Making SharedPlans More Concise and Easier to Reason About*

Luke Hunsberger
luke@eecs.harvard.edu

Harvard University
Cambridge, MA USA

Abstract. SharedPlans is a general theory of collaborative planning that accommodates multi-level action decomposition hierarchies and explicates the process of expanding partial plans into full plans [5, 6]. This paper presents a reformulation of SharedPlans that simplifies the SharedPlans definitions without sacrificing their expressiveness, and enables the specification of conditions under which a set of important theorems about agents and their SharedPlans may be proven to hold. A representative set of such theorems is presented.

1 Introduction

> "Collaboration must be designed into systems from the start; it cannot be patched on." [4]

> "Simply fitting individual agents with precomputed coordination plans will not do, for their inflexibility can cause severe failures in teamwork." [12]

When a group of agents get together to work on some complex group action, whether it be a group of helicopter agents embarking on a scouting mission [12] or a group of people making dinner [5], collaboration does not just happen. It requires the existence or formation of mutual beliefs about the capabilities and commitments of the agents involved, the adoption by individual agents of various intentions (not only *intentions-to* do various actions, but also *intentions-that* certain propositions hold), and a variety of group decision-making and planning processes. Grosz and Kraus's SharedPlans [5, 6] is a general theory of collaborative planning that requires no notion of irreducible joint intentions, accommodates multi-level action decomposition hierarchies, models the collaborative support provided by group members to those agents or subgroups responsible for doing constituent actions, specifies what it means for a group of agents to have a *partial* plan, and explicates the process of elaborating a partial plan into a full plan. Recent implementations of SharedPlans include a collaborative interface agent for an air travel application [10] and a collaborative multi-agent

* The author thanks Barbara J. Grosz for her invaluable guidance during the preparation of this paper. This research has been supported by National Science Foundation grants IRI 95-25915, IRI-9618848 and CDA-94-01024.

system for electronic commerce [7]. In addition, to test, evaluate and improve the theory, the author is currently developing an agent architecture that follows the SharedPlans specifications.

In SharedPlans, the plans of individual agents and groups of agents are modelled by *meta-predicates*—that is, abbreviations for complex logical expressions involving predicates and the following modal operators.

Operator	Instantiation	Interpretation
Bel	$Bel(G, \phi)$	Agent G believes proposition ϕ.
$Int.To$	$Int.To(G, A)$	Agent G intends to do action A.
$Int.Th$	$Int.Th(G, \phi)$	Agent G intends that proposition ϕ hold.
MB	$MB(GR, \phi)$	Group GR mutually believe proposition ϕ.

In the original formulation, henceforth called V_1 (for *Version 1*), the meta-predicate definitions use existential quantification to refer to various agents, subgroups and actions involved in a plan, thereby making it difficult to reason about such things as the conditions under which a group's mutual belief that they have a SharedPlan entails that they do in fact have such a plan.[2] This paper presents a reformulation of SharedPlans, henceforth called V_2, that simplifies and reorganizes the meta-predicate definitions without sacrificing their expressiveness, and enables the specification of knowledge conditions under which a set of important theorems about agents and their SharedPlans may be proven.[3] This paper thus represents a step in the direction of making SharedPlans more practical to implement and reason about, not only for the agents themselves, but also for theorists studying the agents.

V_1 is reviewed in Section 2; V_2 is presented in Section 3; sample theorems and their proofs are given in Section 4; related work is discussed in Section 5; and concluding remarks are given in Section 6.

2 The Original Formulation of SharedPlans: V_1

2.1 Actions and Recipes in V_1

In V_1, actions are either *basic* or *complex*. A basic action is a single-agent action that is treated as atomic and, under certain conditions, is assumed to be executable at will. A complex action may be a single-agent or multi-agent action and is treated as decomposable. A *recipe* for a complex action, A, is a set of actions, $\{A_1, \ldots, A_n\}$, and constraints, $\{\rho_1, \ldots, \rho_m\}$, such that the doing of those actions under those constraints constitutes the doing of A. A *partial* recipe is a set of actions and constraints that can be expanded into a complete recipe. Multi-agent complex actions are assumed to be ultimately decomposable into single-agent

[2] A typical difficulty stems from the fact that $Bel(G, (\exists x)P(x))$ does not, in general, entail $(\exists x)Bel(G, P(x))$.

[3] To simplify the presentation, the case of "contracting out" actions to other agents is ignored, as are parameters not central to the discussion, such as constraints, time, and intentional context.

actions (basic or complex); single-agent complex actions are assumed to be ultimately decomposable into basic actions. Recursive decomposition gives rise to an *action decomposition hierarchy.* An action decomposition hierarchy is called complete if (1) the decomposition of each action in the hierarchy corresponds to a complete recipe, and (2) all leaf actions are basic actions.

2.2 Intention-To and Individual Plans in V_1

The modal operator $Int.To$ models the intention of an agent G to do a single-agent action A. If A is basic, then G intending to do A requires that G believe it is able to execute A and that G be committed to doing so:

$$Basic(A) \wedge Int.To(G, A) \Rightarrow Bel(G, Exec(G, A)) \wedge Commit(G, A).$$

If A is complex, then G intending to do A requires either that G have a *Full Individual Plan* (modelled by the FIP meta-predicate) for doing A or that G have a *Partial Individual Plan* (PIP) for doing A accompanied by an *associate* plan for elaborating its partial plan into a full plan:[4]

$$Complex(A) \wedge Int.To(G, A) \Rightarrow FIP \otimes (PIP \wedge FIP^{Elab}).^5$$

An agent G has a Full Individual Plan for doing A if: (1) G has a complete recipe for doing A, (2) G intends to do each action in that recipe, and (3) G has a subordinate FIP to do each complex action in that recipe. The requirements for a Partial Individual Plan are much weaker. G's recipe for doing A may be partial or even empty—as long as G has an associate plan for extending the partial recipe into a complete recipe. In addition, G need not yet have formed intentions to do the actions comprising its partial recipe; G need only believe that it is *able* to do those actions. The ability of an agent to do a single-agent action is modelled by the *Can-Bring-About* (CBA) meta-predicate which, like $Int.To$, is defined in two parts to handle both basic and complex actions.

In the context of an individual plan, an action in the decomposition hierarchy is called *resolved* if the agent intends to do that action. Furthermore, a complex action resolved by a FIP is called *fully resolved.* Using this terminology, a full plan is characterized by a complete action decomposition hierarchy, each action of which has been resolved, the complex actions fully resolved. On the other hand, a partial plan is characterized by a possibly incomplete action decomposition hierarchy, some or all actions of which may be unresolved.

2.3 SharedPlans in V_1

V_1 provides analogous definitions for the plans of groups of *two* or more agents. A group of agents GR have a SharedPlan (SP) to do some multi-agent action, A, either by having a Full SharedPlan (FSP) to do A or by having a Partial SharedPlan (PSP) to do A accompanied by an associate plan to elaborate the partial plan into a full plan: $SP \Rightarrow FSP \otimes (PSP \wedge FSP^{Elab})$.

[4] Such associate plans are required to be FIPs to avoid problems of infinite recursion.
[5] The arguments of FIP and PIP have been omitted to simplify the presentation.

Unlike *Int.To*, the *SP* meta-predicate is not a modal operator. A SharedPlan is reducible to the individual plans, beliefs and intentions of the various group members; it does not correspond to any sort of irreducible joint intention.

