanism for the situation in which multiple agents choose simultaneously.
2. Agents decide at the time of forming the initial SharedPlan who will identify each parameter and how others will find out its identity.
3. Agents decide at the time of choosing the recipe who will identify each parameter and how others will find out its identity.

Figure 9. Parameter identification strategies

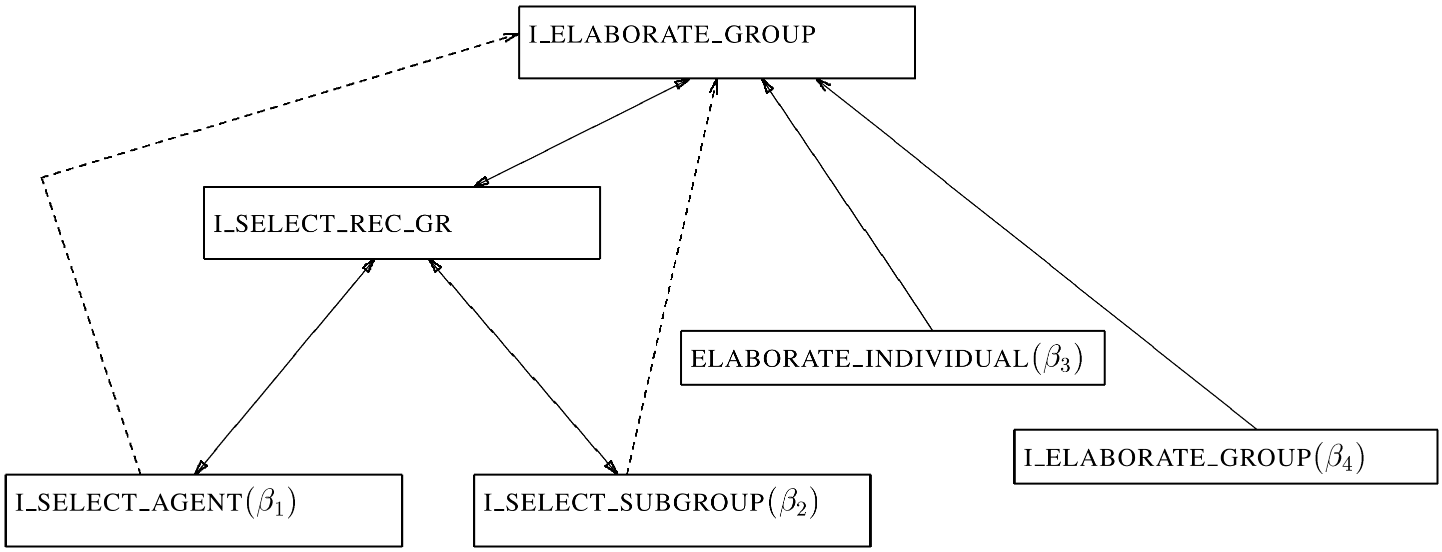


Figure 10. Example of the basic architecture of the SP subsystem in action. In this example, β_1 and β_2 are unresolved (β_k) and β_3 and β_4 are resolved subactions (β_r).

Type	Notation	Meaning	Section
Modal Operators	Int.To	intend-to	
	Int.Th	intend-that	
	Pot.Int.To	potential intention-to	
	Pot.Int.Th	potential intention-that	
	Bel	belief	
	MB	Mutual belief	
	Do	performance of action	
Meta-Predicates (Plans)	FIP	full individual plan	
	PIP	partial individual plan	
	SP	SharedPlan	
	FSP	full SharedPlan	
	PSP	partial SharedPlan	
Meta-Predicates (Ability)	CBA	can bring about	
	CBAG	can bring about group	
Meta-Predicates (subsidiary)	CONF	actions/propositions conflict	
	PSPC	contracting in PSP	
	CCG	group of agents can contract	
Act-types for Planning Actions	Select_Rec	agent selects (extends) recipe	
	Select_Rec_GR	group of agents selects (extends) a recipe	
	Select_Agent	group of agents selects a member for a subaction	2.3
	Select_Subgroup	group of agents selects a subgroup for a subaction	2.3
	Elaborate_Individual	agent extends partial plan	
	Cultivate	cultivate an intention-that	3.3
	Elaborate_Group	group of agents extends a partial SharedPlan	
	I_Select_Rec_GR	individual planning action constituting Select_Rec_GR	*
	I_Select_Agent	individual planning action constituting Select_Agent	*
	I_Select_Subgroup	individual planning action constituting Select_Subgroup	*
	I_Elaborate_Group	individual planning action constituting Elaborate_Group	2.1
	Reconcile	reconcile new intention with old ones	
Predicates (subsidiary)	single.agent	single-agent action	
	multi.agent	group action	
	id.params	parameters had been identified	3.1
	selected.rec	selected recipe	*
Functions	cost	cost of action	
	econ	relativize cost (for benefit comparison)	
	inherit	inherit constraints	2.3
	prec	preconditions	2.3
	con	constraints of action	2.3
Processes	CULTIVATE	process which realized Cultivate	4.1
	I_ELABORATE_GROUP	process which realized I_Elaborate_Group	4.2
	I_SELECT_AGENT	process which realized I_Select_Agent	4.2
	I_SELECT_SUBGROUP	process which realized I_Select_Subgroup	4.2
	I_SELECT_REC_GR	process which realized I_Select_Rec_GR	4.2
	RECONCILE	process which realized Reconcile	4.2

