ELSEVIER

Artificial Intelligence 86 (1996) 269-357

Artificial Intelligence

Collaborative plans for complex group action

Barbara J. Grosz [a,*], Sarit Kraus [b,c,1]

[a] Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA
[b] Department of Mathematics and Computer Science, Bar Ilan University, Ramat Gan, 52900 Israel
[c] Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA

Received October 1995

Abstract

The original formulation of SharedPlans by B. Grosz and C. Sidner (1990) was developed to provide a model of collaborative planning in which it was not necessary for one agent to have intentions-to toward an act of a different agent. Unlike other contemporaneous approaches (J.R. Searle, 1990), this formulation provided for two agents to coordinate their activities without introducing any notion of irreducible joint intentions. However, it only treated activities that directly decomposed into single-agent actions, did not address the need for agents to commit to their joint activity, and did not adequately deal with agents having only partial knowledge of the way in which to perform an action. This paper provides a revised and expanded version of SharedPlans that addresses these shortcomings. It also reformulates Pollack's (1990) definition of individual plans to handle cases in which a single agent has only partial knowledge; this reformulation meshes with the definition of SharedPlans. The new definitions also allow for contracting out certain actions. The formalization that results has the features required by Bratman's (1992) account of shared cooperative activity and is more general than alternative accounts (H. Levesque et al., 1990; E. Sonenberg et al., 1992).

1. Introduction

Cooperative problem solving by teams composed of people and computers requires collaboration and communication. Collaboration is a special type of coordinated activity, one in which the participants work jointly with each other, together performing a task or carrying out the activities needed to satisfy a shared goal. Because collaborative action comprises actions by different agents, collaborative planning and activity involve the

* Corresponding author. E-mail: grosz@eecs.harvard.edu.
[1] E-mail: sarit@bimacs.cs.biu.ac.il.

SSDI 0004-3702(95)00103-4

intentions of multiple agents. As a result, collaborative plans cannot be recast simply in terms of the plans of individual agents, but require an integrated treatment of the beliefs and intentions of the different agents involved. Furthermore, the collaborative planning process is a refinement process; a partial plan description is modified over the course of planning by the multiple agents involved in the collaboration. Thus, capabilities for collaboration cannot be patched on, but must be designed in from the start [23,58].

In this paper we present a formal model of collaborative plans that deals more completely with collaboration than previous existing theories of actions, plans, and the plan recognition process. This model grew out of an attempt to provide an adequate treatment of the collaborative behavior exhibited in dialogues [22]. The collaborative property of dialogue affects communication in all modalities and thus is a factor that must be reckoned with in developing more advanced systems for human–computer communication regardless of the modality of communication. Communication and collaboration also play several important roles in multi-agent actions. First, communication provides a means for working together to achieve shared objectives (see for example [6,13,14,16,65]); most multi-agent systems in which the agents need to coordinate their activities incorporate some mechanism for agents to communicate. Second, many multi-agent situations require that agents have an ability to plan and act collaboratively; the avoidance of conflicting actions is a necessary part of such capabilities but is not sufficient in itself (see amongst others [19,35,37,67]). For example, in some cases agents must decide collectively on the approach they will take to acting (i.e., the constituent actions they will perform) and negotiate about responsibilities for performing the subsidiary actions entailed. The model presented here is intended to provide the basis for constructing computer agents that are fully collaborative as well as to provide a framework for modeling the intentional component of dialogue [22,43].

The original formulation of the SharedPlan model of collaborative planning [23] extended Pollack's mental state model of plans [53,54] to the situation in which two agents together form a plan to perform a complex action requiring contributing activity by both agents. Pollack's definition of the individual plan of an individual agent to do an action α includes four constituent mental attitudes: (1) belief that performance of certain actions β_i would entail performance of α; the β_i constituted "a recipe for α"; (2) belief that the agent could perform each of the β_i; (3) intentions to do each of the β_i; (4) an intention to do α by doing the β_i. To define SharedPlans, Grosz and Sidner [23] modified these components to incorporate multi-agent actions and aspects of mental state needed for a pair of agents to coordinate their activities (e.g., mutual belief). In subsequent work [42,45], algorithms were provided for constructing and augmenting SharedPlans in the context of a dialogue.

Although this formulation overcame several problems with previous models of plan recognition for discourse (e.g., the treatment of intentions of one agent toward another agent's actions in applications of speech act theory [1]), it had several problems that emerged when we attempted to apply it to dialogue processing and complex actions in multi-agent environments [42,45]. First, the original model presumed that every multi-agent action decomposed directly into single-agent actions. As a result, the model did not adequately provide for complex activities entailing joint activity at multiple levels or for meshing of individual plans for individual action with collaborative plans for joint action. Second, the model did not account for the commitment of an agent to the success of a collaborative partner's actions.[2] Third, the agents who undertake the development of a collaborative plan often do not know a complete recipe for accomplishing their joint action; the model did not provide a sufficient means of describing the mental state of agents in this situation. The notion of a partial SharedPlan, SharedPlan*, was intended to represent this kind of partiality, but was never specified in any detail. One or more of these limitations applies to alternative models developed subsequently [28,29,40, 61]. The formulation presented in this paper overcomes each of these deficiencies and thus provides a more complete and accurate model than the original formulation and alternative approaches.

Collaborative activity must rest eventually on the actions of individual agents; as a result, the collaborative plans of a group of agents must include as constituents the individual plans of group members. These individual plans may be more complex than those accounted for in Pollack's formulation [53,54] in three ways. First, Pollack's formulation presumed that an agent had a complete recipe for the action it was performing, whereas individual agents, as well as groups of agents, may initially have only partial knowledge of how to perform an action; one function collaborators may serve is to assist an agent in completing a partial recipe. Second, Pollack considered only two types of action relations, generation [20] and enablement; her formalization of "simple plans" uses only generation (and the plans are named "simple" because of this limitation). Balkanski [4] describes several additional action relations that arise in the performance of complex tasks, including sequential and parallel execution. Third, agents may "contract out" to other agents some of the actions to be done. We provide an extended definition of the plans of an individual agent that overcomes these limitations.

Because the formal plan definitions are complex, highly recursive and dependent on several new modal operators, in the next section we provide informal descriptions of several examples that motivate the definitions presented in the paper. We will refer to these examples throughout the paper to illustrate the range of collaborative behavior the model is intended to cover and the way in which it does so. Section 3 provides an overview of the formalization and its major distinguishing features. Section 4 presents auxiliary functions, predicates, and modal operators that are used in the plan definitions. It includes a characterization of the different intentional attitudes that play a role in collaborative planning followed by definitions and axioms for them. It also provides definitions of predicates used to model an agent's ability to perform an action given different degrees of partial knowledge about how to perform the action, a property that is essential to the plan definitions; and, it describes certain processes that play central roles in expanding partial plans to more complete ones. Sections 5 and 6 provide the formal plan definitions. At each stage we discuss those aspects of the resulting the-

[2] The last clause of the original definition was intended to ensure this commitment as well as other properties of coordinated acting. It specified that the agent performing an action intended to do that action to contribute to the performance of the group action. (See the reply [24] for a discussion of replacing the BY operator used in the original definitions by Contributes, and the paper [45] for a definition of Contributes.) However, this approach is inadequate in general. It seemed to work only because the action decomposition in the original paper was single level, i.e., the first deficiency described above. In the current paper we provide a more principled approach.

ory that address the deficiencies described above to provide a more adequate model of collaborative activity. Section 7 examines claims about collaboration made in the AI and philosophical literatures and describes the way they are accounted for within the framework we present. Section 8 compares our formalization with alternative accounts. The paper concludes with a description of several key problems for future research.

2. Examples of collaborative plans

Our primary example comes from the cooking domain; it is a collaboration of two agents preparing a meal together. Although the problem of collaboratively making a meal may not be an ideal application for robots or other kinds of computer systems, we use it throughout this paper for three reasons: (1) unlike tasks such as constructing space stations [18] or network management [45], making a meal is an undertaking for which almost all readers have first-hand knowledge and good intuitions; (2) this task contains the essential elements of the typical collaborative task in which computer systems and robots in particular may be expected to participate: limited resources (including time constraints), multiple levels of action decomposition, a combination of group and individual activities, partial knowledge on the part of each agent, and the need for coordination and negotiation; (3) previous work on plan recognition using this domain provides a baseline for comparison [30,41]. Mapping from the cooking domain to repair and construction tasks in which robots might more naturally participate is straightforward given the properties of the application. Other applications to human–computer collaboration (e.g., network maintenance) may involve little object construction but more extensive use of information exchange actions [43], though information actions are quite prevalent in the meals domain as well. Still other applications (e.g., coordination of search and rescue missions) will fall somewhere between the construction and information-centered tasks.

In particular, we will consider the collaborative planning that arises when two agents, whom we call Kate and Dan, agree to make dinner together. They decide that Kate will make an appetizer, Dan will make lasagna for the main course, and the two of them together will make the dessert. Thus, Kate and Dan must each form one individual plan, Kate's for the appetizer, and Dan's for the lasagna. They need not know the complete details of each other's individual plans, but they need to avoid conflicts arising between these plans. For example, they cannot both use one pan during the same time interval. Thus, as they develop their individual plans, in choosing how to do actions and what resources to use they must consider potential conflicts with each other and communicate if they detect a possible problem. In addition, Kate and Dan together must form a shared, collaborative plan for the dessert. The particular details of how they will do this must be mutually known to both of them. In forming their plans, Kate and Dan may interleave planning and acting; hence, at any stage of their activity, their plans may be only partial. For example, Kate may have decided to make mushroom puffs for the appetizer, but not yet have chosen a recipe for doing so. Alternatively, she may have chosen the recipe, but not yet decided how she will do some of the subtasks. As we will show below, providing for partial knowledge introduces a number of complexities into the formalization. It is necessary, however, to treat such partiality to have a realistic model.

The second example we will use exemplifies situations in which all that an agent knows about how to perform an action is how to find a description of the way to do it. Purchasers of construction kits that include instructions are often in this situation. We will consider the example of Kate buying a bicycle that comes unassembled in a box accompanied by a set of assembly instructions. The plan definitions below treat the case of Kate forming a plan to build the bicycle given this level of partial knowledge about how to do so. They also cover the case of Kate and Dan forming a plan to assemble the bicycle together. Although analogous situations may arise in cooking (e.g., knowing only to look in a cookbook for a recipe), they are more naturally apparent in the construction-kit example.

Finally, we will use three examples to illustrate the use of contracting. Both individual agents and groups of agents may decide that the best way to perform an intended action is to contract out one of the subsidiary actions. Thus, Kate's plan for doing the 30,000 mile maintenance on her car might include contracting out changing the oil. Likewise, if Kate and Dan decide to renovate their house, they might decide to hire someone to redo the floors. In the meals example, Kate and Dan might decide to contract out making the dessert, Dan might contract out some part of making the lasagna, or Kate might contract out part of making the appetizer. Our model of plans provides for all these cases.

3. Overview of the model

The model given in this paper provides a specification of the capabilities to act and mental attitudes that individual agents must have to participate in collaborative activities with one another. In addition, we provide specifications of plans for individual action that are modified from previous accounts to fit with plans for collaborative activity. The specifications are normative and intended to provide the basis for constructing agents that act rationally [57]. Although our work has been informed by an analysis of human collaborations [4], we do not claim to have a complete descriptive account of human collaborative behavior. However, the model has been used to explain a variety of natural-language dialogues [43,44].

We adopt a mental state view of plans [7,54]; agents have plans when they have a particular set of intentions and beliefs. We distinguish *individual plans* that are formed by individual agents from *SharedPlans* that are constructed by groups of collaborating agents. 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. Each agent may have individual intentions and plans to perform some of the subactions. The agents also have individual intentions toward the successful performance of their individual and group actions. We distinguish between complete plans, plans in which the agent or agents have completely determined the way in which they will perform an action, and partial plans.

Our formalization uses a first-order logic augmented with several modal operators, meta-predicates, and action expressions. 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. These meta-predicates are defined in Sections 5 and 6 using terminology developed in Section 4.[3] Although the plan meta-predicates make claims about the mental states of agents, they are not new modal operators. Rather, each is defined in terms of intentions and beliefs of the agents who have plans of the given type. When PIP or FIP holds for an agent, that agent has the collection of intentions and beliefs (including beliefs about connections among the intentions) specified in the meta-predicate definition. When a group of agents has a SharedPlan (i.e., SP and PSP or FSP holds), then members of the group have the individual intentions and beliefs given in the definitions of these meta-predicates.

An interpretation for the logic is an extension of standard Kripke possible worlds semantics in which each possible world is a temporal structure. The modal operators for which accessibility relations are required (defined in Sections 4.2 and 4.3) include the belief operator; operators representing commitment and four intentional attitudes; and two operators related to the performance of actions. Several structures for interpretation proposed in research on intentions and belief (see for example [11,32,55,66]) provide an appropriate base for this logic. However, the establishment of the full set of constraints on accessibility relations required to prove soundness and completeness results with respect to our axioms requires further study. All of the meta-predicates are defined in terms of the modal operators and standard first-order predicates and functions. Hence, accessibility relations are not needed for them.

The formalization is not intended to be directly implemented, for example, by a theorem proving system. Rather, it is intended to be used as a specification for agent design. In this role, the model constrains certain planning processes (e.g., to meet the axioms of intention) and provides guidance about the information that collaborating agents must establish for themselves and communicate with one another. The SharedPlan formalization has been used in the design of a dialogue system to provide the intentional context in which utterances are interpreted and produced [43,44]. Jennings [29] provides an instance of the use of a similar specification in agent design. He modified Cohen et al.'s formalization [40] to provide an explicit model of cooperation for use in the design of industrial multi-agent systems. Jennings's implementation demonstrates the advantages of incorporating explicit models of the intentions required for collaboration into agent design for situations in which agents have incomplete information and operate in dynamic environments with unpredictable events; it also shows the importance of formalizations of collaboration in designing these models.

Fig. 1 lists key components of the mental states of agents when they have a collaborative plan to do a group action. It provides a high-level overview for the formalization

[3] We classify these as meta-predicates because some of their arguments are propositions, but they are not new modal operators. Rather, each meta-predicate refers to a complex formula, namely the combination of predications in its definition.

To have a collaborative plan for an action, a group of agents must have

(1) mutual belief of a (partial) recipe
(2) (a) individual intentions that the action be done
 (b) individual intentions that collaborators succeed in doing the (identified) constituent subactions
(3) individual or collaborative plans for the subactions

Fig. 1. Key components of collaborative plans.

given in this paper, and a framework in which to describe the overall collaborative planning process we envision. This list highlights three principal ways in which plans for group action differ from plans for individual action. First, a collaborative plan to do an action requires that the group of agents agree on the recipe they will use to perform the action (Item (1)). As a result, agents must have processes for deciding which recipes to use, and for combining their individual knowledge of recipes. Second, agents must have commitments not only to their own individual actions, but also to the actions of the group as a whole (Item (2a)) and to those of other agents (Item (2b)). This need requires introduction of a different type of intention from the usual concept of an agent intending to do an action. Third, the plans for group activities may have as components both the plans of individual agents for constituent subactions and the plans of subgroups (Item (3)). As a result, groups must have ways of deciding on the agent or subgroup who will do the subactions.

Any realistic treatment of planned activity must take into account the dynamic nature of plans: plans are developed over time. Agents begin with partial plans and extend them until they have complete plans. Because an agent's beliefs may be faulty or the world may change while the agent is planning or is acting on the basis of a partial plan, partial plans may have to be revised. To address these needs, the specifications given in this paper provide for both individual and collaborative plans to be partial in a number of ways.

As a result of the dynamic nature of plans, at any given moment while the agents are developing and carrying out a collaborative plan, any of the components in Fig. 1 may be incomplete. For example, the agents may have only a partial recipe for the action; or, they may not yet have decided who will do certain constituent subactions and so may have no individual or collaborative plans for those acts; or, an agent may not have determined whether potential new intentions are compatible with its current commitments and so can be adopted. As the agents reason individually, communicate with one another, and obtain information from the environment, portions of their plans become more complete. If agents determine that the course of action they have adopted is not working, then a plan may revert to a more partial state. For example, if an agent is unable to perform an assigned subaction, then the group may need to revise its recipe.

Thus, the list in Fig. 1 is best viewed schematically; it has different instantiations at different times. Each instantiation provides a snapshot of the beliefs and intentions collaborating agents have at that time with respect to their collaborative activity. The formalization must ensure that the snapshots corresponding to a given plan are coordinated. Several different processes are entailed in completing partial plans (analogous to the transitions between snapshots), including processes for identifying recipes, reconciling intentions, and group decision making.

The major focus in this paper is on providing a specification of the agents' mental state that is comprehensive in its treatment of partiality of belief and intention and that handles complex actions. Although the plan definitions entail that the reasoning mechanisms individual agents utilize for extending partial plans have certain properties, the paper does not provide a complete specification of these processes or individual agent design. Rather, the model presumes a basic design for individual agents that accommodates resource constraints and the need to operate in a dynamic world; such designs are proposed in a variety of current planning architectures and formalizations (see for example [10, 56, 60]). In particular, we assume that the agent design incorporates capabilities for managing pending and adopted intentions, including capabilities for deciding when to consider adopting an intention; choosing among competing options; scheduling and executing the intended actions; and monitoring their effects and the state of the world [10]. The definitions given in this paper entail certain constraints on each of these processes, but leave other options open. We discuss ramifications of those choices that affect collaboration as we develop the model.

The formalization in this paper is significantly more complex than those in the original formulation of SharedPlans or in alternative models. The complexity derives from the interaction of partiality and complex actions. In particular, complexity is increased significantly by providing for multiple levels of recipes to be entailed in performing a complex action and by allowing agents to have incomplete knowledge of the recipes to be used at any level. These features are essential for designing rational agents that collaborate. The complete treatment we give also distinguishes our model from the alternatives, as we discuss in Section 8.

Bratman [9] describes three properties that must be met to have "shared cooperative activity": mutual responsiveness, commitment to the joint activity, and commitment to mutual support. In Section 7, we demonstrate that agents that meet the specifications of our formalization will form collaborative plans that have these properties. Furthermore, they will be able to do so even in situations in which their initial beliefs about how to perform actions are less complete and the types of actions they perform are more complex than those described by Bratman. In addition, the account we provide retains the "broadly individualistic" tenor of Bratman's characterization.

The formalization developed in this paper does not require any unreduced notion of joint intentions or "we-intentions" [58, p. 404]. A notion of collective intentionality presents two possible difficulties. Either one must presuppose some kind of group mental state or one must explain how "we-intentions" can be realized in terms of the mental state of individuals. The notion of group mental state not only presents philosophical problems [58], but also appears to necessitate that any agents that might work together in a group be designed together. Searle [58] explains "we-intentions" as attitudes held by all members of a group toward a group action. As we discuss in Section 7, our approach satisfies Searle's conditions for collective action; it does so using only constructs for individual intentions that are useful in situations other than collaborative group action. In Section 8 we compare our approach with other computational accounts of group action, highlighting differences in flexibility and in the range of group activity covered.

4. Supporting definitions and notation

This section of the paper presents the operators, functions and predicates needed to formalize individual and collaborative plans, and describes basic terminology and notation used throughout the paper. Modal operators are used to represent various facets of the mental state of collaborating agents. Predicates and functions of first-order logic are introduced to represent particular properties of actions and the contexts in which they are planned and performed. Act-types are defined to represent two classes of actions that are central to planning. The plan meta-predicates we introduce are defined in terms of these predicates, functions, and act-types as well as the modal operators and meta-predicates. [4] Several of the definitions incorporate references to the plan meta-predicates defined later; the informal descriptions of plan-types given previously should suffice for understanding these references. To assist the reader, Table 1 lists the constructs to be used, the notation used for each category, and the section in which each is first defined.

4.1. Recipe notation, subsidiary predicates and functions

Actions are abstract, complex entities that have associated with them various properties such as action type, agent, time of performance, and other objects involved in performing the action. In most cases, we will use lower-case Greek letters (e.g., α, β, γ) to refer to actions. We assume a set of functions that can be used to obtain the various properties associated with the action; e.g., a function *type* that can be used to refer to the type of action. However, to simplify the presentation, we introduce simpler notation to refer to action properties where possible. For example, we use the term T_α to refer to the time property of the action α; i.e., T_α is shorthand for $time(\alpha)$. In addition, to refer to the complex processes used by agents in planning, we will use terms of the form $\bar{\alpha}(p_1, \ldots, p_n)$ where $\bar{\alpha}$ denotes $type(\alpha)$, and the p_i are parameters that refer to specific properties of α.