In the context of a SharedPlan, a single-agent action A_i in the decomposition hierarchy is called resolved if: (1) an agent G_i has been selected to do A_i, (2) G_i intends to do A_i, and (3) the other members of the group have a set of supportive mutual beliefs and intentions-that G_i succeed.[6] Similarly, a multi-agent action A_j is called resolved if: (1) a subgroup GR_j has been selected to do A_j, (2) GR_j has a SharedPlan to do A_j, and (3) the other members of the group have a set of supportive mutual beliefs and intentions-that GR_j succeed. As with Individual Plans, a complex action resolved by a full plan (whether a FIP or a FSP) is called fully resolved. Thus, a Full SharedPlan is characterized by a complete action decomposition hierarchy, each action of which has been fully resolved, while a Partial SharedPlan is characterized by a possibly incomplete action decomposition hierarchy, some or all actions of which may be unresolved. (Incidentally, if a complex action in a partial plan is itself resolved by a mere partial plan, the recipe associated with that action may be only partial or even empty.) To elaborate a partial plan into a full plan, for each complex action in the decomposition hierarchy, the agent or group selected to work on that action must select (perhaps incrementally) a recipe for doing that action and, for each action in that recipe, must select an agent or subgroup that is able to do it. The ability of a group to do a multi-agent action is modelled by the *Can-Bring-About-Group* ($CBAG$) meta-predicate.

The following chart illustrates the coverage of the V_1 meta-predicates and the *Int.To* modal operator.

<table>
<tr><td rowspan="3">V_1</td><td>Basic Actions</td><td rowspan="2">$Int.To$</td><td>---</td><td>---</td><td rowspan="2">CBA</td></tr>
<tr><td>Single-Agent Complex Actions</td><td>FIP</td><td>PIP</td></tr>
<tr><td>Multi-Agent Actions ($\geq$ 2 agents)</td><td>SP</td><td>FSP</td><td>PSP</td><td>$CBAG$</td></tr>
</table>

3 The Reformulation of SharedPlans: V_2

3.1 Actions and Plans in V_2

In V_2, for complex actions, the distinction between single-agent and multi-agent actions is deemphasized. Instead, single-agent groups are allowed and an Individual Plan is simply a SharedPlan of a single-agent group. In addition, the V_2 definitions of SP, FSP, PSP and $CBAG$ are made more concise than their V_1 counterparts through the selective use of a new meta-predicate, *Basic-Can-Bring-About* ($B.CBA$), and a new modal operator, *Basic-Intention-To* ($B.Int.To$), defined by those portions of the V_1 definitions of CBA and $Int.To$ that deal with basic actions. The following chart illustrates the coverage of the V_2 meta-predicates and the $B.Int.To$ modal operator.

<table>
<tr><td rowspan="2">V_2</td><td>Basic Actions</td><td>$B.Int.To$</td><td>---</td><td>---</td><td>$B.CBA$</td></tr>
<tr><td>Complex Actions ($\geq$ 1 agents)</td><td>SP</td><td>FSP</td><td>PSP</td><td>$CBAG$</td></tr>
</table>

[6] The properties of *intentions-that* are discussed in detail by Grosz & Kraus [6].

3.2 Plan Trees in V_2

In the process of constructing a SharedPlan, various agents and subgroups may make numerous planning decisions (e.g., selecting recipes and assigning agents to actions) in a distributed fashion and at every level of the evolving action decomposition hierarchy. In V_1, even decisions that have already been made, such as those concerning the elements of a full plan, are modelled implicitly using existential quantification. In V_2, SharedPlan Trees (SPTs) are used to explicitly represent the choices already made by a group working on some SharedPlan. Each node of an SPT corresponds to an action in the incrementally-selected and possibly incomplete action decomposition hierarchy and is explicitly classified according to whether that action is basic or complex, and resolved or unresolved (*vis à vis* the plan). Thus, there are four types of nodes, as summarized below:

Node Type	Node Representation	Action Characteristics	
β	$\langle I_\beta, G_\beta, A_\beta \rangle$	basic	resolved
κ	$\langle I_\kappa, GR_\kappa, A_\kappa \rangle$	complex	resolved
ϵ	$\langle I_\epsilon, A_\epsilon \rangle$	basic	unresolved
μ	$\langle I_\mu, A_\mu \rangle$	complex	unresolved

where $I_\beta, I_\kappa, I_\epsilon$ and I_μ are unique identifiers; $A_\beta, A_\kappa, A_\epsilon$ and A_μ are actions; G_β is an agent (nominally the agent selected to do the action, A_β); and GR_κ is a group of agents (nominally the subgroup selected to do the action, A_κ). All identifiers, agent names and action names are assumed to be rigid designators.

In a SharedPlan Tree, only κ nodes may have child nodes—but these child nodes may be of any of the four types. The set of β child nodes of a given node are termed its βset. Similarly, the sets of κ, ϵ and μ child nodes of a given node are termed its κset, its ϵset, and its μset, respectively.

Because all actions involved in a full plan are, by definition, resolved, a Full Plan Tree (FPT) has only β and κ nodes. Partial plans, however, may have unresolved actions and hence a Partial Plan Tree (PPT) may have nodes of any of the four types. In addition, partial plans typically have a variety of associate plans corresponding to complex planning actions, such as selecting a recipe ($SelRec$), elaborating a partial plan into a full plan ($Elab$), or selecting an agent or subgroup to do some action ($SelAgt$ or $SelSgr$). Thus, each node in a Partial Plan Tree may have one or more additional plan trees associated with it as summarized below.

Node Type	Type of Associate Plan	Functional Notation for Corresponding Plan Tree	Abbreviation for Plan Tree
κ	Elaborate	$ElabPT(\langle I_\kappa, GR_\kappa, A_\kappa \rangle)$	PT_κ^{Elab}
	Select Recipe	$SelRecPT(\langle I_\kappa, GR_\kappa, A_\kappa \rangle)$	PT_κ^{SelRec}
ϵ	Select Agent	$SelAgtPT(\langle I_\epsilon, A_\epsilon \rangle)$	PT_ϵ^{SelAgt}
μ	Select Subgroup	$SelSgrPT(\langle I_\mu, A_\mu \rangle)$	PT_μ^{SelSgr}

Definition. Given some κ node, $N = \langle I, GR, A \rangle$, the **SharedPlan Tree** (or subtree) rooted at N is a 7-tuple:

$$\langle N, \beta set, \kappa set, \epsilon set, \mu set, ElabPT, SelRecPT \rangle,$$

where βset, κset, ϵset and μset are sets of β, κ, ϵ and μ nodes, respectively, such that for each $\langle I_\kappa, GR_\kappa, A_\kappa \rangle$ in κset, the object given by $PlanTree(\langle I_\kappa, GR_\kappa, A_\kappa \rangle)$, abbreviated as PT_κ, is itself a SharedPlan Tree. In a **Full Plan Tree**, $ElabPT$ and $SelRecPT$ are `NIL`, ϵset and μset are empty, and each PT_κ is itself a Full Plan Tree. In a **Partial Plan Tree**, $ElabPT$ is a Full Plan Tree (and hence not `NIL`) and $SelRecPT$ is either `NIL` or a Full Plan Tree.