Table 3. Summary of notations. If there is no entry in the last column, the concept was described in an earlier paper [10]; a * is used for certain individual planning actions that differ slightly from that paper but are not discussed in this paper.

PSP(P_α, GR, $\alpha, T_p, T_\alpha, \Theta_\alpha, IC_\alpha$)
(0) MB(GR, $(\forall G_j \in$ GR) Int.Th(G_j, Do(GR, $\alpha, T_\alpha, \Theta_\alpha$), T_p, T_α, IC$_\alpha$), T_p) $\wedge$
($0a$) MB(GR($\forall G_j \in$ GR)Int.Th(G_j, id.params(GR, α, T_α), T_p, T_α, P_α), T_p) $\wedge$
($0b$) MB(GR($\forall G_j \in$ GR)Int.Th($G_j, \Theta_\alpha, T_p, T_{\Theta_\alpha}, P_\alpha$), T_p)
$(\exists R_\alpha$, B, B$_{res}$, B$_{unres}$, C)[
($1a$) $R_\alpha =$ (B, C) $\wedge$ B = B$_{res} \cup$ B$_{unres} \wedge$
($1b$) [MB(GR, $R_\alpha \in Recipes(\alpha) \wedge$ selected.rec(GR, R_α, P_α), T_p) $\otimes$
($1c$) [MB(GR, $(\exists R^f_\alpha$, B^f, C^f)$R^f_\alpha =$ (B^f, C^f) $\wedge R^f_\alpha \in Recipes(\alpha) \wedge$ B $\subset$ B$^f \wedge$ C $\subset$ C^f, T_p) $\wedge$
($1d$) $(\exists T_{select.rec.g}, P_{select.rec.g}, R_{select.rec.g})$
($1d1$) FSP($P_{select.rec.g}$, GR, Select_Rec_GR(GR, $\alpha, R_\alpha, T_{select.rec.g}$), $T_p, T_{select.rec.g}$, $R_{select.rec.g}, \Theta_{select.rec.g}, P_\alpha$)]] $\bigwedge$
$(\forall \beta_r \in$ B$_{res}$)[
($2a$) [[single.agent(β_r) $\wedge (\exists G_k \in$ GR)
($2a1a$) [Int.To($G_k, \beta_r, T_p, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r), P_α)$\wedge$
($2a1b$) MB(GR, Int.To($G_k, \beta_r, T_p, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r), P_α), T_p)$\wedge$
($2a2$) $(\forall G_i \in$ GR) Int.Th(G_i, MB(GR, $(\exists R_{\beta_r}$)
($2a2a$) CBA($G_k, \beta_r, R_{\beta_r}, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r)), T_{β_r}), $T_p, T_{\beta_r}, P_\alpha$)$\wedge$
($2a3$) MB(GR, $(\forall G_j \in$ GR, $G_j \neq G_k$)Int.Th(G_j, $(\exists R_{\beta_r}$)
($2a3a$) CBA($G_k, \beta_r, R_{\beta_r}, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r)), $T_p, T_{\beta_r}, P_\alpha$), T_p)]] $\bigotimes$
($2b$) [multi.agent(β_r) $\wedge (\exists$GR$_k \subset GR$)
($2b1$) [$(\exists P_{\beta_r}$)SP(P_{β_r}, GR$_k, \beta_r, T_p, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r), P_α)$\wedge$
($2b2$) MB(GR, $(\exists P_{\beta_r}$)SP(P_{β_r}, GR$_k, \beta_r, T_p, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r), P_α), T_p)$\wedge$
($2b3$) $(\forall G_i \in$ GR)Int.Th(G_i, MB(GR, $(\exists R_{\beta_r}$)
($2b3a$) CBAG(GR$_k, \beta_r, R_{\beta_r}, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r)), T_{β_r}), $T_p, T_{\beta_r}, P_\alpha$)$\wedge$
($2b4$) MB(GR, $(\forall G_j \in$ GR $\setminus$ GR$_k$)Int.Th(G_j, $(\exists R_{\beta_r}$)
($2b4a$) CBAG(GR$_k, \beta_r, R_{\beta_r}, T_{\beta_r}$, inherit($\Theta_\alpha, \beta_r$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r)), $T_p, T_{\beta_r}, P_\alpha$), T_p)]]] $\bigotimes$
(3) [PSPC(GR, β_r, T_{β_r}, inherit(Θ_α, β_r) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_r))]] $\bigwedge$