Not all actions are realized by events occurring in the world. We distinguish between an action (an abstraction) and its occurrence (a concrete individual that realizes the action). [5] In the formalization as developed so far we have not needed to refer directly to occurrences and so we do not introduce a function from actions to occurrences. However, we do need to predicate occurrence; we use the operator Do, defined later in this section, to do so.

The function *recipe* associates with each action a set of recipes for doing that action; recipe(α) denotes the set of recipes for action α. As in previous work, a recipe is a specification of a group of actions, which we will denote as β_i $(1 \leqslant i \leqslant n)$, the doing of which under appropriate constraints, denoted as ρ_j $(1 \leqslant j \leqslant m)$, constitutes performance of α [4,45,54]. The indices i and j are distinct; for simplicity of exposition, we omit

[4] Many of the operators and predicates include temporal parameters. Because the formalization does not place any special constraints on temporal reasoning, we do not propose the use of any specific temporal logic.

[5] Thus, our actions are like Pollack's [52], but use a representation that differs from her act-type, agent, time triples. Occurrences correspond to acts in Israel et al.'s [27] theory and differ from actions in a similar manner.

Table 1
Summary of notations

Type	Notation	Meaning	Section	Figures
Modal operators	Int.To	intend-to	4.3.1	3
	Int.Th	intend-that	4.3.1	
	Pot.Int.To	potential intention-to	4.3.1	
	Pot.Int.Th	potential intention-that	4.3.1	
	Exec	ability to perform basic-level actions	4.2	
	Commit	commitment to basic-level actions	4.2	
	Do	performance of action	4.2	
Meta-predicates (plans)	FIP	full individual plan	5.1	9, B.1
	PIP	partial individual plans	5.2	12, B.2
	SP	SharedPlans	6.1	17
	FSP	full SharedPlans	6.2	18, B.3
	PSP	partial SharedPlans	6.4	25, B.6, B.7
Meta-predicates (ability)	CBA	can bring about	4.4	6
	CBAG	can bring about group	4.4	7
Meta-predicates (subsidiary)	CONF	actions/propositions conflict	4.3.3	
	GTD	get to do	4.4	8
	CC	can contract	4.4	6
	CCG	group of agents can contract	4.4	7
	BCBA	believe can bring about	5.3	A.1
	MBCBAG	mutually believe can bring about group	6.5	A.3
	WBCBA	weakly believe can bring about	5.3	A.2
	WMBCBAG	weakly mutually believe can bring about group	6.5	A.4
	MP	member of group performs action	6.2	B.5
	SGP	subgroup performs action	6.2	B.5
	FSPC	contracting in FSP	6.2	B.4
	PSPC	contracting in PSP	6.4	B.8
Act-types for planning actions	Select_Rec	agent selects (extends) recipe	4.5	
	Select_Rec_GR	group of agents selects (extends) recipe	4.5	
	Elaborate_Individual	agent extends partial plan	4.5	
	Elaborate_Group	group of agents extends partial SharedPlan	4.5	
Predicates (subsidiary)	single.agent	single-agent action	4	
	multi.agent	group action	4	
	basic.level	basic-level action	4	
Functions	constr	constraints of a context	4	
	recipe	recipes for action	4	
	cost	cost of action	6.3	
	econ	relativize cost (for benefit comparison)	6.3	

the range specifications in the remainder of the paper. We assume each agent has a library of recipes for action types that it collects and updates over time. When planning to perform a given action α, agents use recipes for the action type $\bar{\alpha}$ to construct elements of recipe(α). Agents' libraries may differ, and the successful completion of a collaborative plan may require integrating recipes from the libraries of different agents.

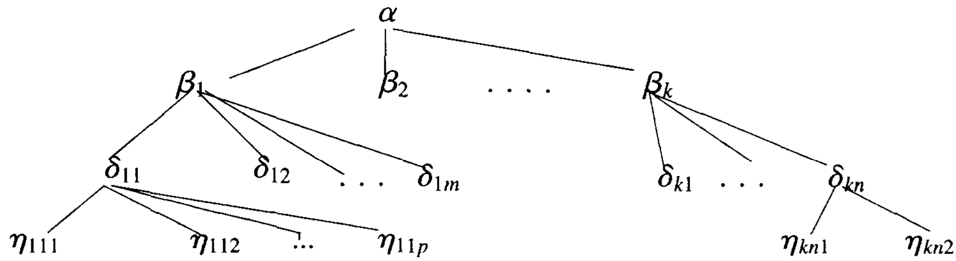

Fig. 2. Recipe tree. β_2, η_{111}, and the other leaf nodes are basic-level actions.

In the definitions, we use the meta-language symbol R_α to denote a particular recipe. That is, to make the definitions more readable, we will write $R_\alpha = \{\beta_i, \rho_j\}$ to indicate that R_α is being used to refer to the set of subsidiary actions and constraints denoted by $\{\beta_i, \rho_j\}$. The subscript (in this case, α) identifies the action for which this is a recipe. To distinguish among alternative recipes for the same action α requires more cumbersome notation, e.g., R^j_α. Because we do not need to make such distinctions in this paper, we do not introduce the additional notation. However, in some cases we may need to refer to a partial (possibly empty) recipe for α; we will use R^p_α to denote such a partial recipe, i.e., $R^p_\alpha \subseteq \{\beta_i, \rho_j\}$.

Recipes may include actions at different levels of abstraction and the parameters of an action may be incompletely specified in a recipe either in the library or in a partial plan. Thus, a recipe may include uninstantiated variables (e.g., for the agent or time of an action) and constraints on these variables. However, for agents to have a complete plan, the parameters must be fully specified in a manner appropriate to the act-type.[6] Lochbaum [43] addresses this issue in the context of using SharedPlans for discourse processing.

The subsidiary actions β_i in the recipe for action α, which we will also refer to as subacts or subactions of α, may either be *basic-level actions* or complex actions. The predicate *basic.level* (α) holds if α is a basic-level action. We assume basic-level actions are executable at will if appropriate situational conditions hold, and do not define this further (see Pollack's argument that this is a reasonable assumption in a computational setting [53]). Furthermore, we assume that agents' beliefs are correct with respect to whether actions are basic level or complex. If an action is basic level, agents believe it is so; if an agent believes an action is basic level, it is.

For those β_i that are complex, there will be recipes, R_{β_i}, that include constituent subactions δ_{iv}. The δ_{iv} may similarly be either basic level or complex. Thus, considering just decomposition and not other constraints represented by the ρ_j, we have the general situation pictured in Fig. 2 in which the leaves of the tree are basic-level actions. We refer to this tree as "the complete recipe tree for α" and use this example for illustrative purposes throughout the paper.

[6] Precisely defining "appropriate to the act-type" raises a variety of complex issue as discussions in several papers make clear (see for example [2,25,48]).

To treat contracting, we will need also to refer to a modified form of recipe tree. The plan definitions provide for contracting out of actions in both individual and SharedPlans. For example, Dan in forming his individual plan for the main course may decide to subcontract to his son one of the tasks required by the recipe he has chosen (e.g., chopping the onions); likewise, while renovating their house, Kate and Dan may subcontract the job of refinishing the floors. When contracting is part of agents' plans, the recipe expansion changes to incorporate contracting actions. In essence, a piece of the complete recipe tree for α is replaced by the recipe tree for the contracting action. For example, if the agent plans to contract out β_j by performing the contracting action γ, then the part of the recipe tree for α below β_j is replaced by a recipe tree for γ. We will refer to this tree as "the extended recipe tree for α"; a formal definition of extended recipe tree is given in Appendix A.1.

Complex actions are further distinguished depending on whether the agent of the action is an individual or a group of agents. The predicate *single.agent* (α) holds if α is a single-agent action, and *multi.agent* (α) holds if the agent of α is a group. All basic-level actions are single-agent actions. To simplify the specifications, we assume that an action is either single-agent or multi-agent, but not both. For example, singing a solo is a single-agent action, whereas singing a duet is [necessarily] a multi-agent action. Likewise, a single agent cooking dinner alone (e.g., Kate's cooking dinner by herself) is a different type of action from multiple agents cooking dinner together (e.g., Kate and Dan cooking dinner).

The intended actions that play a role in individual and collaborative plans are always planned and performed in some context. Various operators, functions, and predicates on actions as well as the plans that are formed for doing them need to refer to this context. We use the notation C_α to refer to the context in which the action α is being done. Two constituents of the context parameter C_α are relevant to this paper. First, C_α includes a "constraints" component that encodes constraints on the performance of α. For example, Kate's individual plan to make the appetizer may have the constraint of being done before a certain time or the constraint of not using a particular pan. The function *constr* maps each context to its constraints component; constr(C_α) denotes the constraints component of the context C_α. Second, C_α includes a representation of the intentional context in which G is doing α. For example, if α is being done as part of doing some higher-level action $\mathcal{A}$, i.e., α is part of the recipe adopted in the plan to do $\mathcal{A}$, then C_α encodes this fact (e.g., using the Contributes relation [45]); alternatively the agent might have chosen to do α to satisfy some independent desire. This constituent of C_α is constructed recursively as an agent chooses recipes and constructs plans for the actions in them. We adopt the notational convention of appending actions in the subscript, e.g., $C_{\beta_i/\alpha}$, to make this fact evident in the definitions that follow. The plan by which α is being done is also part of this constituent; formally, the definitions require that each plan be identified by a name.

4.2. Basic modal operators

We use two standard modal operators for belief, *Bel* and *MB* for belief and mutual belief respectively; they have their usual definitions (see for example [34]). In addition, we specify several modal operators that relate agents and actions: *Exec*, *Commit*, and *Do*, and the intention operators presented in the next section. In this paper, Exec, Commit, and Do are treated as primitive operators; their intended meanings are as follows:

- $\mathrm{Exec}(G, \alpha, T_\alpha, \Theta)$ represents the fact that agent G has the ability to perform action α at time T_α under the constraints Θ. Exec applies only to basic-level actions. The significant difference between this modal operator and Pollack's predicate EXEC [53,54] are the constraints.
- $\mathrm{Commit}(G, \alpha, T_\alpha, T_i, C_\alpha)$ represents the commitment of agent G at time T_i to performing the action α at time T_α. The last parameter, C_α, represents the context in which the agent's commitment is made. Commit also applies only to basic-level actions.
- $\mathrm{Do}(G, \alpha, T_\alpha, \Theta)$ holds when G does action α over time interval T_α under constraints Θ. G may be either a group of agents or a single agent. If T_α is in the past then $\mathrm{Do}(G, \alpha, T_\alpha, \Theta)$ is true if G did α at time T_α.

4.3. Attitudes of intention

4.3.1. Types of intending

The plan definitions require four different intention operators. [7] Two of these, *Int.To* and *Int.Th*, represent intentions that have been adopted by an agent. The other two, *Pot.Int.To* and *Pot.Int.Th*, are variations of the first two that are used to represent *potential* intentions. Potential intentions are used to represent an agent's mental state when it is considering adopting an intention but has not yet deliberated about the interaction of that intention with the others it currently holds. Potential intentions motivate an agent to weigh different possible courses of actions or options [10]. They thus represent intentions that an agent would like to adopt, but to which it is not yet committed. Potential intentions typically arise in the course of means-ends reasoning. Attitudes of Pot.Int.To stem from an agent's deliberations about how to do some action it is committed to performing. Pot.Int.Th's derive from the need to ensure that collaborating agents' plans mesh correctly [9].

Int.To and Pot.Int.To are used to represent an agent's *intentions to* do some action; Int.Th and Pot.Int.Th are used to represent an agent's *intention that* some proposition hold. In the definitions that follow, $\mathrm{Int.To}(G, \alpha, T_i, T_\alpha, C_\alpha)$ represents agent G's intention at time T_i to do action α at time T_α in the context C_α; $\mathrm{Int.Th}(G, prop, T_i, T_{prop}, C_{prop})$ represents an agent G's intention at time T_i that a certain proposition *prop* hold at time T_{prop} in the context C_{prop}. C_{prop} is the analogue for propositions of C_α for actions.

The commonality between intentions-to and intentions-that is that both commit an agent not to adopt conflicting intentions [64] and constrain replanning in case of failure [7]. The significant distinction between them 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.

An Int.To commits an agent to means-ends reasoning [7] and, at some point, to acting. In contrast, an Int.Th does not directly engender such behavior. Int.Th's form

[7] Vermazen [63] describes the need to consider more than a single attitude of intention.

the basis for meshing subplans, helping one's collaborator, and coordinating status updates [9,40,58] all of which play an important role in collaborative plans; any of these functions may lead to the adoption of an Int.To and thus indirectly to means-ends reasoning.

An agent can only adopt an intention-to toward an action for which it is the agent. In addition, the agent must believe it will be able to do the action at the appropriate time. In this paper, we adopt the strong position that an agent must believe it *can successfully* perform any action it intends to do. As others have noted (e.g., Pollack [51, p. 38], and others cited there), this stance is too strong. Although it is clear that the agent can*not* believe it is *incapable* of succeeding, it may have doubts about the success of the intended action [7]. Thus, our formalization would be better served by a probabilistic approach to the modeling of ability, but we have not identified a suitable computational model. Such an approach would enable us to replace "flat-out" belief [7, pp. 36ff] with the more realistic requirement that an agent's belief in the likelihood of success of its actions be above a certain threshold for the agent to be able to intend to perform the act.

The means-ends reasoning and knowledge constraints on intentions-to lead to an asymmetry between Int.To and Int.Th. Any proposition, *prop*, can be converted to an action, *Achieve(prop)*, where *Achieve* is a function that maps arbitrary propositions to generalized actions that have that proposition as an effect. However, an Int.Th$(G, prop, T_i, T_{prop}, C_{prop})$ does not necessarily entail an Int.To$(G, Achieve(prop), T_i, T_{prop}, C_{prop})$, because an agent may be unable to do means-ends reasoning about *Achieve(prop)* or may be incapable of carrying out any particular action that instantiates the *Achieve(prop)*.

The differences between the four types of intentional attitudes may be illustrated with the dinner example introduced in Section 2. Dan and Kate's collaborative plan consists of Kate making an appetizer, Dan the main course, and the two of them together making the dessert. Thus, their plan to make dinner includes Kate having an intention to [Int.To] make the appetizer, a belief that she will be able to do so, and an individual plan for doing so; likewise, it includes Dan having an intention to [Int.To] make the main course, a belief that he can, and an individual plan for doing so; in addition, it includes their having a collaborative plan to make the dessert. The plan for making dinner will also include Dan's intention that [Int.Th] Kate "be able to make" the appetizer, and Kate's intention that [Int.Th] Dan "be able to make" the main course.

If Kate has decided to make mushroom puffs for the appetizer, but not yet chosen a recipe for doing so, her individual plan will be partial. It will include an Int.To select or construct a recipe for mushroom puffs. The identification of the recipe entails means-ends reasoning. As she does this means-ends reasoning, she will determine actions she needs to perform to make the mushroom puffs and will adopt potential intentions to [Pot.Int.To] perform these actions. The potential intentions will become part of a deliberation process [10] and through that process may become Int.To's.

4.3.2. Modal operators for attitudes of intention

The definition of Int.To is given in Fig. 3. The first part of this definition [Clause (1)] deals with the case of an agent intending to do an action that is basic level. Two standard

$\text{Int.To}(G, \alpha, T_i, T_\alpha, C_\alpha)$

(1) $[\text{basic.level}(\alpha) \wedge \text{Bel}(G, \text{Exec}(G, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_i) \wedge$
$\text{Commit}(G, \alpha, T_\alpha, T_i, C_\alpha)] \bigotimes$

(2) $[\neg\text{basic.level}(\alpha) \wedge$

(a) $[(\exists P, R_\alpha)$

(1) $\text{FIP}(P, G, \alpha, T_i, T_\alpha, R_\alpha, C_\alpha)] \bigotimes$

(b) $[(\exists P, P_{elab}, T_{elab}, R_{elab})$

(1) $[\text{PIP}(P, G, \alpha, T_i, T_\alpha, C_\alpha) \wedge$

(2) $\text{FIP}(P_{elab}, G, \text{Elaborate\_Individual}(P, G, \alpha, T_i, T_\alpha, C_\alpha), T_i, T_{elab}, R_{elab}, C_{elab/\alpha}) \wedge$

(3) $\text{Int.To}(G, \text{Elaborate\_Individual}(P, G, \alpha, T_i, T_\alpha, C_\alpha), T_i, T_{elab}, C_{elab/\alpha})]]]$

Fig. 3. The definition of Int.To.

constraints [7] are represented in this part of the definition: that the agent be committed to doing the action and that the agent believe it can execute the action.

The second part of the definition addresses the case of an agent intending to do an action that is complex. Two issues arise, each stemming from needing to generalize the constraints for basic-level actions. First, of what does the commitment component of intention consist in this case; in particular, what other commitments does it entail? Second, to what extent must an agent know how to perform the action and believe that it will be able to do so?

For the agent to perform the intended action completely,[8] it must have a full recipe for the action; i.e., it must have recipes for all of the complex actions in the recipe for α that it is doing itself and for any complex contracting actions that it plans to use, and must likewise have recipes for any complex actions in these subsidiary recipes, and so on recursively to basic-level actions. In addition, the agent must have intentions-to do all of the basic-level actions in the full recipe. From the base case [Clause (1)], the agent must believe it will be able to execute each of these basic-level actions and must be committed to doing so. This degree of knowledge and commitment is too restrictive for the initial adoption of an intention-to. Such a restriction would prohibit partial plans or incremental planning, both of which are important to our approach.

However, the agent must have at least some minimal commitment to the complex act. Furthermore, we argue, this commitment is only meaningful if the agent has some minimal knowledge about how to identify a recipe for the act (either through construction or by choice from several options) and a commitment to identify a recipe. In addition, the agent must believe that the recipe it will select is one it will be able to execute. These additional constraints follow from intentions-to engendering means-ends reasoning [7]. If an agent does not have a recipe for α and furthermore has no idea at all about how to find or construct a recipe, then it cannot do any means-ends reasoning about α.

The definition separates its treatment of an agent intending to do a complex action into two parts. We discuss the major constraints imposed by each part here; they derive from the plan definitions given in Section 5. If the agent has a complete individual plan to do the action [Clause (2a)], then the strongest constraints described above are met.

[8] More precisely, this requirement holds for the agent to perform the action intentionally; the agent might accidentally perform the action, a situation not of concern here.

Clause (2b) covers the case in which the agent's knowledge of how to do the act and commitment are more partial. The partial individual plan in Clause (2b1) establishes, minimally, that the agent has chosen and is committed to some way of identifying a recipe for α. Clause (2b2) represents the agent's commitment to completing this partial plan. [9] Clause (2b3) is entailed by Clause (2b2), as discussed in the next section; we include it explicitly to emphasize that this particular modality must hold of the agent; i.e., that the agent adopts an additional intention-to. Although not formally required, this clause is useful when using the formalization in agent design.

Although the constraints in Clause (2b) might seem strong, they are actually quite weak. Together they ensure only that the agent is committed to meaningful means-ends reasoning about α. First, the partial plan of Clause (2b1) does not require that the agent have a recipe for α, only that the agent know some way of identifying such a recipe. For example, Kate's recipe for identifying a recipe for the appetizer might consist only of calling her mother, or of looking in a cookbook, or of doing both and then deciding whether she wants to use any of those recipes or her old favorite one; or Kate might have a more meta-level recipe for finding a recipe, one that consists of waiting until later and then deciding which of these three options to pursue. Thus, the constraint represented by this clause amounts to stipulating that the agent cannot be at a complete loss about how to find a recipe, nor can the agent be uncommitted to selecting a complete recipe (eventually).

Second, the Elaborate_Individual term in Clause (2b2) refers to general complex planning actions for expanding partial plans (see Section 4.5). Although these actions may at some level refer to recipes in the partial plan, they do not depend on them. Thus, the full plan in Clause (2b2) amounts to a specification that an agent be committed to invoking one of its planning procedures. Substituting the weaker constraint of a partial individual plan for elaboration into Clause (2b2) would lead to an infinite regress. The agent's plan to elaborate the elaboration could be partial, and so on infinitely; none of the intentions to elaborate would need to ground out in Clause (2a). A modification that stopped the recursion after a finite number of steps might seem a more reasonable model. For example, we might stop the recursion at the second step, allowing the agent to have a partial plan for the elaboration, but requiring a full plan for the elaboration of the elaboration. But to do this would require introducing a special stronger notion of intention, and at no extra gain. By allowing the elaboration process itself to be quite general and include exploration of disjunctive possibilities, we achieve the same result.