3.3 V_2 Definitions

V_2 definitions of $B.CBA$, $B.Int.To$, $CBAG$, SP, FSP and PSP are given in Figs. 1 and 2. The definitions of $B.CBA$ and $B.Int.To$ are simply those portions of the V_1 definitions of CBA and $Int.To$ that deal with basic actions. The V_2 definitions of $CBAG$, SP, FSP and PSP are generalizations of their V_1 counterparts in that they allow for single-agent groups. Thus, in V_2 there is no need for a separate set of meta-predicates to handle single-agent plans. The V_2 definitions of $CBAG$, FSP, PSP and SP also differ from their V_1 counterparts in that each takes an explicit plan tree,

$$PT_\alpha = \langle \langle I_\alpha, GR_\alpha, A_\alpha \rangle, \beta set_\alpha, \kappa set_\alpha, \epsilon set_\alpha, \mu set_\alpha, PT_\alpha^{Elab}, PT_\alpha^{SelRec} \rangle,$$

as its only argument. For brevity, the symbol PT_α is used instead of the 7-tuple; but it should be kept in mind that the 7-tuple is the actual argument. For example, $FSP(PT_\alpha)$ represents that the group GR_α has a Full SharedPlan to do the action A_α using the plan tree PT_α.[7]

The V_2 definitions of FSP and PSP are given in terms of subsidiary meta-predicates to distinguish the top-level and recursive portions of the definitions. Making this distinction reflects a fundamental tenet of SharedPlans, namely, that while the entire group needs to be directly involved in the topmost level of a plan, only the agents selected to do a given subaction need to be directly involved in the corresponding subplan. Making this distinction also enables precise specification of the knowledge conditions and mutual beliefs needed for the theorems presented in Section 4.

$FSP.Top$ and $PSP.Top$ model the top-level (or non-recursive portion) of a SharedPlan. As such, their specifications are restricted to the top level of the plan tree (i.e., the root node and its immediate children). For example, they specify various intentions and mutual beliefs pertaining to the immediate children of the root node; but they do not specify, directly or indirectly, anything pertaining to nodes further down in the tree. $FSP.Rec$ and $PSP.Rec$, on the other hand, encapsulate the recursive portions of the FSP and PSP definitions. As such,

[7] In cases where it is desirable to explicitly indicate the group and possibly the action involved, they are included as parameters of the plan tree symbol, as in the fragments, $FSP(PT_\kappa(GR_\kappa))$ and $(\exists PT)FSP(PT(GR_\kappa, A_\kappa))$, from the FSP definition.

(Basic) Can-Bring-About

$B.CBA(G, A) \equiv$

$$Basic(A) \\ \wedge \\ Exec(G, A)$$

(Basic) Intention-To

$B.Int.To(G, A) \equiv$

$$Bel(G, B.CBA(G, A)) \\ \wedge \\ Commit(G, A)$$

SharedPlan

$SP(PT_\alpha) \equiv$

$$FSP(PT_\alpha) \otimes PSP(PT_\alpha)$$

Can-Bring-About (Group)

$CBAG(PT_\alpha) \equiv$

$$\begin{array}{l} PT_\alpha^{Elab} = PT_\alpha^{SelRec} = \mathtt{NIL} \\ \wedge \\ \epsilon set_\alpha = \mu set_\alpha = \emptyset \\ \wedge \\ Top(PT_\alpha) \in Recipes(A_\alpha) \\ \wedge \\ (\forall \langle I_\beta, G_\beta, A_\beta \rangle \in \beta set_\alpha) \\ \quad G_\beta \in GR_\alpha \\ \quad \wedge \\ \quad B.CBA(G_\beta, A_\beta) \\ \wedge \\ (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha) \\ \quad GR_\kappa \subseteq GR_\alpha \\ \quad \wedge \\ \quad CBAG(PT_\kappa(GR_\kappa)) \end{array}$$

Full SharedPlan

$FSP(PT_\alpha) \equiv FSP.Top(PT_\alpha) \wedge FSP.Rec(PT_\alpha)$,
where

Full SharedPlan: Top-Level Portion

$FSP.Top(PT_\alpha) \equiv F_1 \wedge F_2 \wedge F_3 \wedge F_4 \wedge F_5 \wedge F_6 \wedge F_\beta \wedge F_\kappa$
where

$$\begin{array}{l} F_1 \equiv (PT_\alpha^{Elab} = \mathtt{NIL}) \\ F_2 \equiv (PT_\alpha^{SelRec} = \mathtt{NIL}) \\ F_3 \equiv (\epsilon set_\alpha = \emptyset) \\ F_4 \equiv (\mu set_\alpha = \emptyset) \\ F_5 \equiv MB_\alpha((\forall G \in GR_\alpha) Int.Th(G, Do(GR_\alpha, A_\alpha))) \\ F_6 \equiv MB_\alpha(Top(PT_\alpha) \in Recipes(A_\alpha)) \\ F_\beta \equiv (\forall \langle I_\beta, G_\beta, A_\beta \rangle \in \beta set_\alpha) F_{\beta_0} \wedge F_{\beta_1} \wedge F_{\beta_2} \wedge F_{\beta_3} \\ F_\kappa \equiv (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha) F_{\kappa_0} \wedge F_{\kappa_2} \wedge F_{\kappa_3} \end{array}$$

where

$$\begin{array}{l} F_{\beta_0} \equiv G_\beta \in GR_\alpha \\ F_{\beta_1} \equiv B.Int.To(G_\beta, A_\beta) \\ F_{\beta_2} \equiv MB_\alpha(B.Int.To(G_\beta, A_\beta) \wedge B.CBA(G_\beta, A_\beta)) \\ F_{\beta_3} \equiv MB_\alpha((\forall G \in GR_\alpha, G \neq G_\beta) Int.Th(G, B.CBA(G_\beta, A_\beta))) \end{array}$$

and

$$\begin{array}{l} F_{\kappa_0} \equiv GR_\kappa \subseteq GR_\alpha \\ F_{\kappa_2} \equiv MB_\alpha((\exists PT) FSP(PT(GR_\kappa, A_\kappa)) \wedge CBAG(PT(GR_\kappa, A_\kappa))) \\ F_{\kappa_3} \equiv MB_\alpha((\forall G \in GR_\alpha, G \notin GR_\kappa) \\ \qquad\qquad Int.Th(G, (\exists PT) CBAG(PT(GR_\kappa, A_\kappa)))) \end{array}$$

Full SharedPlan: Recursive Portion

$FSP.Rec(PT_\alpha) \equiv (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha) FSP(PT_\kappa(GR_\kappa))$

Fig. 1. V_2 definitions of $B.CBA, B.Int.To, CBAG, SP$ and FSP

Partial SharedPlan
$PSP(PT_\alpha) \equiv PSP.Top(PT_\alpha) \wedge PSP.Rec(PT_\alpha)$,
where