(4) $(\forall \beta_k \in$ B$_{unres}$)[
($4a1$) [single.agent(β_k) $\wedge$
($4a1a$) MB(GR, $(\exists G_k \in$ GR, R_{β_k}, T_{β_k})CBA($G_k, \beta_k, R_{\beta_k}, T_{\beta_k}$, inherit($\Theta_\alpha, \beta_k$) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_k))$\vee$
($4a1a1$) $(\exists T_{\beta_k}$)CCG(GR, β_k, T_{β_k}, inherit(Θ_α, β_k) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_k)), T_p)$\wedge$
($4a1b$) $(\exists T_{select.agent}, P_{select.agent}, R_{select.agent})$ FSP($P_{select.agent}$, GR
($4a1b1$) Select_Agent(GR, $\beta_k, T_{select.agent}$), $T_p, T_{select.agent}, R_{select.agent}$,
($4a1b2$) inherit(Θ_α, β_k) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_k), P_α)]$\otimes$
($4a2$) [multi.agent(β_k)$\wedge$
($4a2a$) MB(GR, $(\exists$GR$_k \subseteq$ GR, R_{β_k}, T_{β_k})$CBAG$(GR$_k, \beta_k, R_{\beta_k}, T_{\beta_k}$,
($4a2a1$) inherit(Θ_α, β_k) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_k))$\vee$
($4a2a1$) $(\exists T_{\beta_k}$)CCG(GR, β_k, T_{β_k}, inherit(Θ_α, β_k) $\wedge \bigwedge_{\rho_j \in \mathrm{C}} \rho_j \wedge$ prec(β_k)), T_p)$\wedge$
($4a2b$) $(\exists T_{select.subgroup}, P_{select.subgroup}, R_{select.subgroup})$ FSP($P_{select.subgroup}$, GR,
($4a2b1$) Select_Subgroup(GR, $\beta_k, T_{select.subgroup}$), $T_p, T_{select.subgroup}, R_{select.subgroup}, \Theta_{ssg}, P_\alpha$)]]]

$\{COMMENT:$
$\Theta_{select.rec.g} \equiv$
$(\exists R^f_\alpha$, B^f, C^f)[
$R^f =$ (B^f, C^f) $\wedge$ MB(GR, $R^f_\alpha \in Recipes(\alpha) \wedge$ B $\subseteq$ B$^f \wedge$ C $\subseteq C^f, T_{select.rec.g}$)$\wedge$
$(\forall \delta_v \in$ B^f)MB(GR, $(\exists T_{\delta_v}$)
[[single.agent(δ_v)$\wedge$
$(\exists G_{\delta_v} \in$ GR, R_{δ_v})CBA($G_{\delta_v}, \delta_v, T_{\delta_v}, R_{\delta_v}$, inherit($\Theta_\alpha, \delta_v$) $\wedge \bigwedge_{\kappa_e \in \mathrm{C}} \kappa_e \wedge$ prec(δ_v))]$\otimes$
[multi.agent(δ_v)$\wedge$
$(\exists$GR$_{\delta_v} \subseteq$ GR, R_{δ_v})CBAG(GR$_{\delta_v}, \delta_v, T_{\delta_v}, R_{\delta_v}$, inherit($\Theta_\alpha, \delta_v$) $\wedge \bigwedge_{\kappa_e \in \mathrm{C}} \kappa_e \wedge$ prec(δ_v))$\vee$
CCG(GR, δ_v, T_{δ_v}, inherit(Θ_α, δ_v) $\wedge \bigwedge_{\kappa_e \in \mathrm{C}} \kappa_e \wedge$ prec(δ_v))]], $T_{select.rec.g}$)] $\bigwedge$
$\Phi_{select.rec.g}$
}

Figure 11. Formal Definition of PSP