In this paper, we do not define Int.Th in terms of more basic constructs. Instead we provide a set of axiom schemas [10] that specify the ways in which intending-to and intending-that interact with each other and with belief. Although we cannot provide a full set of axioms, the central axioms needed for our formalization are given in the next section; additional axioms for intending-that are discussed in Section 6.

The meals example illustrates many of the differences in reasoning and reconciliation that are engendered by intentions-to and intentions-that. Kate's individual plan to make the appetizer leads her to means-ends reasoning about the way in which to do each of

[9] The plan-name parameters in (2b1) and (2b2) are identical to ensure that the particular plan is completed.

[10] All the axioms that are specified in this paper are actually axiom schemas.

the actions in the recipe, including reasoning about reducing each of the steps in the high-level recipe (e.g., a recipe like one that might appear in a cookbook) to basic-level actions. The intentions-to do each of the actions entailed by the recipe are her own and thus under the control of her reconciliation processes. In contrast, if she forms a collaborative plan with Dan to make the dessert, then Dan will be responsible for some of the subactions, say preparing the egg whites, chopping nuts and whipping cream for a cake icing. Dan forms intentions-*to* perform each of these subsections; Kate has intentions-*that* he will be able to do them. Kate can aid Dan, by not presenting him with the need to reconcile additional intentions (e.g., not asking him to plant the vegetable garden at the same time); by being helpful (e.g., discussing approaches to problems Dan encounters, for example the egg whites not whipping); and by not doing things that interfere with his actions (e.g., not using the mixer when he needs it). However, she will not otherwise do any means–ends reasoning about how to chop nuts or to whip egg whites or cream. [11] Furthermore, Dan will be reconciling other intentions with intentions to do the actions he is contributing to the dessert making. His obligations and desires will determine whether he watches the evening news or works on the cake.

4.3.3. Axioms for intention operators

In this section, we present several basic axioms relating intentions to one another and to beliefs. These axioms further constrain the design of computer agents for rational action.

A major role intentions play in planning is to make the process more tractable by constraining agent deliberation [7,8]. Significant focusing of attention results from the constraint that an agent cannot hold two conflicting intentions simultaneously. The axiom stating this constraint uses the meta-predicate *CONF*. $\mathrm{CONF}(\alpha,\beta,T_\alpha,T_\beta,\Theta_\alpha,\Theta_\beta)$ 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. [12] Thus, $\mathrm{CONF}(\alpha,\beta,T_\alpha,T_\beta,\Theta_\alpha,\Theta_\beta)$ is true in the following three situations: [13]

(i) α and β are actions for which G is the agent. G's performance of α under the constraints Θ_α conflicts with its performance of β under constraints Θ_β. This conflict may arise either because the performance of one of the actions will bring about a situation in which it is no longer the case that the agent can perform the other action (formalized using the meta-predicate CBA defined in the next subsection), or because the constraints on the performance of the two actions are in conflict. In either case, this conflict can be formalized as

$$\neg(\exists R_\alpha,R_\beta)([\mathrm{CBA}(G,\alpha,R_\alpha,T_\alpha,\Theta_\alpha)]\wedge[\mathrm{CBA}(G,\beta,R_\beta,T_\beta,\Theta_\beta)]).$$

[11] At least she will not necessarily do such reasoning. She may become involved if Dan needs help determining an appropriate recipe [43].

[12] The definition of CONF encompasses only conflicts among propositions and the performance of single-agent actions. Conflicts that arise from group actions are handled through the inclusion of intentions-that (Int.Th) propositions in the plan definitions.

[13] If α is a proposition then the set of constraints Θ_α is empty; likewise, if β is a proposition then Θ_β is empty.

Axiom (A1):

$$\begin{array}{l}\text{Bel}(G, \text{CONF}(\alpha, \beta, T_\alpha, T_\beta, \text{constr}(C_\alpha), \text{constr}(C_\beta)), T_i) \Rightarrow \\ \quad \{[\text{Int.Tx}(G, \alpha, T_i, T_\alpha, C_\alpha) \Rightarrow \neg(\text{Int.Ty}(G, \beta, T_i, T_\beta, C_\beta))] \wedge \\ \quad\ [\text{Int.Ty}(G, \beta, T_i, T_\beta, C_\beta) \Rightarrow \neg(\text{Int.Tx}(G, \alpha, T_i, T_\alpha, C_\alpha))]\}\end{array}$$

α/β may be either an action (if Int.Tx/y = Int.To) or a proposition (if Int.Tx/y = Int.Th)

Fig. 4. Axiom schema to avoid conflicting intentions.

(ii) α is an action for which G is the agent; β is a proposition. Either G's performance of α will cause β not to hold, or conversely, if β holds, then G cannot perform α. Formally,

$$[\text{Do}(G, \alpha, T_\alpha, \Theta_\alpha) \Rightarrow \neg\beta] \vee [\beta \Rightarrow \neg\exists R_\alpha[\text{CBA}(G, \alpha, R_\alpha, T_\alpha, \Theta_\alpha)]].$$

(iii) α and β are propositions which cannot simultaneously hold, i.e., $[\beta \leftrightarrow \neg\alpha]$.

Each of these conflict situations may be illustrated with the dinner making example. Suppose that Dan and Kate have only one lasagna pan, all lasagna recipes require using this pan, and all recipes for making spinach squares also require the pan. There is a conflict of the first sort between Dan's making lasagna and Kate's making spinach squares during the same time interval. A conflict of the second sort arises between Dan's making lasagna (the action α) and Kate's being able to make spinach squares (the proposition β). The second type of conflict also occurs between the proposition that Dan and Kate are making dinner at time T (i.e., the proposition $\text{Do}(\{\mathit{Dan}, \mathit{Kate}\}, \mathit{make\text{-}dinner}, T, \Theta)$ and Kate's playing basketball at time T. The third conflict situation holds between the pan being clean and empty and the lasagna ingredients being in the pan.

As formalized by the axiom in Fig. 4, an agent cannot knowingly hold conflicting intentions; neither conflicts between intentions of the same type (Int.To or Int.Th) nor conflicts between an Int.To and an Int.Th are allowed. If an agent is unaware of (i.e., does not know about) an existing conflict, this axiom does not apply. Two properties of our formalization—that agents may have partial plans and that collaborating agents do not need to know the complete plans of their partners—are sources of potential unknown conflicts. We discuss the rationale for these properties and the ramifications for agent design in the introduction to Section 6.

Within the formalization we provide, the need to check intention conflicts arises most prevalently in the transitions needed to form complete plans from partial ones. In particular, the transition process includes an agent turning potential intentions into actual intentions. If the agent discovers a conflict between adopting a new intention as a full-fledged intention and intentions it already has, it must reconcile between the competing intentions. The reconciliation process is part of the basic agent design presumed by this paper (as discussed in Section 3). The process of transforming potential intentions to actual intentions is part of the plan elaboration process represented by the act-type term Elaborate_Individual which is described in Section 4.5.

In the dinner making example, Kate's individual plan for making the appetizer cannot produce intentions that conflict with her intention that Dan succeed in making lasagna. For instance, once Kate knows that Dan intends to make lasagna for the main

Axiom (A2): If α is a basic-level action and G believes it intends to do α then G really intends to do α:

$$(\forall \alpha, T_i, T_\alpha)[\text{basic.level}(\alpha) \wedge \text{Bel}(G, \text{Int.To}(G, \alpha, T_i, T_\alpha, C_\alpha), T_i) \Rightarrow \\ \text{Int.To}(G, \alpha, T_i, T_\alpha, C_\alpha)]$$

Theorem (T1):

$$(\forall \alpha, T_p, T_\alpha)[\text{FIP}(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha) \Rightarrow \text{Int.To}(G, \alpha, T_i, T_\alpha, C_\alpha)]$$

Axiom (A3):

$$\text{Bel}(G, \text{Int.Th}(G, prop, T_i, T_{prop}, C_{prop}), T_i) \Rightarrow \text{Int.Th}(G, prop, T_i, T_{prop}, C_{prop})$$

Axiom (A4):

$$\text{Bel}(G, \text{Pot.Int.Tx}(G, \alpha, T_i, T_\alpha, C_\alpha), T_i) \Rightarrow \text{Pot.Int.Tx}(G, \alpha, T_i, T_\alpha, C_\alpha)$$

where Pot.Int.Tx is equal to Pot.Int.To if α is an action or Pot.Int.Th if α is a proposition.

Fig. 5. Intentions and beliefs.

course, she cannot intend to do an action that would use their sole lasagna pan; for example, under the assumptions described above, she cannot intend to make spinach squares. If potential intentions lead to consideration of adopting such conflicting intentions, the reconciliation process will cause one intention to be dropped; as a result, some portion of the SharedPlan will become, or remain, partial. For example, Kate might initially consider making spinach squares for the appetizer, but drop that potential intention when she realizes she cannot do so without conflicting with Dan's ability to make lasagna. She might then adopt a potential intention to make mushroom puffs instead.

Fig. 5 gives the basic axioms relating beliefs and intentions. Axiom (A2) stipulates that an agent actually intends to do any basic-level action that it believes it intends to do.

Theorem (T1) asserts that an agent has an intention to do the complex action α whenever the agent has a full individual plan to do α. This axiom follows in a straightforward manner from the definition of FIP: if the agent has intentions to do all of the basic-level actions required for doing α and furthermore intends these actions in the context of a plan to do α, then it follows that the agent intends to do α. However, an agent does not always have an intention when it has a partial plan. In particular, from the definition of Int.To, if the agent's plan is partial and it has not formed a complete plan to elaborate the partial plan (e.g., it has not reconciled the relevant intentions), then it will not yet have an intention to do α. Furthermore, an agent might have successfully reconciled an intention to do α, but have only some and not all of the beliefs required for a partial plan; in this case, it does not yet have a full-fledged intention to do α.

Finally, Axiom (A3) stipulates that an agent actually intends that a proposition holds if it believes it so intends and Axiom (A4) states that if an agent believes it has a potential intention then it really does.

4.4. Meta-predicates for the ability to act

To represent the knowledge agents have about their own and their collaborators' abilities to perform actions in a plan, we introduce two related meta-predicates: the single-agent meta-predicate *CBA* ("can bring about") and the multi-agent meta-predicate *CBAG* ("can bring about group"). Three additional meta-predicates are defined to treat contracting and the knowledge agents need to have about their own and their contractors' abilities. The meta-predicates *CC* ("can contract") and *CCG* ("group of agents can contract") specify the conditions under which agents can do an action by contracting it out to other agents. Contracting in turn depends on an agent believing that by doing one action (and thus bringing about a certain state of affairs), it can get a different agent to perform another action; the meta-predicate *GTD* ("get to do") is used to represent this state of affairs.

The meta-predicates CBA and CBAG presuppose an omniscient perspective from which the recipes for action α and all its constituent acts are known; e.g., the complete set of recipes for the full recipe tree in Fig. 2 or, if contracting is involved, for the extended recipe tree, is known. The plan definitions, however, use these meta-predicates only within belief or intention contexts that existentially quantify the recipe. Although CBA and CBAG are very strong, the result of embedding them in belief or intention contexts is a very weak statement; in particular, only belief in the existence of some recipe (or intention that one exist) is claimed, not identification of a particular recipe. This result is exactly what is needed for certain portions of partial plans and for representing what collaborating agents know about their collaborators' abilities.

However, the knowledge agents have about their own ability to perform actions in their full individual plans or full SharedPlans is greater than that represented by CBA and CBAG embedded inside belief contexts, but less than that represented by the unembedded meta-predicates. The definitions of FIP and FSP implicitly encode this level of ability knowledge. In a similar manner, the definitions of PIP and PSP implicitly encode agents' knowledge about their own abilities to act when they know only partial recipes for subsidiary actions; e.g., when they know part of the recipe tree of Fig. 2, but not the complete tree.

The remainder of this subsection presents CBA, CBAG, and the meta-predicates used to accommodate subcontracting within their definitions. Sections 5.3 and 6.5 discuss additional constraints on agents' knowledge about actions they perform themselves in individual plans and collaborative plans respectively. To aid in designing agents, we define subsidiary meta-predicates that explicitly represent this level of ability knowledge in Appendix A.

The meta-predicate CBA, given in Fig. 6, represents an agent G's ability to do the action α using the recipe R_α at time T_α and under constraints Θ. The agent may either do all of the subactions in R_α itself, or may contract out some of them.[14] The constraints

[14] Clause (2b1) of the CBA definitions includes simple disjunction (or), rather than exclusive or. The use of or reflects the possibility that an agent may be able to bring about the subsidiary action β_i in more than one way; it may be able to do the action by itself, to contract it out, or both to perform it itself and to contract it out. Because CBA represents ability rather than any commitment to act in a particular way, or is the appropriate logical connective.

$\text{CBA}(G, \alpha, R_\alpha, T_\alpha, \Theta)$

(1) $[\text{basic.level}(\alpha) \wedge \text{Exec}(G, \alpha, T_\alpha, \Theta)] \otimes$
(2) $[\neg\text{basic.level}(\alpha) \wedge$
  (a) $R_\alpha = \{\beta_i, \rho_j\} \wedge R_\alpha \in \mathit{Recipes}(\alpha) \wedge$
  (b) $(\forall \beta_i \in R_\alpha \exists T_{\beta_i})[$
    (1) $[(\exists R_{\beta_i})\text{CBA}(G, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\})] \vee$
    (2) $[\text{CC}(G, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\})]]$

$\text{CC}(G, \beta, T_\beta, \Theta)$

$(\exists G_c, \gamma, T_\gamma, R_\gamma)[$
(1) $\text{CBA}(G, \gamma, R_\gamma, T_\gamma, \Theta) \wedge$
(2) $\text{GTD}(G, \gamma, T_\gamma, G_c, \beta, T_\beta, \Theta)]$

Fig. 6. Definition of CBA (can bring about) and CC (can contract).

argument Θ encodes various situational constraints on the performance of the action; these constraints derive from the particular recipe being used and the context in which the action is being done. In particular, subactions inherit the constraints of the actions of which they are a part [Clause (2b1)]. For example, if Dan decides to make homemade tomato sauce for the lasagna in the dinner he and Kate are preparing, then in deciding whether he can bring about the action of making the sauce, he must consider the time constraints on the overall action of making dinner (e.g., being done by 7 P.M.) as well as any constraints within the sauce recipe itself (e.g., having the ingredients, being able to use the pressure cooker for an hour).

In the case of an agent doing the action itself, if the action is basic level, CBA reduces to Exec [Clause (1)]. If the action is not basic level, then CBA is determined recursively on the basis of the recipe R_α [Clause (2)]. For each subsidiary action β_i in the recipe, this recursion provides either for the agent to carry out β_i itself [Clause (2b1)] or to contract it out to another agent [Clause (2b2)]. The recursion on acts the agent performs itself ends in basic-level acts from the complete recipe for α; for subsidiary actions that the agent contracts out, a contracting action and a recipe for performing it enter into the recursion. The recursion ends in an extended recipe tree with basic-level acts that the agent performs itself as leaves. Thus, CBA requires that the agent be able to execute the basic-level acts entailed in performance of the original action—with the exception of those actions contracted out or entailed by contracted actions—according to the given recipe and under the specified constraints. For those subsidiary actions that are contracted out, the agent must be able to execute the basic-level actions in, or entailed by, the recipe for the agent's contracting action (again with the exception of any that are contracted out or entailed by contracted actions).

The meta-predicate CC specifies the conditions under which an agent is able to contract out an action β to another agent. In particular, the original agent, G, must identify a contractor (G_c) and some action γ that it (G) can perform [Clause (1)] such that by doing the action γ, G can get G_c to do β at the appropriate time and under the appropriate constraints [Clause (2)].

CBAG(GR, α, R_α, T_α, Θ)

(1) $R_\alpha = \{\beta_i, \rho_j\} \wedge R_\alpha \in Recipes(\alpha) \wedge$
    (a) $(\forall \beta_i \in R_\alpha \exists T_{\beta_i})$ [
(2) [[single.agent(β_i) $\wedge$
    (a) $(\exists G_{\beta_i} \in \mathrm{GR}, R_{\beta_i})$
        (1) CBA(G, β_i, R_{β_i}, T_{β_i}, $\Theta \cup \{\rho_j\}$)] $\otimes$
(3) [multi.agent(β_i) $\wedge$
    (a) $(\exists \mathrm{GR}_{\beta_i} \subseteq \mathrm{GR}, R_{\beta_i})$
        (1) CBAG(GR_{β_i}, β_i, R_{β_i}, T_{β_i}, $\Theta \cup \{\rho_j\}$)]] $\bigvee$
(4) [CCG(GR, β_i, T_{β_i}, $\Theta \cup \{\rho_j\}$)]]

CCG(GR, β, T_β, Θ)

(1) $(\exists G_c, \gamma, T_\gamma, R_\gamma)$
(2) [single.agent(γ) $\wedge$
    (a) $(\exists G_\gamma \in \mathrm{GR})$
    (b) CBA(G_γ, γ, R_γ, T_γ, $\Theta \cup \{\rho_j\}$) $\wedge$
    (c) GTD(G_γ, γ, T_γ, G_c, β, T_β, Θ)] $\bigvee$
(3) $(\exists G_c, \gamma, T_\gamma, R_\gamma)$
    [multi.agent(γ) $\wedge$
    (a) $(\exists \mathrm{GR}_\gamma \subseteq \mathrm{GR})$
    (b) CBAG(GR_γ, γ, R_γ, T_γ, Θ) $\wedge$
    (c) GTD(GR_γ, γ, T_γ, G_c, β, T_β, Θ)]

Fig. 7. Definition of CBAG (can bring about group) and CCG (can contract group).

The definition of CBAG, a meta-predicate for groups and multi-agent actions analogous to CBA, is given in Fig. 7. The major difference between CBAG and CBA is that some of the actions in the recipe for the (necessarily) complex action α, i.e., some of the β_i, may be multi-agent actions. For these actions, there must be a subgroup of the whole group that can bring about the action [Clause (3)]; i.e., CBAG recurs on a subact with a subgroup. For those β_i that are single-agent actions, there needs to be a member of the group who has an ability to perform the action [Clause (2) of CBAG]. Furthermore, CCG generalizes CC to handle situations in which a group of agents does the contracting [Clause (3) of CCG]. The single-agent case of CCG is like CC but requires identification of a group member to do the contracting. [15]

The meta-predicate GTD (Fig. 8) treats both single- and multi-agent actions. Thus, the arguments G and G_c may refer either to a single agent or a group of agents and the arguments α and γ may be either single- or multi-agent actions. GTD holds of two agents or groups, G and G_c, two actions γ and α, the times of those actions, T_α and T_γ, and a set of constraints Θ if as a result of G's doing γ at T_γ, G_c commits to doing α at T_α. The constraints Θ originate as constraints on the performance of α. However,

[15] Because an action may be either single-agent or multi-agent but not both and agents' beliefs are correct with respect to whether actions are single- or multi-agent, exclusive or is used in Clause (2a1) of the CBAG definition. However, the group GR may be able to contract out a particular β_i in several different ways. Some individual agent may be able to use a single-agent action γ, or a subgroup might use a (different) action γ which is multi-agent, or both types of contracting actions may be possible. This is indicated by the use of or rather than exclusive or in Clause (2c) of the CCG definition (Fig. 7).

GTD$(G, \gamma, T_\gamma, G_c, \alpha, T_\alpha, \Theta)$

(1) [single.agent$(\alpha) \wedge$
  (a) Do$(G, \gamma, T_\gamma, \Theta) \Rightarrow$
  (b) $[(\exists T_i)$ Int.To$(G_c, \alpha, T_i, T_\alpha, C_{\alpha/f(\alpha,\gamma)}) \wedge$
  (c) $(\exists R_\alpha)$ CBA$(G_c, \alpha, R_\alpha, T_\alpha, \Theta)]] \otimes$
(2) [multi.agent$(\alpha) \wedge$
  (a) Do$(G, \gamma, T_\gamma, \Theta) \Rightarrow$
  (b) $[(\exists P_\alpha, T_i, R_\alpha)$
  (c) SP$(P_\alpha, G_c, \alpha, T_i, T_\alpha, C_{\alpha/f(\alpha,\gamma)}) \wedge$
  (d) CBAG$(G_c, \alpha, R_\alpha, T_\alpha, \Theta)]]$

Fig. 8. Definition of GTD (get to do).

some of those constraints (e.g., constraints on resources) are also applicable to the performance of γ; hence, Θ is a parameter of the Do modal operator in Clauses (1a) and (2a) as well as of the CBA and CBAG meta-predicates in Clauses (1c) and (2d).