Partial SharedPlan: Top-Level Portion
$PSP.Top(PT_\alpha) \equiv P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5 \wedge P_\beta \wedge P_\kappa \wedge P_\epsilon \wedge P_\mu$
where

$$
\begin{aligned}
P_1 &\equiv (PT_\alpha^{Elab} \neq \texttt{NIL}) \\
P_2 &\equiv FSP(GR_\alpha, Elab(GR_\alpha, A_\alpha, Top(PT_\alpha)), PT_\alpha^{Elab}) \\
P_3 &\equiv (PT_\alpha^{SelRec} = \texttt{NIL}) \Rightarrow MB_\alpha(Top(PT_\alpha) \in Recipes(A_\alpha)) \\
P_4 &\equiv (PT_\alpha^{SelRec} \neq \texttt{NIL}) \Rightarrow \begin{cases} MB_\alpha((\exists PT)((Top(PT_\alpha) \subset Top(PT)) \\ \qquad \wedge\, CBAG(PT(GR_\alpha, A_\alpha)))) \\ \wedge \\ FSP(GR_\alpha, SelRec(GR_\alpha, A_\alpha, Top(PT_\alpha)), \\ \qquad PT_\alpha^{SelRec}) \end{cases} \\
P_5 &\equiv MB_\alpha((\forall G \in GR_\alpha) Int.Th(G, Do(GR_\alpha, A_\alpha))) \\
P_\beta &\equiv (\forall \langle I_\beta, G_\beta, A_\beta \rangle \in \beta set_\alpha) F_{\beta_0} \wedge F_{\beta_1} \wedge P_{\beta_{2.1}} \wedge P_{\beta_{2.2}} \wedge F_{\beta_3} \\
P_\kappa &\equiv (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha) F_{\kappa_0} \wedge P_{\kappa_{2.1}} \wedge P_{\kappa_{2.2}} \wedge F_{\kappa_3} \\
P_\epsilon &\equiv (\forall \langle I_\epsilon, A_\epsilon \rangle \in \epsilon set_\alpha) P_{\epsilon_1} \wedge P_{\epsilon_2} \\
P_\mu &\equiv (\forall \langle I_\mu, A_\mu \rangle \in \mu set_\alpha) P_{\mu_1} \wedge P_{\mu_2}
\end{aligned}
$$

where

$F_{\beta_0}, F_{\beta_1}, F_{\beta_3}, F_{\kappa_0}$ and F_{κ_3} are as in the FSP definition
and

$$
\begin{aligned}
P_{\beta_{2.1}} &\equiv MB_\alpha(B.Int.To(G_\beta, A_\beta)) \\
P_{\beta_{2.2}} &\equiv (\forall G \in GR_\alpha) Int.Th(G, MB_\alpha(B.CBA(G_\beta, A_\beta))) \\
P_{\kappa_{2.1}} &\equiv MB_\alpha((\exists PT) SP(PT(GR_\kappa, A_\kappa))) \\
P_{\kappa_{2.2}} &\equiv (\forall G \in GR_\alpha) Int.Th(G, MB_\alpha((\exists PT) CBAG(PT(GR_\kappa, A_\kappa)))) \\
P_{\epsilon_1} &\equiv MB_\alpha((\exists G \in GR_\alpha) B.CBA(G, A_\epsilon)) \\
P_{\epsilon_2} &\equiv FSP(GR_\alpha, SelAgt(GR_\alpha, A_\epsilon), PT_\epsilon^{SelAgt}) \\
P_{\mu_1} &\equiv MB_\alpha((\exists GR \subseteq GR_\alpha, PT) CBAG(PT(GR, A_\mu))) \\
P_{\mu_2} &\equiv FSP(GR_\alpha, SelSgr(GR_\alpha, A_\mu), PT_\mu^{SelSgr})
\end{aligned}
$$

Partial SharedPlan: Recursive Portion
$PSP.Rec(PT_\alpha) \equiv (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha) SP(PT_\kappa(GR_\kappa))$

Fig. 2. V_2 definition of PSP

their specifications refer to the plan subtrees rooted at the κ children of the root node. For example, $FSP.Rec$ requires that the subgroup selected to do the action corresponding to a κ child of the root node have a Full SharedPlan using the plan subtree rooted at that node.

In the V_2 definitions, clauses of the form $MB(GR_\alpha, \phi)$ appear so frequently that they are abbreviated as $MB_\alpha(\phi)$. In addition, the $CBAG$, FSP and PSP definitions refer to $Top(PT_\alpha)$ which denotes the set of actions in the top-level decomposition of PT_α. For example, $Top(PT_\alpha) \in Recipes(A_\alpha)$ represents that the top-level decomposition of A_α in the plan tree PT_α is a (complete) recipe for doing A_α. Finally, the V_1 requirement that a PSP be accompanied by an *Elab* FSP has been folded into the V_2 definition of PSP in clause P_2.

4 Theorems about Agents and Their SharedPlans

Under what knowledge conditions does an agent's belief that it has, say, a Full Individual Plan (FIP) entail that it does in fact have such a plan? In other words, what conditions would ensure that $Bel(G, FIP(G, \alpha, R_\alpha))$ entails $FIP(G, \alpha, R_\alpha)$, for some agent G, some action α, and some recipe R_α? The existential quantification in the V_1 meta-predicate definitions makes questions such as these difficult to answer. For example, in the above case, knowledge conditions might be sought such that the following holds:

$$Bel(G, (\exists R_\delta) FIP(G, \delta, R_\delta)) \models (\exists R_\delta) Bel(G, FIP(G, \delta, R_\delta)),$$

where δ is an action in the recipe R_α. But existential quantifiers may not, in general, be extracted from the scope of modal belief operators.

In V_2, the use of explicit plan trees as arguments in the various meta-predicate definitions eliminates such problems and allows a number of important theorems about agents and their SharedPlans to be formulated and proven. The theorems specify sets of knowledge conditions and sets of mutual beliefs such that under those knowledge conditions the agents have a SharedPlan if (or only if) they hold the specified mutual beliefs. For each action, A, in the decomposition hierarchy, the knowledge conditions stipulate that only those agents selected to work on A need know the top-level contents of the plan subtree associated with A. Similarly, only those agents selected to work on A need participate in the mutual beliefs about whether or not they satisfy the top-level requirements of a SharedPlan. For the theorems pertaining to full plans, the sets of knowledge conditions and mutual beliefs are completely specified and detailed proofs are given. For the theorems pertaining to partial plans and SharedPlans in general, space limitations preclude such a full treatment. Thus, these theorems are simply stated along with brief sketches of the issues involved in their proofs.

Before presenting the theorems, some background assumptions about the belief and mutual belief modal operators are given that lead to preliminary results used throughout the rest of this section. In addition, some assumptions about actions, commitments and intentions-that are made. In all that follows, all free variables are implicitly universally quantified and plan trees are assumed to have finite depth.

4.1 Background Assumptions and Preliminary Results

Bel, the modal belief operator, is assumed to satisfy the standard $KD45$ and necessitation axioms [3]. MB, the modal operator for mutual belief, is assumed to cover arbitrary nestings of Bel. Consequently, the following preliminary results are valid for arbitrary propositions ϕ and ψ.

(P1) $Bel(G, \phi \wedge \psi) \Leftrightarrow Bel(G, \phi) \wedge Bel(G, \psi)$
(P2) $Bel(G, Bel(G, \phi)) \Leftrightarrow Bel(G, \phi)$
(P3) $MB(GR, \phi \wedge \psi) \Leftrightarrow MB(GR, \phi) \wedge MB(GR, \psi)$
(P4) $MB(GR, MB(GR, \phi)) \Leftrightarrow MB(GR, \phi)$

Next, it is assumed that the universe of nodes is fixed. As a result, when the variable of quantification ranges over nodes, both the Barcan formula (B2) and its converse (B1), given below, are valid [3].