For single-agent actions α, GTD states that G's doing of γ will leave G_c in the state of intending to do α and having a particular recipe by which it is able to do it [Clauses (1b) and (1c)]. The subscript notation on the context parameter of the Int.To in Clause (1b) indicates that G_c's intention to do α results from (i.e., is an effect of) G's doing γ. In contrast with most situations encountered in the definitions in this paper, in this case G_c does *not* intend to do α in order to do γ. The functional notation $\alpha/f(\alpha,\gamma)$ makes this clear: the reason for doing α is some functional relationship between α and γ. For instance, if Dan pays his son \$5 to chop the onions, then the context in which his son intends to chop the onions is that of being paid \$5 to do so. The constraints component of the context, constr$(C_{\alpha/f(\alpha,\gamma)})$, includes the constraints Θ.

If α is a multi-agent action, then G's doing of γ will result in the group of agents G_c having a SharedPlan to do α; they must also have a particular recipe which they as a group can use to bring α about [Clauses (2c) and (2d)]. The subscript notation on the context parameter of the SharedPlan in Clause (2c) indicates, analogously to that of the Int.To, that the SharedPlan to do α results from G's doing γ. Again, the constraints Θ are part of constr$(C_{\alpha/f(\alpha,\gamma)})$.

Both the individual plan underlying the Int.To in the individual case, and the SharedPlan in the multi-agent case may be partial. Even so, the definition of GTD may seem too strong, as it presumes both lack of will on the part of G_c and a great deal of knowledge about recipes for actions. However, in the plan definitions and other meta-predicates, GTD is only used within an embedding belief context. Hence, the claim within any of our plans is only that the agent G believes its doing of γ will leave G_c in the state of being able to perform β according to a particular recipe and either intending to do β or having a SharedPlan to do it.

A contractor's plan is not under the control of the contracting agent(s). Thus, even when the contracting is done within a full individual plan or a full shared plan, we do not require that the contractor(s) have a complete plan. Furthermore, as we describe in presenting the SharedPlan definitions (Section 6), contracting differs from having a SharedPlan in the Int.Th's which must hold.

4.5. Complex actions for planning

The terms *Select_Rec* and *Select_Rec_GR* refer respectively to the act-types for the complex planning actions that agents perform individually or collectively to identify ways to perform (domain) actions. The terms *Elaborate_Individual* and *Elaborate_Group* similarly refer respectively to the act-types for individual and group actions for extending partial plans to complete ones. To construct computer agents based on our formalization requires defining processes for selecting recipes and elaborating plans; i.e., defining specific complex actions of the type referred to by Elaborate_Individual, Elaborate_Group, Select_Rec, and Select_Rec_GR. In this paper, we use the term "Elaborate_Individual processes" to refer to computational processes (e.g., mechanisms that implement the complex actions in computer agents) that instantiate actions of the type referred to by Elaborate_Individual; we similarly append "processes" to the other act-type terms to refer to procedures implementing actions referred to by those terms. [16]

Although for each of these types of complex activity a variety of processes are possible, we restrict these terms to refer to a subset that meet certain constraints. In particular, we restrict each to incremental processes that can be interleaved with performing (domain) actions, and we require that they incorporate mechanisms for recovering from failures. [17] Some general constraints on the individual processes are given below; additional constraints are specified in Sections 5 and 6 using the terminology developed in the plan definitions.

Select_Rec$(G, \alpha, R_\alpha^p, T)$ refers to the activity of an individual agent G extending its partial recipe R_α^p for α. If the agent G has not yet begun to form a recipe for the action α, then R_α^p will be empty; in this case Select_Rec$(G, \alpha, \emptyset, T)$ refers to the initial construction of a recipe for α. Select_Rec is used in the definitions of partial plans to help represent an agent's commitment to finding a way to do the actions it intends and its beliefs that it can do so. Standard AI planning procedures can form the core of one class of Select_Rec actions. However, agents may also select recipes by retrieving them from memory, looking them up in manuals, or asking others. To avoid unnecessarily complicating the formalization, we include as part of the Select_Rec process the task of adding potential intentions for subactions in the recipe extension to the agent's set of intentions.

Elaborate_Individual$(P, G, \alpha, T_p, T_\alpha, C_\alpha)$ refers to the process of extending agent G's partial plan P at time T_p to do action α at time T_α. The major task for an Elaborate_Individual process is ensuring that the agent has a means of carrying out each of the constituent actions in the recipe for α associated with P and is committed to doing so. At any point in the planning process, for each β_i in the recipe constructed so far, an Elaborate_Individual process must initiate procedures for reconciling an intention to do β_i with currently adopted intentions. To do so, the reconciliation process must take into account resource constraints and the need to operate in a dynamic world (see for

[16] We recognize in so doing we are somewhat abusing the formal vocabulary; however, the alternative is more complex and less easily understood language.

[17] In addition to being more realistic for planning systems, incremental algorithms are crucial for dialogue models [43].

example [10,56,60]). If β_i is a basic-level action, then the elaboration process must also establish a commitment to do β_i and the belief that it can be performed. If β_i is a complex action, then the Elaborate_Individual process must ensure that a full individual plan is constructed for it. In doing so, it will initiate a Select_Rec process for a recipe for β_i [18] and an Elaborate_Individual process for β_i.

Thus, to design processes for expanding partial individual plans to more complete ones, it is possible to draw on existing AI planning mechanisms both for recipe construction and intention reconciliation. However, significant additional mechanisms are needed to design processes for the more complex multi-agent actions referred to by Select_Rec_GR and Elaborate_Group. These actions incorporate many of the constituents of Select_Rec and Elaborate_Individual, but each also includes some group decision making processes, including mechanisms for negotiating among competing recipe proposals, handling resource conflicts, and reaching consensus. Furthermore, these multi-agent planning processes require that a group have some means of forming mutual belief and agreed upon procedures for reaching consensus.

Select_Rec_GR$(\mathrm{GR}, \alpha, R^p_\alpha, T)$ refers to the activity of a group of agents extending their partial recipe R^p_α for α. Analogously to the individual case, if the group has not yet begun to form a recipe, R^p_α will be empty. The realization of this group recipe selection process is more complex than the one for an individual agent. Each agent in the group must have its own internal process for identifying recipes; this process is equivalent to Select_Rec but leads to different kinds of intentions being considered, as we discuss in Section 6.4. In addition, a group decision making procedure is needed for mediating among different agents' proposals. The agents may also need methods for constructing a new recipe using information from different group members. The interaction between recipe selection and intention adoption is also more complex, especially if no single agent is in charge. For example, Kate may have a recipe for making dinner that she believes will work only if Dan agrees to perform certain actions. Osawa and Tokoro [49] describe one possible Select_Rec_GR based on mechanisms similar to contract nets [14]. However, many collaborative planning situations exhibit less centralized management than these techniques presume.

Elaborate_Group$(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha)$ refers to the group process of extending a collaborating group's partial plan P at time T_p to do the collective action α at time T_α. The major components of this process are identifying agents able to do the constituent actions, choosing a particular agent or subgroup to do them, and ensuring that the agents adopt the requisite intentions-to and intentions-that toward these actions. In addition to having a means of assessing its own capacity to perform actions, each participant in a collaborative activity may need to assess the abilities of others. Agents also need means of communicating about their abilities. Mechanisms for reaching consensus may be invoked to decide who will be the agent of the constituent subactions in the group's recipe for α. Different groups of computer agents may vary in the ways their Elaborate_Group processes deal with selection of the agent to do an action, just as the behavior of groups of people varies.

[18] The rationale for this is given in Section 5.2.

The additional tasks to be done and the particular types of intentions to be adopted depend on whether the constituent action β_i from the recipe constructed so far is a single-agent action or a multi-agent action. If it is a single-agent action, then the choice of agent follows a process analogous to that for Elaborate_Individual, and the other group members adopt potential intentions-that this agent will be able to do the action. If it is a multi-agent action, then the Elaborate_Group process must result in the subgroup that is chosen as agent constructing a full collaborative plan. For this to occur, the subgroup must initiate a Select_Rec_GR process for a recipe for β_i [19] and an Elaborate_Group process for β_i.

The design of Elaborate_Group processes constitutes a large area of inquiry in itself. Sonenberg et al. [61] describe one set of mechanisms for group elaboration and role assignment within a formalization that includes complex actions but does not allow for partial recipes. Jennings [29] describes another mechanism, one in which a "central organizer" identifies team members, determines the recipe, and gets agreement. This approach allows for the organizer to have partial recipe knowledge only in that the organizer does not need to know how the individual team members will carry out their parts. [20] Groups comprised solely of human agents often struggle a while to reach consensus on such matters. To construct computer agents within the framework our formalization requires providing at least some built-in procedures of Elaborate_Group.

5. Individual plans

The definitions for individual plans given in this section extend in three principal ways previous mental state definitions within AI of plans of single agents. First, they accommodate more complex recipes; in particular, they accommodate the action relations and constructors defined by Balkanski [5]. Second, they introduce the possibility of contracting a constituent action to another agent. Third, they generalize to complex actions and to contracting the notion of an agent's ability to execute an action. The definition of partial individual plan further extends this work to represent an agent's mental state when its knowledge of how to do a complex action is partial, its commitment to the basic-level actions entailed in doing the complex action is partial, or it has not fully reconciled some intentions to do some subsidiary actions.

FIP, the meta-predicate for full individual plans defined in Section 5.1, represents the mental state of an agent after it has completely determined a recipe R_α for action α and has full-fledged intentions to do the actions in R_α. Thus, FIP is distinguished by the requirement that the agent knows a complete recipe for doing the action that is the objective of the plan, i.e., α; as a result, the recipe that the agent has adopted for doing α, R_α, is a parameter of the meta-predicate. Most typically an agent will not have a full plan until after it has done some of the actions in R_α; thus, most often agents

[19] The rationale for this is given in Section 6.4.

[20] Several algorithms have been proposed for negotiation and task allocation in work in distributed AI (see for example [15-17,36,38,59,62,67]); however, the appropriateness of these algorithms for a collaborative situation remains to be explored.

have only partial plans. However, the FIP definition provides a significant constraint on Elaborate_Individual processes; it specifies the conditions under which the process has completed its task. When an agent's beliefs and intentions satisfy FIP, then the agent's intention to do the planned action satisfies Clause (2a) of the definition of Int.To and there is no additional need for elaboration. PIP, the meta-predicate for partial individual plans, is defined in Section 5.2; the differences between PIP and FIP specify the information an agent needs to acquire and the intentions it needs to adopt to have a full plan. These differences provide the main driving force for an Elaborate_Individual process.

We will give the plan definitions in several stages. In each case, the definitions have the following major components of the plan: basic assumptions about recipe knowledge; the core case, covering actions the agent will do itself; the contracting case, covering actions the agent decides to contract out to others; and, for partial plans, the case dealing with unreconciled intentions. The figures included in this section provide English glosses of the major elements of the plan definitions. The full formal definitions are provided in Appendix B.

5.1. Full individual plans

The definition of a full individual plan, FIP, specifies those conditions under which an individual agent G can be said to have a plan P, at time T_p, to do action α at time T_α using recipe R_α in the context C_α. The parameter P is a permanent identifier for a plan; as partial plans are completed the other parameters may change. Hence P is needed to provide a way to refer to the evolving plan.[21]

The major constituents of FIP are given in Fig. 9.[22] As noted previously, the recipe R_α is an argument of FIP; FIP requires that the agent have a particular recipe for doing α. Clause (1) represents the agent's belief that R_α is indeed a recipe for α; the meta-language equality statement provides notation enabling the subactions, β_i, and the constraints, ρ_j, to be referred to in the remainder of the definition. Each subaction β_i in the recipe R_α will either be done by the agent itself (the "core case"), or contracted out to another agent (the "contracting case"). The core case of FIP is given in detail in Fig. 10; the contracting case is given in detail in Fig. 11. The full formal definition of FIP is given in Fig. B.1 in Appendix B.

An agent may believe it can do a particular action both by performing the action itself and by contracting it out. However, when developing a plan, the agent must commit to one of the options. Thus, when an agent has a full individual plan, either the core case applies or the contracting case applies, but not both. This property is captured by an exclusive or between FIP Clauses (2) and (3) (see Clause (2b1) in the complete definition of FIP, Fig. B.1). Because having a full individual plan for α entails intending

[21] This plan name is a parameter to Elaborate_Individual throughout the plan formation process.

[22] In this figure and those that follow we put quotation marks around "know" to indicate this use is its weak, colloquial sense with no assumption of correctness of belief; the philosophically more correct "believe" produces an incorrect English statement. We do not use scare quotes in the body of text as we presume the colloquial sense is apparent there.

FIP($P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha$)

(1) Agent G "knows" a recipe for doing α; i.e., it "knows" the subactions entailed and the constraints on them:

$$R_\alpha = \{\beta_i, \rho_j\} \wedge \mathrm{Bel}(G, R_\alpha \in Recipes(\alpha), T_p)$$

For each β_i either (2) or (3):

(2) *Core case*: Agent G intends to do the subact β_i itself.

(3) *Contracting-out case*: G intends to contract out to another agent G_c the performance of subact β_i.

Fig. 9. English description of the FIP (full individual plan) definition.

FIP($P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha$)

(2) *Core case*: Agent G intends to do the subact β_i by itself.

(a) G intends to do the subact:

$$\mathrm{Int.To}(G, \beta_i, T_p, T_{\beta_i}, C_{\beta_i/\alpha})$$

(b) If the subact is not a basic-level action, G has a full individual plan for β_i using a recipe R_{β_i}:

$$(\exists P_{\beta_i}, R_{\beta_i})\ \mathrm{FIP}(P_{\beta_i}, G, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})$$

Fig. 10. FIP: core case.

to do α by Theorem (T1), the definition of FIP does not need to include the proposition Int.To($G, \alpha, T_p, T_\alpha, C_\alpha$) within it.

The core case of a FIP for α, given in Fig. 10, requires that the agent intend to do each of the subactions β_i in the recipe for α. Each of the intentions to do a subaction in the recipe is covered either by Clause (1) (if the subaction is basic level) or by Clause (2a) (if the action is complex) of the definition of Int.To (Fig. 3). If the action β_i is complex (i.e., not basic level), then there must be some recipe R_{β_i} that G can use to do β_i and G must have a full individual plan to do β_i using that recipe. The context parameter in Clauses (2a) and (2b), $C_{\beta_i/\alpha}$, records the fact that β_i is being done as part of doing α; e.g., it includes the proposition *Contributes*(β_i, α). This Contributes component of $C_{\beta_i/\alpha}$ is used in any replanning involving β_i. The constraints component, constr($C_{\beta_i/\alpha}$), is equal to the union of constr(C_α) and $\{\rho_j\}$. As discussed in Section 5.3, G's beliefs that it will be able to perform each of the β_i, and hence α, are established by recursion in FIP in combination with the Exec clause in the definition of Int.To.

The possibility of an agent contracting out an action to another agent has not been discussed in previous work on multi-agent plans, but clearly is an option often employed by human agents.[23] The example of Kate contracting out the oil-change operation

[23] Previous work that employs the term "contracting" (see for example [14,46]) has used it to refer either to the kinds of coordination we accomplish with SharedPlans or to "helpful behavior" like that achieved by the intention-that axioms described later in the paper. "Contracting" as we use it is closer to the concept of "incentive contracting" used in the economics literature (see for example [3]).

$\text{FIP}(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$

(3) *Contracting-out case*: G intends to get another agent G_c to do the subact β_i.
(a) G believes that by doing γ it can get G_c to do the subact:

$$\text{Bel}(G, \text{GTD}(G, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(b) G intends to do the "contracting" act γ:

$$\text{Int.To}(G, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha})$$

(c) G is committed to G_c's success in doing the subact:

$$\text{Int.Th}(G, (\exists R_{\beta_i})\text{CBA}(G_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_i}, C_{cba/\beta_i/\alpha})$$

(d) If the "contracting" action is not basic level, G has a full individual plan for γ using a recipe R_γ:

$$(\exists R_\gamma, P_\gamma)\ \text{FIP}(P_\gamma, G, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta_i/\alpha})$$

Fig. 11. FIP: contracting-out case.

required for car maintenance is but one instance of such contracting. As shown in Fig. 11, to contract out an action, an agent G must believe there is some action γ that it can use to get another agent, G_c, to do β_i. Clause (3a) uses the GTD meta-predicate to represent this requirement. This meta-predicate is embedded in a belief context because the efficacy of γ in getting G_c to do β_i is a matter of belief; G could be wrong.

For contracting out, the agent must have the same intentions and abilities with respect to γ that the core case requires with respect to β_i. The agent must intend to do γ [Clause (3b)] and, if γ is not basic level, have a full individual plan to do it [Clause (3d)].

Contracting has one additional requirement. An agent who employs a contractor must have some commitment to the contractor being able to complete the job for which it was hired. Thus, G should not adopt any intentions that would conflict with G_c being able to do β_i. For example, Kate should not both intend to use her car to drive to a meeting on Monday afternoon and expect the person she hires to change the oil to do that Monday afternoon. Clause (3c) represents this commitment using the Int.Th modal operator and the CBA meta-predicate. Contracting, unlike collaborative plans, does not require reciprocity in this commitment; contracting is not in and of itself collaborative [33]. Thus, there is no correlate of Clause (3c) for the contractor G_c.[24] A different situation holds among the agents of a SharedPlan, as will be discussed in Section 6.

According to the current definition, an agent can contract out part of its individual plan only to another single agent. However, there are situations in which an agent might include multi-agent subacts in its individual plans with a presumption that it could contract out such actions to some group. For example, Dan's individual plan to sell his car might include contracting out to a group of mechanics a complete check-up of the

[24] Some mechanism, typically involving communication, is needed for G to believe that G_c will actually perform the contracted action. Legal contracts serve this purpose [33].

car. The definition of FIP does not include this case because it would complicate all parts of the definition. However, the only significant change would be to Clause (3c); this clause would need to be replaced by the following:

(3c1) $[\text{single.agent}(\beta_i) \wedge$

(3c2) $\text{Int.Th}(G, (\exists R_{\beta_i})\text{CBA}(G_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}),$
$T_p, T_{\beta_i}, C_{\beta_i/\alpha})] \otimes$

(3c3) $[\text{multi.agent}(\beta_i) \wedge$

(3c4) $\text{Int.Th}(G, (\exists R_{\beta_i})\text{CBAG}(G_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}),$
$T_p, T_{\beta_i}, C_{\beta_i/\alpha})]$

We will illustrate the FIP by showing its use in describing Dan's individual plan for making the lasagna in the meals example. According to Clause (1), Dan believes that a particular recipe, say his mother's recipe for lasagna, is a good recipe to use. This recipe provides a specification of a set of actions the doing of which under certain constraints constitutes the performance of making lasagna. For each action β_i in the recipe (e.g., making noodles, preparing sauce), he must either intend to do the action (Clause (2)) or believe that he can get someone else to do the action (Clause (3)). Suppose Dan decides that the most efficient way to make the lasagna is to get Tony to make the noodles and sell them to him, but to do the other actions himself. Then the "make noodles" subaction of the recipe will be contracted out, while all of the other actions will fall under the core case.

Dan's individual plan will include an action γ, say making a barter agreement to exchange the noodles for an evening's child care, that results in Tony's providing the noodles. For Dan's plan to be complete, Dan must believe that this action γ is either a basic-level action that he is able to do or is an action for which he knows a recipe and for which he has a full individual plan. As a consequence of Clause (3c), Dan must not knowingly do anything that would prevent Tony from making the noodles; from the axioms of helpful behavior described in Section 6.2 and given in Fig. 24, Dan must also be willing to assist in Tony's success, e.g., helping him find a place to hang the noodles to dry if necessary.

When Dan has a full individual plan, he will also have recipes for all of the actions in his mother's lasagna recipe other than "make noodles" as well as for making the barter agreement, and he will have a full individual plan for using each of these recipes to do the action for which it is a recipe.

Thus, the FIP definition for full individual plans extends previous work by treating more complex recipes (the formalization of recipes is more general than that in the original formulation [23]), providing an expanded notion of what it means to be able to carry out a complex action, and allowing for contracting to another agent. When an agent's mental state satisfies the FIP definition, the agent knows a complete recipe tree and is fully committed to all of the basic-level actions in it. Thus it satisfies the most stringent correlate of the conditions for intending to do a basic-level action. However, as discussed in Section 4.3.1, this constraint is too strong in general; partial individual plans, which we describe next, are essential to providing the weaker constraint in the definition of Int.To.

PIP($P, G, \alpha, T_p, T_\alpha, C_\alpha$)

(1) Agent G believes that there is a way to perform α; its recipe for doing α may be partial, i.e., it may "know" only *some* of the subactions that need to be performed; it intends to complete its partial recipe.

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

(2) *Core case*: Agent G intends to do the subaction by itself, but may have only a partial plan for doing it.
(3) *Contracting-out case*: G intends to contract out to another agent G_c the performance of the subaction, but may not have a full plan for doing the "contracting" action.
(4) *Unreconciled case*: G has not yet reconciled the intention to do the subaction; i.e., it has only a potential intention to do it.

Fig. 12. English description of the PIP (partial individual plan) definition.

5.2. Partial individual plans

When an agent adopts an intention to do a complex action, its knowledge of how to do that action may be partial. Furthermore, its commitment to the basic-level actions entailed may be partial; it may not have fully reconciled some of these subactions and it cannot intend any basic-level actions it does not yet know about. The definition in this section treats both partiality of knowledge and partiality of intention. Although we are considering partial individual plans in the context of an agent intending to do an action, it is possible for an agent to have a partial plan to do some action without having yet formed an intention to do that action. For example, Dan may know a recipe for making lasagna and have potential intentions to do the actions in this recipe, but may not yet have committed himself to fully elaborating a partial plan using this recipe. The PIP definition thus accommodates a lower level of commitment than intending to do an action. By using PIP, the formalization is able to represent mental state at an important stage in the planning process without having to overly weaken the notion of intending to do an action.

The PIP definition provides for individual plans to be partial in four ways. First, the agent may have only a partial recipe for the plan's action; this partiality is represented by the difference between Clause (1) in the definition of PIP in Fig. 12 and Clause (1) in the FIP definition (Fig. 9). Second, the agent may have only partial plans for some of the β_i in the recipe for α; this partiality is represented by the difference between Clause (2) in the PIP definition and the corresponding clause in the FIP definition. Third, there may be some subactions in the recipe for which the agent has only potential intentions. In particular, when an agent selects a recipe for α, it directly adopts Pot.Int.To do the actions β_i in that recipe; these potential intentions must be reconciled with other intentions and partial plans constructed for them. This type of partiality underlies the additional clause, Clause (4), in the PIP definition. Finally, there may be partiality in the contracting out case [Clause (3)]; the agent may have only a partial plan for its contracting action.

The bicycle kit example may be used to illustrate all four types of partiality. Kate's initial partial plan to assemble her bicycle from a kit might include the belief that the accompanying instructions are complete, that she can read the instructions, and

PIP($P, G, \alpha, T_p, T_\alpha, C_\alpha$)

(1) Agent G believes that there is a recipe for performing α:

$$\mathrm{Bel}(G, (\exists R_\alpha)[R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha], T_p)$$

(a) If G has only a partial recipe,

$$[\mathrm{Bel}(G, (\exists R_\alpha)[R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subset R_\alpha], T_p) \Rightarrow]$$

(b) there is a recipe $R_{select.rec}$ for determining an appropriate complete recipe for α such that,
(1) G intends to determine (i.e., select or find) an appropriate recipe for α:

$$\mathrm{Int.To}(G, \mathrm{Select\_Rec}(G, \alpha, \{\beta_i, \rho_j\}, T_{select.rec}), T_p, T_{select.rec}, C_{select.rec/\alpha})$$

(2) G has a full plan using $R_{select.rec}$ to find an appropriate recipe for α:

$$\mathrm{FIP}(P_{select.rec}, G, \mathrm{Select\_Rec}(G, \alpha, \{\beta_i, \rho_j\}, T_{select.rec}), T_p, T_{select.rec}, R_{select.rec}, C_{select.rec/\alpha})]]$$

where every subaction in the recipe R_α, G selects for α, is one that G believes it can either perform or contract:

$$\begin{array}{l} \mathrm{constr}(C_{select.rec/\alpha}) \supseteq \\ \quad (\exists R_\alpha = \{\delta_v, \kappa_e\}) \\ \quad [\mathrm{Bel}(G, R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha, T_{select.rec}) \wedge \\ \quad\; (\forall \delta_v \in R_\alpha)\mathrm{Bel}(G, (\exists T_{\delta_v})[\\ \qquad\qquad (\exists R_{\delta_v})\mathrm{CBA}(G, \delta_v, T_{\delta_v}, R_{\delta_v}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\}) \vee \\ \qquad\qquad \mathrm{CC}(G, \delta_v, T_{\delta_v}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\})], T_{select.rec})] \end{array}$$

Fig. 13. PIP: finding a recipe.

that she believes she can perform or contract out each of the subactions described in the instructions at the requisite time. While reading the instructions, Kate will adopt potential intentions to do the subactions. As she reconciles these potential intentions with other commitments she has she will determine ways of doing each subaction or of contracting it out. For example, she might decide to attach the front fork and the wheels to the frame herself, but to pay Dan to attach and lubricate the chain. If so, her partial plan will expand to include her having an intention *to* pay Dan and intentions *that* he will be able to do the attachment and lubrication actions as well as a set of *potential* intentions *to* do the subactions involved in assembling the frame and attaching the wheels.

We will examine each of these sources of partiality in turn and discuss the requirements they impose on an Elaborate_Individual process. Fig. 13 contains the portion of the definition of PIP that deals with the case of the recipe being partial. In this case, the minimal requirements for the agent having a partial individual plan are that the agent believes there is a recipe for α and that it has a complete individual plan for determining that recipe. The PIP, unlike the FIP, does not have the recipe R_α as a parameter; instead the existence of the recipe is asserted within a belief context in Clause (1). The set of constituent acts and constraints that the agent knows, $\{\beta_i, \rho_j\}$, may be only a subset of those in the full recipe [Clause (1a)]; the set may even be empty.

In this case, the PIP definition requires that three conditions hold. First, the agent must intend to identify a recipe for α [Clause (1b1)]. As we discussed in Section 4.5, the action Select_Rec that appears in this definition takes a partial recipe $\{\beta_i, \rho_j\}$ for the action α and extends this partial recipe to a complete one, $\{\delta_v, \kappa_e\}$. It also adds potential intentions for all the new subactions to the agent's set of intentions. The Select_Rec process may be general. It may invoke a plan formation process to construct a new recipe, or select among existing recipes in the agent's recipe library, or it may give a way of finding a recipe (e.g., looking up how to do the action in a manual; knowing someone to ask); processes that combine several of these options with a decision among the results of each are also instances of Select_Rec.

Second, the agent's intention to determine the recipe must have more than a partial plan associated with it; the agent must have a full plan (FIP) for determining the recipe [Clause (1b2)]. This requirement is less strong than it may first appear because Select_Rec itself can be general. This clause embodies a claim that an agent does not have a partial plan to do an action unless it knows some way of finding out how to do the action and is committed to finding out. This constraint derives from the role PIP plays in the definition of Int.To; it represents a commitment to means-ends reasoning about the intended act. Requiring an agent to have this minimal commitment to means-ends reasoning also constitutes a reasonable constraint on what it means to have a plan to do an act.

In addition, the agent must believe that the recipe (or recipe extension) it finds or selects is an appropriate one, namely a recipe (or extension) comprising constituent subactions that the agent believes it either will be able to perform under the constraints in the recipe or will be able to contract out successfully. If a partial recipe R^p_α is being extended, then the actions and constraints in the new recipe must be a superset of those in R^p_α; i.e., this must be an extension of the original partial recipe. These additional constraints are encoded in the constraints component of $C_{select.rec/\alpha}$; a formal specification of this constituent of constr($C_{select.rec/\alpha}$) is noted at the end of the figure and in the complete formal PIP definition in Appendix B.

As was discussed in Section 4.5, as a result of the Select_Rec action, the agent G will have at least a Pot.Int.To do each constituent subaction in the recipe that it has for α.[25] The plan definition distinguishes those subactions for which the agent has adopted full-fledged intentions from those that still need to be reconciled. The former subactions fall into the "core case" (Fig. 14) or the "contracting-out case" (Fig. 15) and are denoted as β_r. The subactions corresponding to unreconciled intentions (Fig. 16) are denoted as β_k.

For each action β_r in the PIP core case (Fig. 14), the agent G has an intention to do β_r [Clause (2)]. This requirement of the partial individual plan definition resembles the one for the full individual plan. However, in the PIP, the plan associated with this Int.To may be only partial. Hence, there is no correlate in the PIP definition of the FIP in Clause (2b) in Fig. 10. In addition, the recursion of partial plans in this case imposes a much weaker requirement on the agent's beliefs about its abilities to do actions in a recipe for β_r than that imposed in the FIP; this difference is discussed in Section 5.3.

[25] A Select_Rec process might operate incrementally. If it does, then Pot.Int.To's will be formed only for a subset of the subactions at any one time.

PIP($P, G, \alpha, T_p, T_\alpha, C_\alpha$)

(2) *Core case*: Agent G intends to do the subaction β_r by itself, but may not have a full plan for doing β_r:

$$\text{Int.To}(G, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha})$$

Fig. 14. PIP: core case.

PIP($P, G, \alpha, T_p, T_\alpha, C_\alpha$)

(3) *Contracting-out case*: G intends to contract out to another agent G_c the performance of the subact β_r, but G may not have full plan for the contracting action γ.

(a) G believes that by doing γ it can get G_c to do the subact:

$$\text{Bel}(G, \text{GTD}(G, \gamma, T_\gamma, G_c, \beta_r, T_{\beta_r}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(b) G intends to do the "contracting" act γ but may not have a full plan for doing γ:

$$\text{Int.To}(G, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha})$$

(c) G is committed to G_c's success in doing the subact:

$$\text{Int.Th}(G, (\exists R_{\beta_r})\text{CBA}(G_c, \beta_r, R_{\beta_r}, T_{\beta_r}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_r}, C_{cba/\beta_r/\alpha})$$

Fig. 15. PIP: contracting-out case.

To remove the partiality of the core case requires that the agent identify recipes for each non-basic-level β_r and form full plans for doing the actions they require. The formation of the full plan will entail determining that the agent is able to perform the constituent subactions in the recipe (or contract them out), in particular, that it can execute all the basic-level actions. Clause (2b) of the definition of Int.To (Fig. 3), and in particular the agent's performance of the Elaborate_Individual action, is the locus of the actions an agent must take to achieve these completions.

The difference between the contracting-out cases of the partial and full individual plans is similar to that of the core cases. The portion of PIP detailed in Fig. 15 differs from the FIP version in Fig. 11 only in the lack of a correlate to Clause (3d). In the PIP case, the agent may have only a partial plan to do the contracting action γ. This type of partiality is resolved analogously to that for the core case; the only difference is that the agent is dealing with the contracting action rather than a subaction of the recipe for α.

The final way in which a plan may be partial is for the agent to have unreconciled potential intentions about some of the subactions β_k. As detailed in the component of the PIP definition given in Fig. 16, the agent may consider both doing the action itself and contracting it out. The agent's consideration of doing β_k itself has two components: the agent must have a potential intention to do the subaction β_k [Clause (4a1)] and believe that some recipe exists by which it will be able to perform β_k [Clause (4a2)]. Clause (4a2) makes only the weakest form of claim on the agent's ability to act; the

$\text{PIP}(P, G, \alpha, T_p, T_\alpha, C_\alpha)$

(4) *Unreconciled case*: Agent G has not reconciled the intention to do the subaction β_k.
  (a) *Core case*: G considers doing the subact β_k by itself:
    (1) G has a potential intention to do the subact:

$$\text{Pot.Int.To}(G, \beta_k, T_p, T_{\beta_k}, C_{\beta_k/\alpha})$$

    (2) G believes that there is a recipe which it can use to perform the subact:

$$\text{Bel}(G, (\exists R_{\beta_k})\text{CBA}(G, \beta_k, R_{\beta_k}, T_{\beta_k}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

  (b) *Contracting case*: G considers getting another agent G_c to do the subact β_k.
    (1) G has a potential intention to do a contracting action γ:

$$\text{Pot.Int.To}(G, \gamma, T_p, T_\gamma, C_{\beta_k/f(\beta_k,\gamma)/\alpha})$$

    (2) G believes that there is a recipe which it can use to perform the contracting action γ and by doing γ it can get G_c to do the subact:

(a) $\text{Bel}(G, (\exists R_\gamma)\text{CBA}(G, \gamma, R_\gamma, T_\gamma, \text{constr}(C_\alpha) \cup \{\rho_j\}) \wedge$
(b) $\text{GTD}(G, \gamma, T_\gamma, G_c, \beta_k, T_{\beta_k}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$

Fig. 16. PIP: unreconciled case.

agent G must believe there is some recipe R_{β_k} for β_k, but G may not yet have figured out how to determine that recipe. Furthermore, G's belief about its ability to carry out the subactions of the recipe are necessarily weak, because G does not yet know what the actions are. The embedding of CBA in a belief context represents this weak belief.

The agent's consideration of contracting out β_k has two similar components: the agent must have a potential intention to perform a contracting action γ [Clause (4b1)] and a belief that it can perform γ [Clause (4b2a)]. In addition, the agent must believe that by performing γ it will get some other agent to do β_k [Clause (4b2b)].

To remove this last kind of partiality, for each action β_k in Clause (4) the agent must move the β_k either to the core case or to the contracting-out case. To move the β_k to the core case, the agent must turn its potential intention to do β_k [Clause (4a1)] into a full-fledged intention to do this action that has an associated partial individual plan and a FIP to perform the action of Elaborate_Individual on this PIP. To move the β_k to the contracting-out case, the agent must turn its potential intention to do a contracting action [Clause (4b1)] into a full-fledged intention to do the contracting action γ that has an associated partial individual plan; its beliefs about its ability to do the action and thereby get another agent to do β_k [Clause (4b2)] play a role in this transition. In addition, the agent must adopt an intention-that a particular contractor be able to do β_k.

The process of transforming the potential intentions-to into full-fledged intentions-to for the core and contracting cases is similar, only the target action differs (β_k itself or γ); we will describe only the transformation to the core case. First, the agent must reconcile the Pot.Int.To do β_k with all other intentions (both Int.To's and Int.Th's) it currently has. If β_k is a basic-level action, then G also needs to establish that it can execute β_k, and adopt a commitment to do so. If β_k is not basic level, then G must select a recipe $R_{select.rec_{\beta_k}}$ for determining a recipe for β_k and develop a full individual plan for determining a recipe for β_k using $R_{select.rec_{\beta_k}}$. The agent's having a full plan for selecting a recipe will satisfy the minimal constraints for the agent to have a partial individual plan P_{β_k} to do β_k. In addition, to satisfy the minimal conditions for having an intention to do β_k, the agent must form a full individual plan to elaborate (using its Elaborate_Individual process) the partial plan P_{β_k}. The contracting case requires additional deliberation to transform the potential intention that the contractor be able to perform β_k into a full-fledged intention; this intention must also be reconciled with all other intentions the agent has.

If the Elaborate_Individual process succeeds in these transformations, it removes the Pot.Int.To for β_k from its set of potential intentions. However, it may fail in several ways. For example, it may not be able to figure out a way to obtain a recipe for doing β_k. Alternatively, the reconciliation process may result in the agent deciding it cannot now adopt an intention to do β_k. In either case, the partial individual plan might "regress" and become more partial while the agent searches for a recipe for α that does not require β_k.

The partial individual plan might also regress if, in the process of reconciling intentions, the agent decides to drop an intention-to for one of the core case subactions or a contracting action. In these cases, the agent will once again have a potential intention to do the relevant subaction, i.e., what was a β_r might again become a β_k.

The PIP definition extends previous work by treating situations in which agents have incomplete knowledge about how to do a complex action. The minimal constraints on having a partial plan to do action α are such that the agent does not initially need even a partial recipe for α; it just needs some idea of how to get a recipe. By accommodating this lower level of knowledge, the formalization is able to cover an important class of planning situations (e.g., the one that arises in the construction kit example) that have not been handled by previous formalizations. The definition also treats partiality in intention adoption, including the state in which an agent has not yet decided whether to do an action or to contract it. And, partiality is allowed recursively in the plans for constituent actions. The analysis and formalization of what it means for agents to have partial plans revealed several interesting new issues, including (1) determining a minimal level of knowledge required about how to do an action (to rule out cases of agents planning to do actions for which they have no possibility of overcoming insufficient knowledge); (2) specifying a minimal level of knowledge and commitment for an agent to intend to do an action before it has complete information about how to perform the action (i.e., defining Int.To for partial plans); (3) identifying core recipe determination and intention reconciliation processes for extending a partial plan to a more complete one (i.e., specifying Elaborate_Individual) and specifying conditions under which they have completed their tasks (when FIP clauses hold); and (4) specifying what agents need to ascertain about their ability to perform a complex action given incomplete information about how to do the action (to allow agents to have a plan before they can completely establish capabilities). The PIP definition has clauses that directly refer to the first three issues. We address the ways in which the fourth issue is handled in the next subsection.

5.3. Capabilities to perform actions in individual plans

For an agent to intend to do some action, it must believe that it is capable of doing the action (see caveat in Section 4.3.1). If the action is basic level, the definition of Int.To requires explicitly that the agent believe it can execute the intended action. If the intended action is complex, the requirements on an agent's beliefs about its capabilities to perform the action depends on whether the agent's plan is complete or partial. The definitions of FIP and PIP implicitly encode the requisite beliefs about capability. However, both to understand the kinds of agent behavior these plan definitions engender and to guide agent design, it is useful to specify this ability knowledge separately. In this section, we briefly describe the constraints the plan definitions place on ability knowledge; formal definitions and theorems establishing these constraints are given in Appendix A.

For an agent to have a complete individual plan (satisfying FIP), it must know a complete recipe for the action to be performed; that is, it must have determined recipes for the complete extended recipe tree to all levels of detail for doing the action (e.g., the full tree for the example in Fig. 2). From the FIP definition, by recursion, it must intend to do all the subsidiary actions except those covered by contracting and for contracting it must intend to do the contracting action. As a result, from the Int.To definition, the agent must believe that it can execute all of the basic-level actions in the extended recipe tree for α.

This level of capability knowledge is less strong than that CBA represents because the agent's beliefs about the recipes may be in error; the agent may believe R_α is a recipe for α when it is not. However, the requirement is stronger than belief in CBA, i.e., than $\mathrm{Bel}(G, (\exists R_\alpha)\mathrm{CBA}(G, \alpha, \ldots))$. A mental state intermediate between CBA embedded in Bel and CBA unembedded is needed. Because recipes may include complex subactions to arbitrary levels, it will not suffice simply to pull the quantifier outside the embedding belief context at a single level of description.

The meta-predicate BCBA ("believe can bring about") defined in Appendix A, Fig. A.1, represents the level of belief in ability to perform an action required for an agent to have a complete individual plan. As is the case for FIP, the recipe for α, R_α, is an argument of BCBA, reflecting the fact that a particular set of constituent subactions and constraints is known to the agent and not just the existence of some recipe. Only the belief that R_α is a recipe for α is part of the definition of the BCBA; the existential binding for R_α is outside the scope of the definition. BCBA appears recursively within its definition with the constituent acts β_i of α as arguments and the recipe R_{β_i} stipulated outside any belief context.

This same level of recipe knowledge is represented implicitly in the definition of FIP, through interaction with the definition of Int.To. For basic-level actions, Int.To yields BCBA Clause (1). For complex subactions, Clause (2a) in BCBA is just Clause (1) of FIP; if G plans to perform the action itself, FIP Clauses (2a) and (2b) combine recursively with Int.To to yield BCBA Clause (2b1); for subactions G contracts out, FIP Clauses (3b) and (3d) similarly yield BCBA Clause (2b2), and FIP Clause (3a) gives BCBA Clause (2b2b). Thus, as stated in Fig. A.1, this level of recipe knowledge is entailed by the FIP definition. A formal proof is given in Appendix A.1.

In contrast, when an agent has a partial plan for α, its beliefs about its capabilities may be quite limited, because its knowledge of the recipe it will use is incomplete. The agent may only believe that there is some way to find a recipe that it can use to perform α. Until it knows the constituent actions in the recipe for α, the agent cannot make any determination about its abilities to perform these subactions. While the lack of a recipe makes the agent's knowledge in this situation weaker than in the FIP, the agent's beliefs about its ability must be stronger than the CBA embedded in Bel yields. In particular, the agent must believe that it can determine a complete recipe and will be able to perform, or to contract out, each of the actions δ_v in the complete recipe once it is determined. As the agent identifies pieces of the recipe, it must also establish its ability to perform the actions in that piece or to contract them.