(B1) $Bel(G,(\forall x)P(x)) \Rightarrow (\forall x)Bel(G,P(x))$
(B2) $(\forall x)Bel(G,P(x)) \Rightarrow Bel(G,(\forall x)P(x))$

By providing appropriate knowledge conditions, these formulas may be extended to cover the case of the relativized universal quantifier: $(\forall x \in X)$.[8] For example, P5 below extends formula B1 using the knowledge condition K_1. K_1 requires that whenever x is in X, the agent G *believes* x is in X (i.e., G's beliefs about x being in X are *complete*). Similarly, P6 below extends formula B2 using the knowledge condition K_2. K_2 requires that G believe x is in X only when x actually is in X (i.e., G's beliefs about x being in X are *correct*).

(P5) $Bel(G,(\forall x \in X)P(x)) \wedge K_1 \models (\forall x \in X)Bel(G,P(x))$,
where $K_1 \equiv (\forall x \in X)Bel(G, x \in X)$.
(P6) $(\forall x \in X)Bel(G,P(x)) \wedge K_2 \models Bel(G,(\forall x \in X)P(x))$,
where $K_2 \equiv (\forall x)(Bel(G, x \in X) \Rightarrow (x \in X))$.

Furthermore, these results have mutual belief analogues, as follows.

(P7) $MB(GR,(\forall x \in X)P(x)) \wedge K_3 \models (\forall x \in X)MB(GR,P(x))$,
where $K_3 \equiv (\forall x \in X)MB(GR, x \in X)$.
(P8) $(\forall x \in X)MB(GR,P(x)) \wedge K_4 \models MB(GR,(\forall x \in X)P(x))$,
where $K_4 \equiv (\forall x)(MB(GR, x \in X) \Rightarrow (x \in X))$.

Finally, agents are assumed to have correct and complete beliefs about whether actions are basic or complex, and about their individual commitments to do actions and their individual intentions-that propositions hold.[9]

(A1) $Bel(G, Basic(A)) \Leftrightarrow Basic(A)$
(A2) $Bel(G, Complex(A)) \Leftrightarrow Complex(A)$
(A3) $Bel(G, Commit(G,A)) \Leftrightarrow Commit(G,A)$
(A4) $Bel(G, Int.Th(G,\phi)) \Leftrightarrow Int.Th(G,\phi)$

4.2 Theorems

Theorem 1 states that an agent G has an intention to do some basic action A if and only if G *believes* it has such an intention. Note that an analogous result does *not* hold for $B.CBA$, since an agent may be mistaken about its ability to do some basic action.

Theorem 1. $B.Int.To(G,A) \Leftrightarrow Bel(G, B.Int.To(G,A))$

[8] $(\forall x \in X)\psi(x)$ is an abbreviation for $(\forall x)((x \in X) \Rightarrow \psi(x))$.

[9] Assumptions A1 and A2 are made by Grosz & Kraus, assumption A3 ($\Rightarrow$) follows from Axiom 2 in V_1, and assumption A4 ($\Rightarrow$) is Axiom 3 in V_1 [5].

Proof of Theorem 1. Theorem 1 follows directly from the definition of $B.Int.To$, preliminary results P1 and P2, and assumption A3. If A3 is weakened to only a single direction of implication, then Theorem 1 must be similarly weakened. □

The rest of the theorems in this section specify the knowledge conditions sufficient to ensure that a group of agents hold a SharedPlan if (or only if) they hold a specified set of mutual beliefs. For example, Theorem 2 states that under the knowledge conditions given by $GrKnowFPT$, if a group of agents hold the set of mutual beliefs given by $RMB.FSP$, then they necessarily have a Full SharedPlan. $GrKnowFPT$ and $RMB.FSP$ are defined in Fig. 3.

Theorem 2. $RMB.FSP(PT_\alpha) \wedge GrKnowFPT(PT_\alpha) \models FSP(PT_\alpha)$

$GrKnowFPT(PT_\alpha)$ represents that the group of agents GR_α know the structure and contents of the plan tree PT_α with the caveat that for the action associated with any given κ node, only the group of agents GR_κ selected to work on that action are required to know anything about the structure and contents of the plan subtree PT_κ being used to do that action. Similarly, $RMB.FSP(PT_\alpha)$ represents that the group GR_α mutually believe that they have a Full SharedPlan using PT_α with the caveat that for the action associated with any given κ

(Group) Know Full Plan Tree

$GrKnowFPT(PT_\alpha) \equiv GrKnowFPT.Top(PT_\alpha) \wedge GrKnowFPT.Rec(PT_\alpha)$,
where

(Group) Know-Full-Plan-Tree: Top-Level Portion

$GrKnowFPT.Top(PT_\alpha) \equiv K_1 \wedge K_2 \wedge K_3 \wedge K_4 \wedge K_\beta \wedge K_\kappa$
where

$$K_1 \equiv MB_\alpha(PT_\alpha^{Elab} = \texttt{NIL}) \Rightarrow PT_\alpha^{Elab} = \texttt{NIL}$$
$$K_2 \equiv MB_\alpha(PT_\alpha^{SelRec} = \texttt{NIL}) \Rightarrow PT_\alpha^{SelRec} = \texttt{NIL}$$
$$K_3 \equiv MB_\alpha(\epsilon set_\alpha = \emptyset) \Rightarrow \epsilon set_\alpha = \emptyset$$
$$K_4 \equiv MB_\alpha(\mu set_\alpha = \emptyset) \Rightarrow \mu set_\alpha = \emptyset$$
$$K_\beta \equiv \begin{cases} (\forall \langle I_\beta, G_\beta, A_\beta\rangle \in \beta set_\alpha) MB_\alpha(\langle I_\beta, G_\beta, A_\beta\rangle \in \beta set_\alpha) \\ \wedge \\ (\forall \langle I_\beta, G_\beta, A_\beta\rangle \in \beta set_\alpha)(MB_\alpha(G_\beta \in GR_\alpha) \Rightarrow (G_\beta \in GR_\alpha)) \end{cases}$$
$$K_\kappa \equiv \begin{cases} (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha) MB_\alpha(\langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha) \\ \wedge \\ (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha)(MB_\alpha(GR_\kappa \subseteq GR_\alpha) \Rightarrow (GR_\kappa \subseteq GR_\alpha)) \end{cases}$$

(Group) Know-Full-Plan-Tree: Recursive Portion

$GrKnowFPT.Rec(PT_\alpha) \equiv (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha) GrKnowFPT(PT_\kappa(GR_\kappa))$

Restricted Mutual Belief in a Full SharedPlan

$$RMB.FSP(PT_\alpha) \equiv \begin{cases} MB_\alpha(FSP.Top(PT_\alpha)) \\ \wedge \\ (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha) RMB.FSP(PT_\kappa(GR_\kappa)) \end{cases}$$

Fig. 3. Definitions of $GrKnowFPT$ and $RMB.FSP$

node, only the agents selected to work on that action are required to participate in the mutual beliefs pertaining to the subplan for that action. More formally, the first part of the $RMB.FSP$ definition requires that the parent group GR_α mutually believe that the top level of their plan satisfies the top-level requirements of an FSP, while the second part recursively requires, for each κ node child of the root node, that the selected subgroup GR_κ holds the mutual beliefs specified by $RMB.FSP$ with respect to the plan subtree PT_κ.