The meta-predicate WBCBA ("weakly believe can bring about") defined in Appendix A, Fig. A.2, represents the level of belief an agent G must have about its abilities to select an appropriate recipe and perform or contract out each of the constituent subactions in this recipe. This same level of belief is represented implicitly in the definition of PIP, as stated in Theorem (T3). Clause (1) of WBCBA is established by PIP Clause (1); the condition in Clause (1a) of WBCBA is similar to that of Clause (1a) of PIP. Clause (1b) of WBCBA follows from the FIP in PIP Clause (1b2) and Theorem (T2); as shown in Fig. 13, the constraints component of the FIP context, constr($C_{select.rec/\alpha}$), contains the constraints in Clauses (1b1a)–(1b1e) of WBCBA. If G intends to do the subaction β_i by itself and β_i is a basic-level action then Clause (1) of Int.To (Fig. 3) and Clause (0) of WBCBA used recursively yield Clause (2) of WBCBA. If G intends to do a complex subaction β_i by itself, then Theorem (T3) applied recursively to Clause (2b1) of Int.To yields Clause (2) of WBCBA. For subactions G contracts out, Clauses (3a) and (3b) of PIP similarly yield WBCBA Clause (3a) and Clause (3b). For the unreconciled subactions, Clause (4a) of WBCBA is established by Clause (4a2) of PIP and Clause (4b) is established by Clause (4b2) of PIP.

6. SharedPlans and intending-that

The definitions in this section, utilizing those given in the preceding sections, provide a model of collaborative behavior that has several distinguishing properties, all of which are maintained under conditions of partial knowledge. The key properties of the model are as follows:

(i) it uses individual intentions to establish commitment of collaborators to their joint activity;
(ii) it establishes an agent's commitments to its collaborating partners' abilities to carry out their individual actions that contribute to the joint activity;
(iii) it accounts for helpful behavior in the context of collaborative activity;
(iv) it covers contracting actions and distinguishes contracting from collaboration;
(v) the need for agents to communicate is derivative, not stipulated, and follows from the general commitment to the group activity;
(vi) the meshing of subplans is ensured; it is also derivative from more general constraints.

The attitude of intending-that plays a significant role in establishing several of these properties. It is the basis for agents to avoid adopting intentions that conflict with those that arise from the group's plan [needed for Properties (i) and (ii)], and it engenders helpful behavior [Property (iii)]. The way in which Int.Th is used in SharedPlans captures the difference between agents having a SharedPlan and one agent contracting to another agent [Property (iv)]. When agents have intentions-that, they are required to provide information about their progress to each other in certain circumstances, leading to communication [Property (v)]. Together with mutual belief, intentions-that contribute to meshing subplans [Property (vi)]. We discuss each of these roles in Section 7.

The belief and intention operators are used in different ways in the SharedPlan definitions. Mutual belief requires infinite nestings of individual beliefs, but utilizes only a single belief operator, Bel. In contrast, to handle the intentions that arise in SharedPlans, we need two operators—Int.To and Int.Th—but there is no need for infinite embeddings of these operators either in themselves or within one another. However, both operators may be embedded within the mutual belief operator, MB.

Two important properties of collaborating agents' beliefs and intentions are captured in the definitions that follow. First, an agent only has intentions-to toward acts for which it is the agent; intentions-that represent its responsibilities with respect to the actions of other agents. Second, agents do not need to know complete recipes for those actions that they are not personally committed to doing [64]. In the meals example, Kate and Dan need to establish mutual belief of a recipe for making dinner, namely that it will comprise Kate's making the appetizer, Dan the main course, and the two of them together making the dessert. Only Kate needs to know the recipe for the appetizer; but Dan and Kate must have mutual belief that Kate has such a recipe and can carry it out. The analogous case holds for Dan and a recipe for the main course. In contrast, Dan and Kate need mutual belief of the recipe to be used for making dessert.

The SharedPlan definitions stipulate only minimal constraints on what agents need to know about the recipes for actions to be done by other agents. As a result, it is possible that an agent constructed according to the SharedPlan specifications will not recognize some conflicts between its intentions-that its collaborators succeed and its other intentions. In particular, if an agent does not know particulars of a recipe, it may not know about a conflict, and thus, Axiom (A1) (Fig. 4) does not apply. Resource conflicts present an obvious case of this problem. If Kate does not know that Dan's lasagna recipe calls for using mushrooms, she will not detect the conflict between an intention to make mushroom puffs using all the mushrooms currently on hand and an intention that Dan be able to make the lasagna.

Collaborative agents could only be sure they could detect all conflicts if either (a) they could compute all the possible ways that other agents might do their actions and all the resources they would use and thus all the conflicts that might arise; or, (b) they continuously communicated full information about their plans. Possibility (a) not only has computational problems, but would lead to so many alternatives that avoiding conflict with all of them would significantly limit options. For every group member to be told about the full details of the recipes being used by other agents and subgroups, as suggested by approach (b), would require an enormous amount of communication. Thus, neither of these alternatives seems practical.

Several mechanisms have been developed for conflict detection and resolution in the context of cooperation of autonomous agents (see for example [31,39,50]) and for global information management using local autonomous agents (see for example [26]). Other research has addressed this problem in the context of task allocation among autonomous agents under incomplete information [47]. Each of these approaches requires that different specific information be communicated when less than the full information can be. Thus, a range of options are possible all of which provide reasonable, though different, support for collaborative activity. The SharedPlan definitions stipulate only minimal constraints on shared knowledge of recipes; they provide a framework in which designers may implement different strategies depending on the specifics of the collaborative activity and the environment. We conjecture that the determination of an appropriate strategy is, in part, domain dependent; for example, the recipes for a domain might need to specify the resources that could be in contention. Furthermore, agent design will vary depending on the level of risk of failure from unforeseen conflicts that designers are willing to incur; the more costly such failures are, the more designers will err on the side of encoding additional constraints on recipes and on the elaboration processes so that agents have sufficient knowledge to avoid intention conflict.

6.1. Definition of SharedPlan

The SharedPlan meta-predicate, SP, representing that a group of agents GR has a collaborative plan to perform together some action α, is defined recursively in terms of full and partial SharedPlans. A *full SharedPlan*, FSP, is the collaborative correlate of a full individual plan and includes full individual plans among its constituents. A *partial SharedPlan*, PSP, is the collaborative correlate of a partial individual plan. A principal way in which SharedPlans and individual plans differ is that knowledge about how to act, ability to act and commitment to act are distributed in SharedPlans. Even when a group's plan is complete, there may be no one individual who knows the complete recipe tree; no single agent needs to be able to perform all the basic-level actions the collaborative action comprises; and the requisite intentions to act are distributed among group members. The group has a SharedPlan, but no individual member alone has the SharedPlan.

The challenge in defining SharedPlans is to provide for this distributed knowledge and commitment to act, while ensuring that the group members have adequate knowledge about each other's capabilities and sufficient commitment to their joint activity. In particular, the group analogue of the Exec and Commit constraints in the basic.level action component of the definition of Int.To must not only treat complex actions and partial knowledge, but also accommodate the distributed character of group activity. In addition, the establishment of certain mutual beliefs plays a central coordinating role in the SharedPlan definitions. Communication among agents is essential to establishing the requisite mutual beliefs.

As shown in Fig. 17, a group of agents GR has a SharedPlan P at time T_p to do α at time T_α in the context C_α just in case either (1) they have a full SharedPlan for doing α, or (2) they have a partial SharedPlan to do α, and a full SharedPlan to complete that partial plan. The meta-predicates, SP, PSP, and FSP enable representation of the mental

$\mathrm{SP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha)$

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

Fig. 17. Definition of SP (SharedPlan).

states of agents in a collaborating group throughout the planning process, from inception of a partial SharedPlan through to completion (and execution) of the full SharedPlan.

SP plays a role for plans of groups of agents analogous to the one played by Int.To for individual plans, but there are several important differences. Group members may have different reasons for engaging in the collaborative activity of doing α, so C_α may vary across group members. For example, hunger might underlie Kate's making dinner with Dan, whereas a desire for social interaction underlies Dan's making dinner with Kate (cf. [9]). In addition, because the beliefs and intentions about the plan are distributed, each of the agents in GR will have its own internal name for the plan; P refers to an agent internal name. Thus, the distributed property of SharedPlans yields an additional constraint on agent design. To engage in an Elaborate_Group process, agents must have a means of referring to their collaborative plan in their communication; i.e., they must be able to form an externally useful reference to P.

The most significant difference between SP and Int.To, however, is that SP is a meta-predicate not a modal operator. There is no attitude of "we-intending" [58] or joint-intention [40]. As a consequence, the definition of SP has one less clause than that for Int.To. Whereas the definition of Int.To separately asserts that the agent intends to do the elaboration (Clause (2b3) in Fig. 3), there is no separate clause in the SP definition asserting SP of the Elaborate_Group. Although the Int.To is entailed by the definition of FIP, including this clause in the definition of Int.To makes explicit the assertion of an additional agent attitude and thus may be useful for agent design. The SP is likewise entailed by the definition of FSP. In this case, however, there is no additional agent attitude to assert separately, because SP is a meta-predicate not a modal operator. Furthermore, there are no axioms constraining the SP meta-predicate that are analogous to those for the modal operators Int.To and Int.Th.

The intention-based constraints on agent design that are imposed by having SharedPlans are derived from those entailed by the individual intentions that are part of the agent's SharedPlans, including any subsidiary individual plans. Both FSP and PSP entail individual intentions to do actions, including actions of elaborating or extending partial plans. As others have argued [9, 12, 58], individual intentions to act and mutual belief of such intentions are not sufficient for representing the mental state of the participants in collaborative activities. To satisfy the additional requirements of collaboration, the SharedPlan definitions include various Int.Th clauses.

The definitions of FSP and PSP, like those for individual plan, have four components—basic assumptions about recipe knowledge; the core case; the contracting case; and, for

FSP(P, GR, α, T_p, T_α, R_α, C_α)

(0) The group GR has mutual belief that all members of the group are committed to the success of the group's doing α:

$$\text{MB}(\text{GR}, (\forall G_j \in \text{GR})\,\text{Int.Th}(G_j, \text{Do}(\text{GR}, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_p, T_\alpha, C_\alpha), T_p)$$

(1) The group GR has mutual belief of the acts (β_i) that they need to perform to accomplish α and the constraints (ρ_j) on them:

$$R_\alpha = \{\beta_i, \rho_j\} \wedge \text{MB}(\text{GR}, R_\alpha \in \mathit{Recipes}(\alpha), T_p)$$

For each β_i either (2) or (3):

(2) *Core case*:
  (a) *Subaction β_i is a single-agent action*: Some member of the group will do the subaction.
  (b) *Subaction β_i is a multi-agent action*: Some subgroup will do the subaction.
(3) *Contracting case*:
  (a) *Subaction β_i is a single-agent action*: The group will get another agent, G_c, to do the act.
  (b) *Subaction β_i is a multi-agent action*: The group will get another group of agents to do the subaction.

Fig. 18. English description of the FSP (full SharedPlan) definition.

partial plans, the unreconciled intentions case. However, for SharedPlans the core, contracting and unreconciled cases subdivide depending on whether the subaction to be done is single-agent or multi-agent.[26] Our discussion of the plan definitions will focus on differences between SharedPlans and their individual counterparts. Again, figures in this section provide English glosses of the major elements of the plan definitions while the full formal definitions are provided in Appendix B.

6.2. Full SharedPlans

The meta-predicate FSP is used to represent the situation in which a group of agents has completely determined the recipe by which they are going to do some group activity, and members of the group have adopted intentions-to toward all of the basic-level actions in the recipe as well as intentions-that toward the actions of the group and its other members. Most typically a group of agents will not have such a complete plan until after they have done some of the actions in the recipe. Groups, like the individual agents they comprise, typically have only partial plans. Analogously to FIP, the FSP definition specifies the conditions under which the Elaborate_Group process has completed its task. Differences between FSP and PSP specify the information agents need to acquire, individually and mutually, and the intentions they need to adopt to have a complete collaborative plan.

The definition of the meta-predicate FSP specifies when the group GR has a complete plan P at time T_p to do action α at time T_α using recipe R_α in context C_α. Fig. 18

[26] The recipe case applies to the action α which is the objective of a SharedPlan and hence is necessarily multi-agent.

gives the major constituents of the definition of the meta-predicate FSP; the full formal definition appears in Appendix B. The plan definition uses P to identify the plan.

Two key characteristics of collaboration derive from Int.Th in Clause (0): (1) agents avoiding the adoption of intentions that conflict with the joint activity; and, (2) agents adopting intentions to communicate about the plan and its execution. The intention-that in Clause (0) explicitly represents each group member's commitment to the group's performance of α.[27] Through Axiom (A1) this Int.Th directly contributes to (1). Some of the ways in which it leads to communication actions are described in Section 7.

When agents have a complete collaborative plan to do an action α, they must have a complete recipe for that action. However, the recipe knowledge is typically distributed. Clause (1) ensures that the agents agree on the recipe they will use to perform α. The distributed knowledge about how to perform the constituent actions β_i is handled in the core and contracting cases.

For each constituent action β_i in the core case, the FSP definition requires that some agent or subgroup is committed to doing β_i and is able to carry it out, and that the full group has knowledge of the agent's or subgroup's commitment and capability. In addition, to ensure that subplans are compatible or "mesh" in Bratman's terminology [9] and provide sufficiently for helpful behavior, the definition requires that the full group form a commitment to the ability of the agent or subgroup to carry out β_i.

The single-agent portion of the core case of FSP, shown in Fig. 19, and the multi-agent portion, given in Fig. 20, address these constraints in analogous ways. The two principal distinctions between these portions of FSP and the core case of FIP are (1) the representation of the commitment of every member of the group to the ability of an agent or subgroup to carry out a constituent action [Clauses (2a4) and (2b3)]; and (2) the distinction between what the agent(s) who are doing an action must know about the recipe and their ability to act [Clauses (2a3a) and (2b1)] and the information that other group members require [Clauses (2a3b) and (2b2)], represented by the difference in quantifier scoping between the relevant clause pairs. We present the single-agent case first, contrasting it where relevant with individual plans, and giving rationale for these distinctions. We then discuss the additional issues raised by the multi-agent case.

Clauses (2a1) and (2a4) specify the commitments group members must have toward the performance of β_i and the mutual beliefs they must have of these commitments. Clause (2a1a) establishes G_k's intention to do β_i, and Clause (2a1b) requires group mutual belief that G_k has this intention. Clause (2a4) represents commitment on the part of all other agents in the group to ensuring that G_k is able to perform β_i. That each G_j actually has the appropriate intention-that can be inferred from Axiom (A3) in Fig. 5. Because G_k has an intention-to toward β_i, there is no need for G_k to have an intention-that to provide for avoiding conflicting intentions; other entailments of intending-that, e.g., helpful behavior, that are not also entailments of intending-to, do not apply for G_k.

[27] Because agents are assumed actually to hold any intentions-that that they believe they hold (Axiom (A3) in Fig. 5), this clause establishes not only mutual belief in the intention, but also that the agents hold the intention.

FSP(P, GR, $\alpha, T_p, T_\alpha, R_\alpha, C_\alpha$)

(2) *Core case*:
  (a) *Subaction β_i is a single-agent action*:
    A member of the group, G_k, will do the subaction.
    (1) G_k's intentions and the group's related beliefs:
      (a) G_k intends to do β_i:

$$\text{Int.To}(G_k, \beta_i, T_p, T_{\beta_i}, C_{\beta_i/\alpha})$$

      (b) GR mutually believe that the agent G_k intends to do the act:

$$\text{MB}(\text{GR}, \text{Int.To}(G_k, \beta_i, T_p, T_{\beta_i}, C_{\beta_i/\alpha}), T_p)$$

    (2) The subaction β_i is a basic-level action:
      (a) The group mutually believe that G_k is able to perform the subaction:

$$\text{MB}(\text{GR}, \text{CBA}(G_k, \beta_i, R_{Empty}, T_{b_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

    (3) The subaction β_i is not a basic-level action:
      (a) There is a recipe R_{β_i} for subaction β_i such that,
      G_k has a full individual plan for the subact that uses the recipe R_{β_i}:

$$(\exists R_{\beta_i}, P_{\beta_i})\text{FIP}(P_{\beta_i}, G_k, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})$$

      (b) The group mutually believe that there is a recipe R_{β_i} such that,

$$\text{MB}(\text{GR}, (\exists R_{\beta_i}, P_{beta_i})$$

        (1) G_k is able to perform the subaction using the recipe R_{β_i}:

$$\text{CBA}(G_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\})$$

        (2) G_k has a full individual plan to do the subaction that uses the recipe R_{β_i}:

$$\text{FIP}(P_{\beta_i}, G_k, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha}), T_p)$$

    (4) GR mutually believe that all members of the group are committed to agent G_k's being able to do subaction β_i:

$$\begin{aligned}&\text{MB}(\text{GR}, (\forall G_j \in \text{GR}, G_j \neq G_k)\\ &\quad \text{Int.Th}(G_j, (\exists R_{\beta_i})\text{CBA}(G_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}),\\ &\qquad T_p, T_{\beta_i}, C_{cba/\beta_i/\alpha}), T_p)\end{aligned}$$

Fig. 19. FSP: core case: single-agent action.

Although the preceding clauses have some analogue in the definition of FIP, Clause (2a4) does not. This clause establishes the intentions (Int.Th's) needed to mesh G_k's individual plan for doing β_i with the plans for other subsidiary actions. Together with the axioms to avoid conflicting intentions (Fig. 4), it ensures that agents will not knowingly adopt subplans that conflict. In addition, as discussed in the next subsection, the Int.Th in Clause (2a4) is the source of other group members helping G_k to do β_i, which again would be much more difficult to achieve with Clause (0) and beliefs alone.

For example, Clause (2a4) ensures that Kate will not adopt an intention to use the (only) lasagna pan for making her appetizer, because that would conflict with her intention that Dan be able to make a lasagna main course. The Int.Th of Clause (0) is not sufficient in itself to ensure meshing subplans. Kate's intention that she and Dan make dinner will not by itself prevent her from using the lasagna pan; she might believe there is an alternative recipe for making dinner (e.g., Dan could make spaghetti instead) that would not conflict with her intention to use the lasagna pan. Although Kate's belief that Dan intends to make lasagna as part of their plan to make dinner, together with Clause (0), could be used to achieve the same constraint as Clause (2a4), such an approach would necessitate more complex mechanisms for reconciling intentions.

Clause (2a3) specifies the beliefs and intentions that G_k and the other group members must have if β_i is a complex action. Clause (2a3a) specifies that G_k have a complete individual plan to do β_i. Clause (2a3b) states that the other group members must mutually believe that there is some recipe which G_k can use to perform β_i and that G_k has a complete plan to do β_i. The different scopings of the existential quantifier in Clauses (2a3a) and (2a3b) accurately capture an important distinction. Whereas G_k must know the recipe it will use in its FIP to perform β_i, the other members of the group do not need to know this recipe. Rather, the other members of the group need only to mutually believe that there is some recipe that G_k can use.

In addition to having shared knowledge about G_k's intentions to perform β_i, the group must also have shared knowledge about G_k's ability. That agent G_k itself believes it will be able to perform β_i, is established directly by the Int.To in Clause (2a1a) if β_i is basic level and, as described in Section 5.3, from the FIP in Clause (2a3a) and Theorem (T2) if it is a complex action. Other members of the group must hold two mutual beliefs: first, they must mutually believe that G_k is able to perform β_i; second they must mutually believe that G_k has a complete recipe for β_i and believes it is able to perform β_i according to that recipe. Neither of these beliefs entails the other, so they must be independently established.

The meals example may be used to illustrate these two types of belief and their difference. First, Dan must believe that there is some recipe Kate can use to make the appetizer; i.e., he must believe that Kate will be able to make the appetizer. Second, he must believe that Kate knows a particular recipe and believes she can make the appetizer using that recipe; i.e., he must believe that Kate believes she can make the appetizer. Dan might hold the first belief and not the second; he might think Kate can make the appetizer but also that she does not herself believe she can. Alternatively, he might believe she thinks herself capable and yet himself not believe she has a recipe that will work. For Kate and Dan to have a complete plan, Dan must hold both beliefs.

For basic-level β_i, Clause (2a2) establishes the group's beliefs in G_k's ability. Their mutual belief that G_k believes it is capable is entailed by their mutual belief in its intention-to [Clause (2a1b)] and the definition of Int.To. For complex β_i, Clause (2a3b1) establishes that the group believes G_k is able to do β_i. Their mutual belief that G_k has a complete recipe for β_i and believes it is capable of doing the actions in the recipe (i.e., that BCBA holds) follows from their mutual belief that G_k has a complete plan [Clause (2a3b2)] and Theorem (T2).

The difference in scoping between Clauses (2a3a) and (2a3b) is important here as well. G_k must know the recipe and believe it is able to perform or contract out all the subactions entailed in doing β_i. In contrast, the other members of the group can only have a weak form of belief in G_k's ability to perform β_i. The embedding of the CBA in Clause (2a3b1) in MB, accurately represents this weaker belief; it reflects the fact that

$\mathrm{FSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$

(2) *Core case*:

(b) *Multi-agent action*: A subgroup GR_k will perform the subaction β_i.

(1) There is a recipe R_{β_i} such that, the subgroup has a full SharedPlan to do the subaction using this recipe:

$$(\exists R_{\beta_i}, P_{\beta_i})\mathrm{FSP}(P_{\beta_i}, \mathrm{GR}_k, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})$$

(2) The group GR mutually believe that there is a recipe R_{β_i} such that,

$$\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_i})$$

(a) the subgroup is able to perform the subaction using the recipe R_{β_i}:

$$\mathrm{CBAG}(\mathrm{GR}_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\})$$

(b) the subgroup has a full SharedPlan to do the subaction using this recipe:

$$\mathrm{FSP}(P_{\beta_i}, \mathrm{GR}_k, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha}), T_p)$$

(3) The full group GR mutually believe that all members in the group are committed to the subgroup GR_k being able to do the subaction:

$$\begin{aligned}&\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR} \setminus \mathrm{GR}_k)\\ &\quad \mathrm{Int.Th}(G_j, (\exists R_{\beta_i})\mathrm{CBAG}(\mathrm{GR}_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}),\\ &\qquad T_p, T_{\beta_i}, C_{cbag/\beta_i/\alpha}), T_p)\end{aligned}$$

Fig. 20. FSP: core case: multi-agent action.

other members of the group may not know the recipe G_k is using. To establish even this weak form of belief, however, agents must communicate enough about their individual or subgroup plans to convince other agents of their abilities to carry out constituent actions.

As discussed in the introduction to this section, this definition provides only minimal constraints on shared knowledge of the recipes for constituent acts. Agents can only avoid conflicting intentions, form subplans that mesh, and assist their collaborators to the extent they have knowledge about recipes and the resources they require. By stipulating minimal constraints, the definitions provide a framework in which designers can examine trade-offs; stronger constraints, both on recipe knowledge and on communication demands, can be added where warranted.

The multi-agent component of the core case (Fig. 20) is, significantly, missing two elements of the single-agent component. First, there are no clauses representing an intention to do the subaction β_i corresponding to Clause (2a). Intentions are individual attitudes; β_i is a multi-agent action to be done by a group. The need for the members of the group to have intentions to do the single-agent subactions entailed in the recipe selected for β_i (or its children subactions) will be established in the recursively embedded individual plans for these subactions. Second, because β_i is a multi-agent action it is necessarily a complex action; thus there is no clause for basic-level actions corresponding to Clause (2a2).

Clauses (2b1) and (2b2) have the same difference in quantifier scoping as Clauses (2a3a) and (2a3b). Again, the full group needs to have mutual belief that there is some recipe that the subgroup GR_k plans to use for doing β_i, whereas the members of the subgroup must know the recipe. This difference in recipe knowledge is also reflected in different ability constraints on the performing subgroup and the rest of the group. As in the single-agent case, because the members of the group who are not in the subgroup GR_k may not know the recipe the subgroup is using, they can only have a weak form of belief in the abilities of the subgroup to perform the action. Clause (2b2a) embeds CBAG in MB to represent this level of belief in ability. The constraints on members of the subgroup GR_k are stronger; they must know the recipe for β_i and believe that together they will be able to perform or contract out each of the actions in the recipe. Just as in the single-agent case, the full group must also believe that the subgroup GR_k knows a complete recipe for β_i and believes it is capable of doing the actions in the recipe. As described in Section 6.5, all of the requisite ability beliefs are entailed by the FSP definition.

As with single-agent actions, the full group must be committed to the subgroup's ability to perform β_i. Clause (2b3) in Fig. 20 is identical to Clause (2a4) in Fig. 19 for single-agent actions, with the exception of not including the subgroup GR_k rather than just the individual agent G_k in the members who hold the Int.Th. The subgroup is excluded from this Int.Th to avoid unnecessary redundancy; the commitments and intentions related to β_i of members of the subgroup are established through the FSP in Clause (2b1). Meshing subplans and helpful behavior derive from this intention-that.

Clauses (0), (2a4), and (2b3) establish a significant distinction between the full SharedPlans of two agents and the situation in which one of these agents contracts to another. In a full SharedPlan, the group GR comprises agents all of whom are committed (1) to the performance by the group of α; (2) through Int.To's and Int.Th's to the subactions β_i in the recipe they use for doing α; and (3) through the context parameter, to β_i's being done as part of doing α. In contrast, when in doing an action α' one agent, G, contracts to another, G_c, the doing of some subsidiary action β_i', G has an intention-that toward G_c's successful performance of β_i', but G_c does not necessarily have any intentions-that toward G's success in doing α'.

The contracting case for SharedPlans divides into four subcases. These vary along two dimensions, (1) the subaction β_i may be either single-agent or multi-agent; and (2) the contracting action γ may be either single-agent or multi-agent, requiring respectively that an individual or a subgroup do the contracting. As a result, the full definition of this case is quite long and Figs. 21 and 22 contain only the high-level detail. However, this case can be formalized simply by combining elements of contracting from the FIP with elements from the core case of FSP. The only additional machinery needed to handle contracting comes in extending the Elaborate_Group process. The group needs mechanisms for deciding on a contractor. We note, though, that the determination of group members who will perform the contracting action follows the same process as identification of agents for subactions in the core case.

To simplify the presentation of the contracting case, we identify two subsidiary meta-predicates, MP (MemberPerform) and SGP (SubgroupPerform). MP is directly analogous to the single-agent portion of the FSP (i.e., to Clauses (2a3)-(2a4) in Fig. 19) with the contracting action γ replacing the subaction β_i and other parameters of the operators adjusted accordingly. Similarly, SGP is directly analogous to the multi-agent

$\text{FSP}(P, \text{GR}, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$

(3) *Contracting case*: $\text{FSPC}(\text{GR}, \beta_i, T_{\beta_i}, T_p, C_\alpha, \{\rho_j\})$
(a) *Single-agent subaction*: By doing γ, the group GR will get another agent G_c to do the subaction β_i.
(1) The group GR mutually believe that all members of the group are committed to G_c's ability to perform β_i:

$$\text{MB}(\text{GR}, (\forall G_j \in \text{GR}) \\ \text{Int.Th}(G_j, (\exists R_{\beta_i})\text{CBA}(G_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}), \\ T_p, T_{\beta_i}, C_{cba/\beta_i/\alpha}), T_p)$$

(2) (a) The "contracting" act γ is single-agent; there is a member of the group G_k such that,
(1) The group mutually believe that by doing γ G_k can get G_c to do β_i:

$$\text{MB}(\text{GR}, \text{GTD}(G_k, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \text{constr}(c_\alpha) \cup \{\rho_j\}), T_p)$$

(2) G_k intends to do the "contracting" action:

$$\text{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha})$$

(3) The group mutually believe that G_k intends to do the "contracting" action:

$$\text{MB}(\text{GR}, \text{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}), T_p)$$

(4) The group has the requisite mutual beliefs about G_k's abilities and plans to do γ and Int.Th's G_k succeed;
G_k has the requisite beliefs, abilities, and plans:

$$\text{MP}(\text{GR}, G_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})$$

(b) The "contracting" act is a multi-agent action; there is a subgroup GR_k of the group such that,
(1) The group mutually believe that by doing γ subgroup GR_k can get G_c to do β_i:

$$\text{MB}(\text{GR}, \text{GTD}(\text{GR}_k, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) The group has the requisite mutual beliefs about GR_k's abilities and plans to do γ and Int.Th's GR_k succeed; GR_k has the requisite abilities and plans:

$$\text{SGP}(\text{GR}, \text{GR}_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})$$

Fig. 21. Contracting in a FSP: single-agent subactions.

portion of the FSP (i.e., to Clauses (2b1)–(2b3) in Fig. 20) with the contracting action γ replacing the subaction β_i and other parameters of the operators adjusted accordingly. These meta-predicates are expanded in Fig. B.5 in Appendix B. [28]

6.3. *Intentions-that in SharedPlans*

It is quite apparent from the FSP definition that Int.Th plays a central coordinating role in the formalization of collaborative plans. Agents' intentions-that toward the capabilities of other agents and toward the successful performance of the actions of groups of which they are a part are key to achieving the collaboration needed for their joint actions to succeed. The Int.Th of Clause (0) of the FSP definition is the source both of avoiding the

[28] To simplify the full definitions, given in Fig. B.4 in Appendix B, the MP and SGP meta-predicates are used there as well.

$\mathrm{FSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$

(3) *Contracting case*: $\mathrm{FSPC}(\mathrm{GR}, \beta_i, T_{\beta_i}, T_p, C_\alpha, \{\rho_j\})$

(b) *Multi-agent subaction*: By doing γ, the group GR will get another group of agents GR_c to do the subaction β_i.

(1) The group GR mutually believe that all members of the group are committed to GR_c's ability to perform β_i:

$$\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR}) \; \mathrm{Int.Th}(G_j, (\exists R_{\beta_i})\mathrm{CBAG}(\mathrm{GR}_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_\beta, C_{cbag/\beta_i/\alpha}), T_p)$$

(2) (a) The "contracting" act γ is single-agent; there is a member of the group G_k such that,

(1) The group mutually believe that by doing γ G_k can get GR_c to do β_i:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(G_k, T_\gamma, \gamma, \mathrm{GR}_c, \beta_i, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) G_k intends to do the "contracting" action:

$$\mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}), T_p)$$

(3) The group mutually believe that G_k intends to do the the "contracting" action:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}), T_p)$$

(4) The group has the requisite mutual beliefs about G_k's abilities and plans to do γ and Int.Th's G_k succeed; G_k has the requisite beliefs, abilities, and plans:

$$\mathrm{MP}(\mathrm{GR}, G_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})$$

(b) The "contracting" act is a multi-agent action; there is a subgroup GR_k of the group such that,

(1) The group mutually believe that by doing γ subgroup GR_k can get G_c to do β_i:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(\mathrm{GR}_k, \gamma, T_\gamma, \mathrm{GR}_c, \beta_i, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) The group has the requisite mutual beliefs about GR_k's abilities and plans to do γ and Int.Th's GR_k succeed; GR_k has the requisite beliefs, abilities, and plans:

$$\mathrm{SGP}(G, \mathrm{GR}_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})$$

Fig. 22. Contracting in a FSP: multi-agent subactions.

adoption of intentions that conflict with the joint activity and of communication actions. Intentions-that required in the core and contracting cases ensure that the subsidiary plans (individual and group) for doing the subsidiary actions β_i in R_α mesh and lead agents to form intentions to help each other in the performance of the group action.

Alternative approaches to modeling collaboration (see [28,29,40,61]) represent the Clause (0) aspect of commitment to the joint activity using some kind of joint-intention operator. These joint-intention operators do not treat meshing subplans; the approaches neither provide in general for this constraint[29] nor accommodate helpful behavior. In contrast, we are able to use a single modal operator, Int.Th, to provide all three properties of collaborative activity.

[29] Sonenberg et al. (op. cit.) achieve meshing subplans, but only do so when agents have complete plans. The meshing is assured by using pre-established recipes. As will become apparent in the next section, we cover partial plans and partial recipe knowledge.

Axiom (A5): G Int.Th some *prop* which G does not believe is true:

$$(\forall \alpha, T_\alpha)[\\ \quad \text{Int.Th}(G, prop, T_i, T_{prop}, C_{prop}) \wedge \neg\text{Bel}(G, prop, T_i) \wedge$$

G believes it can do something (α) to help:

$$\text{Bel}(G, (\exists R_\alpha)[[\text{Do}(G, \alpha, T_\alpha, \text{constr}(C_{prop})) \Rightarrow prop] \wedge \\ \text{CBA}(G, \alpha, T_\alpha, R_\alpha, \text{constr}(C_{prop}))], T_i) \Rightarrow$$

G will consider doing α:

$$\text{Pot.Int.To}(G, \alpha, T_i, T_\alpha, C_{\alpha/prop})]$$

Axiom (A6): G_1 Int.Th some *prop* which G_1 does not believe is true:

$$(\forall \alpha, T_\alpha)[\\ \quad \text{Int.Th}(G, prop, T_i, T_p, C_{prop}) \wedge \neg\text{Bel}(G, prop, T_i) \wedge$$

G_1 believes it can do something (α) that will help indirectly by allowing another agent to help directly:

$$\begin{aligned}
&\text{Bel}(G_1, \text{Do}(G_1, \alpha, T_\alpha, \text{constr}(C_{prop})) \Rightarrow \\
&\quad [[(\exists \beta, G_2, R_\beta, T_\beta) \\
&\quad\quad [\text{single.agent}(\beta) \wedge \\
&\quad\quad \text{CBA}(G_2, \beta, R_\beta, T_\beta, \text{constr}(C_{prop})) \wedge \\
&\quad\quad [\text{Do}(G_2, \beta, T_\beta, \text{constr}(C_{prop})) \Rightarrow prop]]] \vee \\
&\quad [(\exists \beta', \text{GR}_2, R_{\beta'}, T_{\beta'}) \\
&\quad\quad [\text{multi.agent}(\beta') \wedge \\
&\quad\quad \text{CBAG}(\text{GR}_2, \beta', R_{\beta'}, T_{\beta'}, \text{constr}(C_{prop})) \wedge \\
&\quad\quad [\text{Do}(\text{GR}_2, \beta', T_{\beta'}, \text{constr}(C_{prop})) \Rightarrow prop]]]], T_i) \wedge \\
&\text{Bel}(G, (\exists R_\alpha)\text{CBA}(G, \alpha, T_\alpha, R_\alpha, \text{constr}(C_{prop})), T_i) \Rightarrow
\end{aligned}$$

G will consider doing α:

$$\text{Pot.Int.To}(G, \alpha, T_i, T_\alpha, C_{\alpha/prop})]$$

Fig. 23. Axioms for intending-that.

Several axioms are needed to support the roles of Int.Th in ensuring that agents avoid conflict, assist each other, and provide status information when necessary. The axioms to avoid conflicting intentions in Fig. 4 constrain an agent's adoption of intentions (both intentions-to and intentions-that) so that it does not simultaneously hold conflicting intentions. Figs. 23 and 24 provide axioms that represent the adoption of helpful behavior. We describe them briefly here, but a full formalization of Int.Th must be the subject of another paper.

Axiom (A5) applies when an agent has an intention-that toward some proposition, currently does not believe this proposition holds and, furthermore, believes it is able to do some act α that will bring about the proposition's holding. Thus, the axiom will only apply if the time of the proposition, T_{prop}, is in the future. The axiom states that under these conditions, the agent will consider doing α. The agent adopts a potential intention to do α that will cause deliberation about adopting an intention to do it, and, barring conflicts, lead to this becoming a full-fledged intention. For example, if Kate believes

Axiom (A7):

$(\forall \eta, G_1, T_\eta, T_i)[$

G_1 *is committed to* G_2*'s success in doing* β *as part of* G_1*'s participation in a SharedPlan for* α:

$$
\begin{aligned}
&[\text{single.agent}(\eta) \wedge \\
&\quad (\exists \alpha, T_\alpha, R_\alpha, \beta, G_2)[\\
&\quad\quad \text{multi.agent}(\alpha) \wedge \\
&\quad\quad (\exists \text{GR})[\\
&\quad\quad\quad (G_1 \in \text{GR}) \wedge \text{SP}(\text{GR}, \alpha, T_i, T_\alpha, C_\alpha) \wedge \\
&\quad\quad\quad [[\text{single.agent}(\beta) \wedge G_2 \in \text{GR} \wedge \\
&\quad\quad\quad\quad \text{Int.Th}(G_1, (\exists R_\beta)\text{CBA}(G_2, \beta, R_\beta, T_\beta, \text{constr}(C_{\beta/\alpha})), T_i, T_\beta, C_{cba/\beta/\alpha})] \otimes \\
&\quad\quad\quad\ [\text{multi.agent}(\beta) \wedge G_2 \subseteq \text{GR} \wedge \\
&\quad\quad\quad\quad \text{Int.Th}(G_1, (\exists R_\beta)\text{CBAG}(G_2, \beta, R_\beta, T_\beta, \text{constr}(C_{\beta/\alpha})), T_i, T_\beta, C_{cba/\beta/\alpha})]] \wedge
\end{aligned}
$$

It is "cheaper" to G_1 to help G_2 in doing β by doing η:

$$
\begin{aligned}
&[\text{cost}(G_1, \text{Do}(\text{GR}, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_\alpha, C_\alpha, R_\alpha, \\
&\quad\quad \neg\text{Do}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})) \wedge \text{Do}(G_2, \beta, T_\beta, \text{constr}(C_{\beta/\alpha}))) - \\
&\quad \text{cost}(G_1, \text{Do}(\text{GR}, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_\alpha, C_\alpha, R_\alpha, \\
&\quad\quad \text{Do}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})) \wedge \text{Do}(G_2, \beta, T_\beta, \text{constr}(C_{\beta/\alpha}))) > \\
&\quad \text{econ}(\text{cost}(G_1, \text{Do}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})), T_\eta, C_{\eta/cba/\beta/\alpha}, R_\eta))] \wedge
\end{aligned}
$$

G_1 *believes it can perform* η:

$$\text{Bel}(G_1, (\exists R_\eta)\text{CBA}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})), T_i)]]] \Rightarrow$$

G_1 *will consider doing* η:

$$\text{Pot.Int.To}(G_1, \eta, T_i, T_\gamma, C_{\eta/cba/\beta/cba})]$$

Fig. 24. Helpful-behavior axiom for intending-that.

Dan may *not be able* to make the main course and furthermore believes that she can take some action to remove a possible roadblock to his being able to do so (e.g., pick up his child at child care), then Kate will adopt a potential intention to do that action. However, if Kate believes that Dan is capable of making the main course in the current situation, then Axiom (A5) would not apply.

Axiom (A6) provides for more indirect helpful behavior. It states that if an agent has an intention-that toward some proposition that it currently does not believe holds and the agent believes it is able to do some act α that will bring about a condition enabling another agent (or group of agents) to do an action β that will bring about the proposition's holding, then the agent will consider doing α. For instance, if Kate believes if she calls Jon he will pick up Dan's child and thus enable Dan to make the main course, then she will adopt a potential intention to do so. The potential intention to do α will cause deliberation about adopting an intention to do it, and, barring conflicts,[30] lead to this becoming a full-fledged intention.

[30] The conjunct, "and assuming sufficient resources", belongs in this phrase; however, as mentioned in Section 4.3.3, the specification of the processes for reasoning about costs and resource bounds is beyond the scope of this paper.

Axiom (A7) provides a basis for helpful behavior in the SharedPlan context, i.e., for helping a collaborative partner. It uses two auxiliary functions, *cost* and *econ*. The function cost computes the costs to an agent of the performance of an action; $\text{cost}(G_e, \text{Do}(G_p, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_\alpha, C_\alpha, R_\alpha, \Theta)$ refers to G_e's cost, given the constraints Θ, of G_p doing α at time T_α in the context C_α using recipe R_α. G_p may be either an individual or a group; if an individual, it may be G_e or some other agent. The function econ provides a means of relativizing cost tradeoffs; it specifies the proportionate amount of savings required for helpful behavior to be worth the effort required.

The initial clauses in Axiom (A7) establish a collaborative context: agent G_1 is a member of a group GR that has a SharedPlan to perform the action α; G_2 is either a member of GR or a subgroup of the group GR; G_1 has an intention-that G_2 will be able to perform (CBA or CBAG depending on whether G_2 is a single agent or a subgroup) the action β where β is being done as part of GR's plan to do α. In this context, the axiom asserts that G_1 will adopt a potential intention to do an action η if G_1 believes that its own overall cost [31] of the group GR's doing α is less when it does η to assist G_2's performance of β than it is when G_2 does β and G_1 does not help by doing η. In the meals example, Axiom (A7) would account for why Dan would offer to pick up the ingredients Kate needs for the mushroom puffs. It also would account for why, while chopping onions for the sauce for the lasagna, Dan would chop an extra one for Kate to use in making the mushroom puffs.

The consequent in Axiom (A7) is a potential intention rather than a full-fledged intention, because the cost computation is local; it does not involve consideration of competing intentions. The axiom reflects an intuition that it is useful to delay the processes involved in forming a full-fledged intention for a helpful action until after an agent has determined that the performance of the action would be beneficial.