If, instead of satisfying the comparatively weak requirements of $GrKnowFPT$ and $RMB.FSP$, the agents in GR_α had knowledge of the structure and contents of the *entire* plan tree PT_α and, furthermore, they mutually believed that their plan satisfied the requirements of an FSP *at every level of the action decomposition hierarchy* (i.e., $MB_\alpha(FSP(PT_\alpha))$), then the following would be entailed:[10]

$$\begin{array}{l} MB_\alpha(FSP.Top(PT_\alpha)) \\ \wedge \\ (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha) MB_\alpha(FSP(PT_\kappa(GR_\kappa))) \end{array}$$

The top-level portion of the above expression is identical to the top-level portion of the $RMB.FSP$ definition; but the recursive portion of the above expression is much stronger than its $RMB.FSP$ counterpart. In particular, for each κ node child of the root node, it requires that the *entire* group GR_α mutually believe that the selected subgroup GR_κ has a Full SharedPlan using the specified plan subtree PT_κ, whereas the recursive portion of $RMB.FSP$ only requires that the *subgroup* GR_κ participate in mutual beliefs pertaining to that subplan.

Proof of Theorem 2. Given the definitions of $RMB.FSP$, $GrKnowFPT$ and FSP, it suffices to show the following:

$$\text{(2a)}\quad MB_\alpha(FSP.Top(PT_\alpha)) \wedge GrKnowFPT.Top(PT_\alpha) \models FSP.Top(PT_\alpha)$$

$$\begin{array}{ll} \text{(2b)} & (\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha)(RMB.FSP(PT_\kappa(GR_\kappa)) \\ & \qquad \wedge GrKnowFPT(PT_\kappa(GR_\kappa))) \models FSP.Rec(PT_\alpha) \end{array}$$

First, consider (2a), which involves only the top level of the plan tree PT_α. Since $FSP.Top(PT_\alpha)$ is the conjunction of several clauses, preliminary result P3 gives that it is sufficient to find, for each conjunct C, the knowledge condition, K, such that $MB_\alpha(C) \wedge K \models C$. The conjunction of these knowledge conditions defines $GrKnowFPT.Top(PT_\alpha)$ in Fig. 3.

For C of the form, $MB_\alpha(\phi)$, for some ϕ, no knowledge condition is necessary, since $MB_\alpha(MB_\alpha(\phi)) \models MB_\alpha(\phi)$ by preliminary result P4. For C a statement that PT_α^{Elab} or PT_α^{SelRec} is `NIL`, or that ϵset_α or μset_α is empty, K is given by: $MB_\alpha(C) \Rightarrow C$. (See clauses K_1 through K_4 in the $GrKnowFPT.Top$ definition.)

For C of the form, $(\forall x \in X)P(x)$, as in the F_β and F_κ clauses in the FSP definition (where X, a set of nodes, is either βset_α or κset_α), it is sufficient to show that $MB_\alpha((\forall x \in X)P(x)) \models (\forall x \in X)MB_\alpha(P(x)) \models (\forall x \in X)P(x)$.

[10] From the definition of FSP and preliminary results P3 and P7.

The first entailment follows from preliminary result P7, given the knowledge condition, $K' \equiv (\forall x \in X)MB_\alpha(x \in X)$. (See the first conjuncts in the K_β and K_κ clauses in the $GrKnowFPT.Top$ definition.) To get the second entailment, note that $P(x)$ is itself a conjunction: $P(x) \equiv P_1(x) \wedge \ldots \wedge P_n(x)$. Thus, by preliminary result P3, it is sufficient to find, for each conjunct $P_i(x)$, the knowledge condition $K_i(x)$ such that $(\forall x \in X)(MB_\alpha(P_i(x)) \wedge K_i(x)) \models (\forall x \in X)P_i(x)$. The knowledge condition for the second entailment is the conjunction of the individual $K_i(x)$. For $P_i(x)$ being either $(G_\beta \in GR_\alpha)$ or $(GR_\kappa \subseteq GR_\alpha)$, $K_i(x)$ is $MB_\alpha(P_i(x)) \Rightarrow P_i(x)$. (See the second conjuncts in the K_β and K_κ clauses of $GrKnowFPT.Top$.) For $P_i(x)$ being $B.Int.To(G_\beta, A_\beta)$ or $MB_\alpha(\ldots)$, no knowledge conditions are required, given Theorem 1 and preliminary result P4, respectively. This exhausts the cases for (2a). Hence, (2a) holds for any PT_α.

Next, (2b) is proved by induction on the depth of PT_α. In the base case, PT_α has depth 0 (i.e., the root node is the only node in the tree). In particular, κset_α is empty and (2b) is vacuously true. For the recursive case, assume that (2b) holds for all plan trees with depth at most k and suppose that PT_α is some plan tree with depth $k+1$. For each κ child of the root node of PT_α, the plan subtree PT_κ rooted at that node is of depth at most k. Hence, (2b) holds for each such PT_κ. But (2a) also holds for each such PT_κ. Hence, Theorem 2 holds for each such PT_κ, which, given the definition of $FSP.Rec$, is equivalent to (2b) holding for PT_α. $\square$

Theorem 3. $RMB.PSP(PT_\alpha) \wedge GrKnowPPT(PT_\alpha) \models PSP(PT_\alpha)$

Theorem 4. $RMB.SP(PT_\alpha) \wedge GrKnowPT(PT_\alpha) \models SP(PT_\alpha)$

Theorems 3 and 4 are the PSP and SP analogues of Theorem 2. Their proofs (omitted due to space limitations) are intertwined by the presence of the clause, $(\forall \langle I_\kappa, GR_\kappa, A_\kappa \rangle \in \kappa set_\alpha)SP(PT_\kappa(GR_\kappa))$, in the definition of $PSP.Top$. (Recall that $SP \equiv FSP \otimes PSP$.) Furthermore, the presence of associate plan trees complicates the definitions of $GrKnowPPT$ and $RMB.PSP$ (definitions omitted). Nonetheless, the proof of Theorem 3 is similar to that of Theorem 2. (Since the associate plans must be full plans, they are handled by appeals to Theorem 2.) For Theorem 4, the group's knowledge of whether the *Elab* plan tree associated with the root node is `NIL` or not is used to distinguish the FSP and PSP cases, followed by appeals to Theorems 2 and 3.

Theorems 2, 3 and 4 specify knowledge conditions sufficient to ensure that if a group of agents hold a specified set of mutual beliefs, then they necessarily have a SharedPlan. By altering the knowledge conditions, it is fairly straightforward to come up with theorems that are, in spirit, the converses of Theorems 2, 3 and 4.[11] For example, given slightly different knowledge conditions, having an FSP entails restricted mutual belief in that FSP (i.e., $RMB.FSP$).

[11] The different knowledge conditions are due in part to the proofs of these *quasi-converses* using preliminary result P8 where Theorems 2, 3 and 4 use P7.

The theorems presented so far involve the RMB meta-predicates that capture the intuitively appealing idea that only the agents working on any given action need participate in the mutual beliefs pertaining to how that action is being done. Next, some theorems are presented that restrict attention to mutual beliefs held by the entire group. The meta-predicates in these theorems are not recursive, dealing only with the top level of the plan tree. The consequents of such theorems are necessarily weaker, stipulating mutual belief in the mere *existence* of a SharedPlan rather than mutual belief in a SharedPlan using a *particular* plan tree. For example, Theorem 5 states that if the top level of a group's plan meets the requirements of $FSP.Top$ (see Fig. 1), then, given the knowledge conditions modelled by $GrKnowFPT.Top_2$, they necessarily mutually believe that they have *some* Full SharedPlan, as modelled by the $ExistsFSP$ meta-predicate.[12]

Theorem 5. $FSP.Top(PT_\alpha) \wedge GrKnowFPT.Top_2(PT_\alpha) \models MB_\alpha(ExistsFSP(PT_\alpha))$

$GrKnowFPT.Top_2$ and $ExistsFSP$ are defined in Fig. 4. $ExistsFSP$ represents that the top-level of the group's plan meets the requirements of $FSP.Top$ and, in addition, for each κ child of the root node, the selected subgroup has an FSP to do the corresponding action using *some* (existentially quantified) plan tree. Aside from the existentially quantified plan trees in the recursive clause, the definition of $ExistsFSP$ is identical to that of FSP (in Fig. 1).

(Group) Know-Full-Plan-Tree: Top-Level Portion (Version 2)

$GrKnowFPT.Top_2(PT_\alpha) \equiv K_1 \wedge K_2 \wedge K_3 \wedge K_4 \wedge K_\beta \wedge K_\kappa$, where

$$K_1 \equiv (PT_\alpha^{Elab} = \texttt{NIL}) \Rightarrow MB_\alpha(PT_\alpha^{Elab} = \texttt{NIL})$$
$$K_2 \equiv (PT_\alpha^{SelRec} = \texttt{NIL}) \Rightarrow MB_\alpha(PT_\alpha^{SelRec} = \texttt{NIL})$$
$$K_3 \equiv (\epsilon set_\alpha = \emptyset) \Rightarrow MB_\alpha(\epsilon set_\alpha = \emptyset)$$
$$K_4 \equiv (\mu set_\alpha = \emptyset) \Rightarrow MB_\alpha(\mu set_\alpha = \emptyset)$$
$$K_\beta \equiv \begin{cases} (\forall \langle I_\beta, G_\beta, A_\beta\rangle)(MB_\alpha(\langle I_\beta, G_\beta, A_\beta\rangle \in \beta set_\alpha) \Rightarrow (\langle I_\beta, G_\beta, A_\beta\rangle \in \beta set_\alpha)) \\ \wedge \\ (\forall \langle I_\beta, G_\beta, A_\beta\rangle \in \beta set_\alpha)(G_\beta \in GR_\alpha) \Rightarrow MB_\alpha(G_\beta \in GR_\alpha) \end{cases}$$
$$K_\kappa \equiv \begin{cases} (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle) \\ \qquad (MB_\alpha(\langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha) \Rightarrow (\langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha)) \\ \wedge \\ (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha)(GR_\kappa \subseteq GR_\alpha) \Rightarrow MB_\alpha(GR_\kappa \subseteq GR_\alpha) \end{cases}$$

There Exists a Full SharedPlan

$$ExistsFSP(PT_\alpha) \equiv \begin{cases} FSP.Top(PT_\alpha) \\ \wedge \\ (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha)(\exists PT)FSP(PT(GR_\kappa, A_\kappa)) \end{cases}$$

Fig. 4. Definitions of $GrKnowFPT.Top_2$ and $ExistsFSP$

[12] $ExistsFSP$ is used instead of $(\exists PT)FSP(PT(GR_\alpha, A_\alpha))$ because the latter involves second-order problems of existential quantification over an object partially defined using functions.

Proof of Theorem 5. As in the proof of Theorem 2, it is sufficient to deal with each conjunct, C, of $FSP.Top$ individually. For each such conjunct, the corresponding knowledge condition, K, gives that $C \wedge K \models MB_\alpha(C)$.

For C stating that PT_α^{Elab} or PT_α^{SelRec} is `NIL`, or that ϵset_α or μset_α is empty, K is of the form, $C \Rightarrow MB_\alpha(C)$. (See clauses K_1 through K_4 in the definition of $GrKnowFPT.Top_2$.) For C of the form, $MB_\alpha(\phi)$ for some ϕ, no knowledge conditions are required, by preliminary result P4.

For C of the form, $(\forall x \in X)P(x)$, as in the F_β and F_κ clauses of the $FSP.Top$ definition, it is sufficient to show that

$$(\forall x \in X)P(x) \models (\forall x \in X)MB_\alpha(P(x)) \models MB_\alpha((\forall x \in X)P(x)).$$

For the first entailment, since $P(x)$ is a conjunction of clauses, $P_i(x)$, it is sufficient to give a conjunction of knowledge conditions, $K_i(x)$, such that for each i, $(\forall x \in X)(P_i(x) \wedge K_i(x)) \models (\forall x \in X)MB_\alpha(P_i(x))$. For $P_i(x)$ of the form, $MB_\alpha(\ldots)$ or $B.Int.To(G_\beta, A_\beta)$, no knowledge conditions are required, by preliminary result P4 and Theorem 1, respectively. For $P_i(x) \equiv (G_\beta \in GR_\alpha)$, $K_i(x) \equiv (G_\beta \in GR_\alpha) \Rightarrow MB_\alpha(G_\beta \in GR_\alpha)$. For $P_i(x) \equiv (GR_\kappa \subseteq GR_\alpha)$, $K_i(x) \equiv (GR_\kappa \subseteq GR_\alpha) \Rightarrow MB_\alpha(GR_\kappa \subseteq GR_\alpha)$.

The second entailment follows from preliminary result P8, given the knowledge condition, $K' \equiv (\forall x)(MB_\alpha(x \in X) \Rightarrow x \in X)$.

Thus, $FSP.Top(PT_\alpha) \wedge GrKnowFPT.Top_2(PT_\alpha) \models MB_\alpha(FSP.Top(PT_\alpha))$. To conclude the proof, observe that $FSP.Top$ contains the clause,

$$C \equiv (\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha)MB_\alpha((\exists PT)FSP(PT(GR_\kappa, A_\kappa))).$$

But using K' above, under appropriate substitutions, P8 gives the following:

$$C \wedge K' \models MB_\alpha((\forall \langle I_\kappa, GR_\kappa, A_\kappa\rangle \in \kappa set_\alpha)(\exists PT)FSP(PT(GR_\kappa, A_\kappa))). \ \square$$

Theorem 6. $PSP.Top(PT_\alpha) \wedge GrKnowPPT.Top_2(PT_\alpha) \models MB_\alpha(ExistsPSP(PT_\alpha))$

Theorem 7. $SP.Top(PT_\alpha) \wedge GrKnowPT.Top_2(PT_\alpha) \models MB_\alpha(ExistsSP(PT_\alpha))$

Theorems 6 and 7 are the PSP and SP analogues of Theorem 5. The metapredicate $GrKnowPPT.Top_2$ (definition omitted), is more complex than its FSP counterpart. For example, it requires that the group know the "tops" of the associate plan trees, PT_α^{Elab} and PT_α^{SelRec}. Thus, the proof of Theorem 6, while similar to that of Theorem 5, is more complicated, including appeals to Theorem 5 to get that $ExistsFSP$ holds for the $Elab$ and $SelRec$ plan trees. In addition, because the PSP clauses, $P_{\beta_{2.2p}}$ and $P_{\kappa_{2.2p}}$, are not embedded in mutual belief contexts, Theorem 6 requires an additional (strong) condition, namely, that the group's mutual beliefs about the intentions-that specified in these clauses must be *correct*. For Theorem 7, $GrKnowPT.Top_2$ (definition also omitted) only requires that the group be able to distinguish the FSP and PSP cases. The proof then appeals to Theorems 5 and 6, as appropriate.

The Case of Single-Agent Groups. As noted in the previous section, an Individual Plan in V_2 is simply a SharedPlan of a single-agent group. However, a single-agent group is special because that single agent must be the responsible agent for each action in the hierarchy. Consequently, the theorems presented above become simpler in the case of a single-agent group. For example, since

$$Bel(G, FSP(PT_\alpha(\{G\}))) \wedge GrKnowFPT(PT_\alpha(\{G\})) \\ \models RMB.FSP(PT_\alpha(\{G\})),$$

the single-agent version of Theorem 2 may be stated as

$$Bel(G, FSP(PT_\alpha(\{G\}))) \wedge GrKnowFPT(PT_\alpha(\{G\})) \models FSP(PT_\alpha(\{G\})),$$

where $MB(\{G\}, \phi) \equiv Bel(G, \phi)$ by preliminary result P2. Similarly, the single-agent version of the quasi-converse of Theorem 2 may be stated as

$$FSP(PT_\alpha(\{G\})) \wedge GrKnowFPT_2(PT_\alpha(\{G\})) \models Bel(G, FSP(PT_\alpha(\{G\}))),$$

where $GrKnowFPT_2$ (definition omitted) represents slightly different knowledge conditions than $GrKnowFPT$. This result obviates the need for a single-agent version of Theorem 5, the whole point of which was that some agents in the group were likely to be unaware of what others were doing. Similar remarks apply to the single-agent versions of Theorems 3, 4, 6 and 7.

5 Related Work

Many researchers are actively investigating frameworks for reasoning about collaborative activity in multi-agent systems. Although they address similar issues, their different frameworks and perspectives lead to consideration of different technical problems.

Kinny et al. [8] present a framework in which a *joint plan* specifies (1) a recipe for a group action, and (2) an abstract team structure onto which the group doing the action must be mapped. While their joint plan representation implicitly allows abstract plans to be only partially specified, their definition of a *joint intention* requires a fully specified plan and a hierarchy of subordinate intentions analogous to a Full SharedPlan. Kinny et al. do not formally model the group's commitment to elaborate a partial plan into a full plan; but they do provide algorithms for team formation and role assignment that enable agents to simultaneously adopt a fully instantiated plan. The representation allows agents to reason in advance about whether or not a given unstructured group of agents "has the skills to execute" some abstract joint plan, but the question of precisely which knowledge conditions and mutual beliefs are sufficient to ensure that a team actually has a joint intention to do some action is not addressed.

More recently, some of the same authors (Rao et al. [9]) have presented an axiomatization of team knowledge in which teams are treated as first class entities to which team knowledge is directly ascribed. They claim that this *team-oriented approach*, which employs a separate team knowledge modal operator for each team, might enable the designer of a multi-agent system to focus on knowledge relationships between teams without necessarily having to consider in detail the knowledge of individual agents. They plan to "extend the team-oriented model to include the mental attitudes of mutual belief, joint goals, and joint intentions."

Cavedon and Sonenberg [2] focus on *roles* to which the goals "required of a socially committed agent" may be attached. Eschewing "the commitment to joint intention, [they instead] see participation in a team-plan as socially committing [an] agent to the role it adopts in that plan as well as to the other agents involved in the plan." They see roles as providing a way "to specify how the agent should balance competing obligations." In future work, they plan to tie these concepts "more completely to team plans and the process of their selection and execution."

Tambe [13] presents STEAM, an implemented model of teamwork based primarily on Cohen et al.'s theory of Joint Intentions, but informed by key concepts from SharedPlans. Following Cohen et al., a team initially adopts "a joint intention for a high-level team goal" that includes commitments to maintain the goal until it is deemed already achieved, unachievable or irrelevant. The agents then construct a hierarchy of individual and joint intentions "analogous to partial SharedPlans." Tambe notes that as the hierarchy evolves, "if a step involves only a subteam then that subteam must form a joint intention to perform that step", and the remaining team members need only *track* the subteam's joint intention, requiring that they be able to infer whether or not the subteam intends to, or is able to, execute that step. Thus, Tambe informally addresses some of the central issues in this paper.

Stone and Veloso [11] use *locker room agreements* (i.e., pre-determined sets of fixed protocols and *flexible teamwork structures*) to allow teams of agents operating in dynamic domains (e.g., robotic soccer) to avoid much of the negotiation and communication that might otherwise be required to establish and maintain the network of intentions and mutual beliefs that are essential for the collaboration. Rather than hierarchically decomposing the task space in terms of *actions*, the teamwork structures hierarchically decompose the task space in terms of *formations, sub-formations* and *roles*, where each role has an associated set of behaviors. Locker room agreements may stipulate that certain events shall trigger the adoption of new formations and may specify efficient protocols to allow subsets of agents to flexibly switch roles within a formation. The primary concern is to avoid periods of uncoordinated activity arising from inconsistent beliefs about which formation the team is using and which agents are filling which roles. The theorems in this paper apply directly to such concerns.

6 Conclusions

A reformulation of the theory of SharedPlans has been presented that makes the theory more concise and that enables a set of important theorems about agents and their SharedPlans to be formulated and proven. The theorems specify knowledge conditions sufficient to ensure that a group of agents have a SharedPlan if (or only if) they hold a specified set of mutual beliefs. Thus, the theorems may be used to guide the designer of a multi-agent system by clearly specifying the mutual beliefs agents need to establish and the knowledge conditions they need to satisfy as they construct their SharedPlans. The theorems also indicate the potential cost of weakening any of the underlying assumptions. SharedPlan Trees were introduced to make the meta-predicate definitions from the original formulation more concise and to enable precise specification of the knowledge conditions and mutual beliefs appearing in the theorems.

References

1. *Third International Conference on Multi-Agent Systems (ICMAS-98).* IEEE Computer Society, 1998.
2. Lawrence Cavedon and Elizabeth Sonenberg. On social commitment, roles and preferred goals. [1], pages 80–87.
3. D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors. *Handbook of Logic in Artificial Intelligence and Logic Programming.* Clarendon Press, Oxford, 1993.
4. Barbara J. Grosz. AAAI-94 Presidential Address: Collaborative systems. *AI Magazine*, pages 67–85, Summer 1996.
5. Barbara J. Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86:269–357, 1996.
6. Barbara J. Grosz and Sarit Kraus. The evolution of SharedPlans. In A.S. Rao and M. Wooldridge, editors, *Foundations and Theories of Rational Agencies.* 1997. To appear.
7. Merav Hadad. Using SharedPlan model in electronic commerce environment. Master's thesis, Bar Ilan University, Ramat-Gan, Israel, 1997.
8. D. Kinny, M. Ljungberg, A.S. Rao, E. Sonenberg, G. Tidhar, and E. Werner. Planned team activity. In C. Castelfranchi and E. Werner, editors, *Artificial Social Systems*, Lecture Notes in Artificial Intelligence. Springer Verlag, Amsterdam, 1994. Volume 830.
9. A.S. Rao, E. Sonenberg, and G. Tidhar. On team knowledge and common knowledge. [1], pages 301–308.
10. Charles Rich and Candace L. Sidner. Collagen: When agents collaborate with people. In *First International Conference on Autonomous Agents.* Marina del Ray, CA, Feb. 1997.
11. P. Stone and M. Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In J. P. Müller, M. P. Singh, and A. S. Rao, editors, *Intelligent Agents V — Proceedings of the Fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL-98)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Heidelberg, 1999. In this volume.
12. Milind Tambe. Agent architectures for flexible, practical teamwork. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence.* 1997.
13. Milind Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

This article was processed using the LaTeX macro package with LLNCS style