Subgroups can also provide helpful behavior; i.e., the helping action η might be a multi-agent action for which a subgroup of GR forms a SharedPlan. In this case, the cost evaluation is different. Furthermore, the formation of the helping subgroup and its adoption of a subsidiary SharedPlan to do η are more complicated.

Helpful behavior is also appropriate in contracting situations. For instance, in the meals example, Dan might offer to pick up the noodle ingredients for Tony, if he's contracted to Tony to make the noodles for the lasagna. A straightforward analogue of Axiom (A7) that captures this case is given in Appendix B. A separate axiom is needed because the cost evaluation differs; this difference arises because in contracting the agent evaluating the cost of possible helpful behavior is, by itself, the agent doing α. Although the underlying intentional motivations also differ (an Int.To rather than a SP), this difference alone could be captured with a simple disjunction in Axiom (A7).

6.4. *Partial SharedPlans*

Partial SharedPlans, like their counterpart partial individual plans, differ from full ones in four ways as identified in Clauses (1)–(4) of the outline of PSP in Fig. 25. In

[31] This axiom is more straightforward than, though it presumes a less charitable G_1 than, an alternative axiom in which the cost to other agents and not just to G_1 would be lower.

PSP(P, GR, α, T_p, T_α, C_α)

(0) The group GR has mutual belief that all members of the group are committed to the success of the group doing α:

$$\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Int.Th}(G_j, \mathrm{Do}(\mathrm{GR}, \alpha, T_\alpha, \mathrm{constr}(C_\alpha)), T_p, T_\alpha, C_\alpha), T_p)$$

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

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

(2) *Core case*:
  (a) *Single-agent subaction*: A member of the group G_k intends to do the subaction, but may have only a partial plan for doing it.
  (b) *Multi-agent subaction*: A subgroup GR_k has a shared plan (SP) to do the subaction, but this plan may be only partial.
(3) *Contracting case*:
  (a) *Single-agent subaction*: The group has decided to subcontract an outside agent G_c to do the subaction, but may have only a partial plan (PIP or PSP) for doing the "contracting" action.
  (b) *Multi-agent subaction*: The group has decided to subcontract an outside group GR_c to do the subaction, but may have only a partial plan (PIP or PSP) for doing the "contracting" action.
(4) *Unreconciled case*: GR has not deliberated about the subaction; no decision has been made about which agent(s) will do it.

Fig. 25. English description of the PSP (partial SharedPlan) definition.

particular, (1) the agents may have only a partial recipe for doing the action; (2) they may have only partial individual plans or partial SharedPlans for doing some of the subsidiary actions in the recipe; (3) they may have only partial individual plans or partial SharedPlans for doing some of the contracting actions; and (4) there may be some subactions about which the group has not deliberated and for which there is as yet no agent (individual or subgroup) selected to perform the subaction. Thus, as in the case of PIP, the formalization of PSP distinguishes those actions β_r about which the group has deliberated and for which it has chosen an agent (Clause (2), the "core case") from those actions β_k for which it has not yet decided on an agent (Clause (4), the "unreconciled case"). Because the contracting case [Clause (3)] for PSP differs from that for FSP in the same ways that the core cases do, we do not discuss it further, and in Appendix B we include only the abridged version with English glosses of the contracting case.

By handling both complex types of actions and partiality, the formalization of partial SharedPlans constitutes an advance over previous approaches. The complexities introduced by doing so are most evident in two places: (1) the treatment of partiality of knowledge about the recipe to be used [Clause (1)], and the corresponding Select_Rec_GR process; (2) the handling of unreconciled actions and the corresponding Elaborate_Group process. Partiality in the core case can be treated by recursion on either individual plans (single-agent actions) or SharedPlans (multi-agent actions). We treat this case briefly first, and then discuss the recipe and unreconciled cases. In each

PSP(P, GR, α, T_p, T_α, C_α)

(1) The group GR mutually believe that there is a recipe for α:

$$\text{MB}(\text{GR}, (\exists R_\alpha)[\{\beta_i, \rho_j\} \subseteq R_\alpha \wedge R_\alpha \in \textit{Recipes}(\alpha)], T_p)$$

(a) GR has only a partial recipe:

$$[\text{MB}(\text{GR}, (\exists R_\alpha)[\{\beta_i, \rho_j\} \subseteq R_\alpha \wedge R_\alpha \in \textit{Recipes}(\alpha)], T_p) \Rightarrow [$$

(b) There is a recipe $R_{select.rec.g}$ for finding the full appropriate recipe for α such that the group GR has a Full SharedPlan for finding the recipe for α using the recipe $R_{select.rec.g}$:

$$\text{FSP}(P_{select.rec.g}, \text{GR}, \text{Select\_Rec\_GR}(\text{GR}, \alpha, \{\beta_i, \rho_j\}, T_{select.rec.g}), T_p, T_{select.rec.g}, R_{select.rec.g}, C_{select.rec.g.\alpha})$$

where every subaction in the selected recipe R_α, is one that GR mutually believes it can either perform or contract:

$$\text{constr}(C_{select.rec.\alpha}) \supseteq (\exists R_\alpha = \{\delta_r, \kappa_r\})[$$

GR mutually believe that R_α is an extension of $\{\beta_i, \rho_j\}$:

$$\text{MB}(\text{GR}, R_\alpha \in \textit{Recipes}(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha, T_{select.rec.g})$$

GR mutually believe that for all the subactions in the recipe for α it finds the following holds:

$$(\forall \delta_r \in R_\alpha)\text{MB}(\text{GR},$$

Single-agent subaction: there is an agent G_{δ_r} in the group that can bring about the act:

$$(\exists G_{\delta_r} \in \text{GR}, R_{\delta_r})\text{CBA}(G_{\delta_r}, \delta_r, T_{\delta_r}, R_{\delta_r}, \text{constr}(C_\alpha) \cup \{\kappa_r\})$$

Multi-agent subaction: there is a subgroup that can bring about the subaction:

$$(\exists \text{GR}_{\delta_r} \subseteq \text{GR}, R_{\delta_r})\text{CBAG}(\text{GR}_{\delta_r}, \delta_r, T_{\delta_r}, R_{\delta_r}, \text{constr}(C_\alpha) \cup \{\kappa_r\}))]$$

Contracting case: the group can contract the subaction to another agent or a group of agents:

$$\text{CCG}(\text{GR}, \delta_r, T_{\delta_r}, \text{constr}(C_\alpha) \cup \{\kappa_r\})$$

Fig. 26. PSP: finding a recipe.

instance we look first at the constraints on mental state imposed by the definition, and then at the constraints on the various processes involved in completing the partial plans.

Figs. 27 and 28 give the major constituents of the core case of the PSP for single-agent and multi-agent subactions. Two constraints are placed on the design of collaborating agents. First, in elaborating their individual and group plans for subactions, they must develop plans that mesh so that the intentions-that in Clauses (2a3) and (2b4) hold. Thus, agents' elaboration processes must take into account the constraints imposed by the agents' intentions-that other agents are able to do their parts. The constraints that are imposed are contained in the constraints parameters of the Int.To, Int.Th and SP clauses

$\mathrm{PSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha)$

(2) *Core case*:

(a) *Single-agent subaction*: A member of the group G_k will do the subaction (but may not have a full plan for it).

(1) G_k's intention and related beliefs:

(a) G_k intends to do β_r:

$$\mathrm{Int.To}(G_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha})$$

(b) GR mutually believe that the agent G_k intends to do the act:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{Int.To}(G_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha}), T_p)$$

(2) GR mutually believe that G_k can bring about the action:

$$\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_r})\mathrm{CBA}(G_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(3) The group mutually believe that all of its members are committed to G_k's success:

$$\begin{aligned}&\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR}, G_j \neq G_k)\\ &\quad \mathrm{Int.Th}(G_j, (\exists R_{\beta_r})\mathrm{CBA}(G_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}),\\ &\qquad T_p, T_{\beta_r}, C_{cba/\beta_r/\alpha}), T_p)\end{aligned}$$

Fig. 27. PSP: core case: single-agent act.

$\mathrm{PSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha)$

(2) *Core case*:

(b) *Multi-agent action*: Subgroup GR_k will do the subaction (but they may not have a FSP for it).

(1) The subgroup has SharedPlan for the subaction:

$$\mathrm{SP}(P_{\beta_r}, \mathrm{GR}_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha})$$

(2) GR mutually believe that the subgroup has a SharedPlan to do the subaction:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{SP}(P_{\beta_r}, \mathrm{GR}_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha}), T_p)$$

(3) The group GR mutually believe that the subgroup can bring about the subaction:

$$\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_r})\mathrm{CBAG}(\mathrm{GR}_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(4) The group mutually believe that all of its members are committed to GR_k's success:

$$\begin{aligned}&\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \{\mathrm{GR} \setminus \mathrm{GR}_k\})\\ &\quad \mathrm{Int.Th}(G_j, (\exists R_{\beta_r})\mathrm{CBAG}(\mathrm{GR}_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}),\\ &\qquad T_p, T_{\beta_r}, C_{cbag/\beta_r/\alpha}), T_p)\end{aligned}$$

Fig. 28. PSP: core case: multi-agent act.

in the PSP definition. Second, collaborators must communicate sufficient information for the mutual beliefs of intentions [Clause (2a1b)], plans [Clause (2b2)], and ability [Clauses (2a2) and (2b3)] to be established. As in the case of individual plans, PSP imposes weaker constraints on agents' beliefs in their own capabilities to perform the constituent actions β_r, either individually or in subgroups, than does FSP. Section 6.5 describes how these constraints are met by PSP.

Partiality in the recipe, as specified in Fig. 26, leads to the need for the group to agree on a way to find, construct, or complete a recipe for the action. As we discussed in Section 4.5, the group action Select_Rec_GR, like the individual action Select_Rec, takes a partial recipe $\{\beta_i, \rho_j\}$ for the action α and extends this partial recipe to a complete one, $\{\delta_r, \kappa_e\}$. Although this process is analogous to Select_Rec, it leads to different kinds of intentions being considered by the agents in the group. In particular, in place of the Pot.Int.To's that Select_Rec generates, Select_Rec_GR generates the Pot.Int.Th's in Clause (4) of PSP for all the group members. Select_Rec_GR is more complex than Select_Rec for several reasons. First, any Select_Rec_GR process must include a way for the group to reach agreement on the recipe. Because we allow very general Select_Rec_GR actions, some may include ways for the group to reach consensus on the recipe for getting a recipe as well as on the recipe itself. Second, Select_Rec_GR processes will often entail individual agents invoking their own Select_Rec processes; because partial recipes of different group members may be combined for a complete recipe, agents need ways to determine when to stop with a partial solution.[32]

The unreconciled case of the PSP is given in Fig. 29. The definition allows for β_k either to be done directly by group members or to be contracted out. For each unreconciled action β_k to be done by a group member, the PSP definition requires that the members of the group GR mutually believe some agent (single-agent actions) or subgroup (multi-agent actions) is capable of doing the action. In addition, all members of GR must potentially intend-that there be an individual or subgroup to perform the action. Clauses (4a1b) and (4a2b) engender this second constraint; from Axiom (A4) and the definition of MB, each agent must actually have the potential intentions-that embedded in these clauses.

The notable feature of the treatment of unreconciled actions β_k that are contracted out is that the definitions allow for participants to consider both individual and subgroup contracting. That is, regardless of whether β_k is single-agent or multi-agent, both single-agent and multi-agent contracting actions are possible. In addition to stipulating mutual belief that the group can contract out β_k [Clause (4b1)], the definition requires that all members of the group adopt potential intentions that there be a contracting action γ and a contractor G_c (individual or group) such that some member or subgroup performs γ and this performance of γ will suffice to get G_c to perform β_k [Clause (4b2b)].

The differences between the intentions and beliefs about capability required in the unreconciled case [Clause (4)] and those in the core [Clause (2)] and contracting [Clause (3)] cases yield additional requirements on the Elaborate_Group process. To move constituent actions from this case to the core case, the Elaborate_Group process must provide for the agents (1) to reach agreement on which agent(s) will perform the action; (2) to adopt appropriate intentions; (3) to communicate sufficient information that the requisite mutual beliefs about agents' abilities and intentions to do actions are established.

[32] We note that there is no clause in this part of the PSP definition corresponding to the Int.To in Clause (1b1) of the PIP definition. The formalization does not require any group intention; that the group has a FSP for selecting an appropriate recipe suffices to represent the necessary mental state conditions. The commitment represented by the Int.To of the individual case is satisfied by the intending-that in Clause (0) of the FSP.

PSP(P, GR, α, T_p, T_α, C_α)

(4) *Unreconciled case*: GR hasn't deliberated on the subaction.

(a) GR considers that one of its members or a subgroup will do the subaction.

(1) *Single-agent subaction*:

(a) The group GR mutually believe that there is a member of the group G_k that can perform the action:

$$\mathrm{MB}(\mathrm{GR}, (\exists G_k \in \mathrm{GR}, R_{\beta_k})\mathrm{CBA}(G_k, \beta_k, R_{\beta_k}, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(b) The group GR mutually believe that all its members are considering being committed to the performance of the subaction by that agent:

$$\begin{array}{l}\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\\ \quad \mathrm{Pot.Int.Th}(G_j, (\exists G_k \in \mathrm{GR}, T_{\beta_k})\mathrm{Do}(G_k, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}),\\ \quad\quad T_p, T_{\beta_k}, C_{do/\beta_k/\alpha}), T_p)\end{array}$$

(2) *Multi-agent subaction*:

(a) The group GR mutually believe that there is a subgroup GR_k that can perform the action:

$$\begin{array}{l}\mathrm{MB}(\mathrm{GR}, (\exists \mathrm{GR}_k \subseteq \mathrm{GR}, R_{\beta_k})\\ \quad \mathrm{CBAG}(\mathrm{GR}_k, \beta_k R_{\beta_k}, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)\end{array}$$

(b) The group GR mutually believe that all of its members are considering being committed to the performance of the subaction by a subgroup:

$$\begin{array}{l}\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\\ \quad \mathrm{Pot.Int.Th}(G_j, (\exists \mathrm{GR}_k \subseteq \mathrm{GR}, T_{\beta_k})\mathrm{Do}(\mathrm{GR}_k, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}),\\ \quad\quad T_p, T_{\beta_k}, C_{do/\beta_k/\alpha}), T_p)]\end{array}$$

(b) *Contracting case*: GR considers getting another agent or a subgroup G_c to do the subact β_k.

(1) The group GR mutually believe that they can contract the subaction β_k:

$$\mathrm{MB}(\mathrm{GR}, (\exists T_{\beta_k})\mathrm{CCG}(\mathrm{GR}, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) The group GR mutually believe that all its members are considering being committed to the performance of a contracting action γ by a member or a subgroup G_k, and that by doing γ, G_k can get a contractor or a group of contractors to do the subact β_k:

$$\begin{array}{l}\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\\ \quad \mathrm{Pot.Int.Th}(G_j, (\exists \gamma, T_\gamma, G_c, G_k)\\ \quad\quad [\,[(\mathrm{single.agent}(\gamma) \wedge G_k \in \mathrm{GR}) \otimes (\mathrm{multi.agent}(\gamma) \wedge G_k \subseteq \mathrm{GR})] \wedge\\ \quad\quad \mathrm{Do}(G_k, \gamma, T_\gamma, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}) \wedge\\ \quad\quad \mathrm{GTD}(G_k, \gamma, T_\gamma, G_c, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\})\,],\\ \quad\quad T_p, T_{\beta_k}, C_{cont/\beta_k/\alpha}), T_p)]\end{array}$$

Fig. 29. PSP: unreconciled case.

The unreconciled case of the PSP is thus more complex than that of the PIP for several reasons. First, group decision making processes are required to determine which individuals or subgroups will do each β_k. Second, if the action is multi-agent, after the subgroup has been identified, its members must form a partial plan to do β_k. Third, agents must communicate sufficient information to the group to establish the mutual beliefs required in the core case. Finally, agents' reconciliation processes must be able to handle a greater variety of intentions and to weigh tradeoffs among intentions that derive from group activities and intentions that stem solely from individual plans.

The selection of an agent (or subgroup) is driven by the potential intention that there exist some agent (or subgroup) who does the action. This potential intention [in Clauses (4a1b) and (4a2b) respectively] must first be reconciled and turned into a full-fledged intention-that. The intention-that will then lead to some agent adopting an intention to do (or some subgroup forming a SharedPlan for) β_k and the remainder of the full group forming an intention that they will be able to do the action. So long as a group member can take some action that will lead to some agent (or subgroup) forming an intention to do (or SharedPlan for doing) β_k, Axioms (A5) and (A6) ensure that this action will be considered. That is, once an agent forms the intention that someone do β_k, it must consider doing actions that will help contribute to this intention-that being satisfied. For instance, it will consider doing β_k itself or trying to convince someone else to do β_k. Thus, the selection process requires several steps not evident from the PSP definition.

If β_k is a single-agent action, then some agent, G_k, must first adopt a potential intention to do β_k, then reconcile this intention with others and form both a partial individual plan to do this action and a full plan to elaborate the partial plan. In some circumstances, several group members may consider doing β_k; i.e., they may go through the cycle of forming a potential intention and reconciling it. This intention adoption process is similar to the one an agent uses when forming an intention to do a subaction within its Elaborate_Individual process for an individual plan. However, the alternatives to be weighed by the reconciliation process in the context of a SharedPlan differ from those available in the context of an individual plan. For instance, G_k may take into account the possibility and costs of having another agent in the group do β_k. G_k may also consult the members of the group for assistance in forming the partial individual plan required by the definition of Int.To, including asking for advice about the recipe to use.

If β_k is a multi-agent action, then some subgroup GR_k must construct a partial shared plan to do β_k to satisfy Clause (2b1) of the core case for multi-agent actions (Fig. 28). To do so, its members need (a) to agree to act jointly to do β_k; this is required to satisfy Clause (0) of the PSP for doing β_k; (b) to agree on the procedure $R_{select.rec.g_{\beta_k}}$ they will use to determine a recipe for doing β_k [Clause (1b)] and form a full shared plan for using $R_{select.rec.g_{\beta_k}}$; and (c) to agree on the Elaborate_Group process to be used, as required by Clause (2b) of the SP definition (Fig. 17). Again, the subgroup may consult other group members for advice on recipes.

In forming the full shared plan to use recipe $R_{select.rec.g_{\beta_k}}$ to determine a recipe for β_k, the group will transform the weak belief, represented in Clause (4a2a), that some subgroup could do β_k into the stronger belief about ability required for them to have a partial shared plan. In particular, they will establish the mutual belief that the recipe they select comprises constituent subactions that the members of GR_k mutually believe they either will be able to perform or will be able to contract out successfully. These mutual beliefs are required to satisfy the constraints component of $C_{select.rec.g/\alpha}$ of the FSP in Clause (1b). In the next section, we describe how the PSP represents the ability knowledge agents must have for subactions in the core case.

For both single- and multi-agent β_k, other group members must transform their potential intentions that some individual or subgroup do β_k [in Clauses (4a1b) and (4a2b)] into a full-fledged intention that the chosen agent(s) be able to perform the action [required by Clauses (2a3) and (2b4) of PSP's definition]. To make this transition, the agents must individually accept and together agree on the choice of agent(s) and reconcile potential intentions that the agent(s) be able to do β_k. Thus, these other group members must go through an intention reconciliation process similar to G_k's (or the members of GR_k), but they reconcile an intention-that, not an intention-to.

Furthermore, in both cases, the agent(s) committed to performing β_k must communicate sufficient information for other group members to be able to form the mutual beliefs about the abilities and intentions (or SharedPlans) of the individual (or subgroup) to do β_k required by Clauses (2a1b) and (2a2) [or Clause (2b2) and (2b3)] of PSP's definition. In addition, the other group agents must communicate the information required to establish mutual belief of their commitment to the individual (or subgroup) performance of β_k (Clauses (2a3) and (2b4) respectively).

Thus, the multi-agent actions Select_Rec_GR and Elaborate_Group are more complex than their single-agent counterparts. Each of these multi-agent actions must include participation in some group decision making process, as well as incorporating many of the actions in Select_Rec and Elaborate_Individual. Furthermore, the group needs to agree on procedures for reaching consensus. Sonenberg et al. [61] describe one set of mechanisms for group elaboration and role assignment within a formalization that includes complex actions but does not allow for partial recipes. As described previously, Jennings [29] utilizes a central organizer to make these decisions; his system handles partiality only in letting the time and agent of an individual agent action be unspecified initially. The problems of formalizing and designing such processes remain significant areas of inquiry. Designers of computer agents need to weigh the tradeoffs between the time required to identify an appropriate agent and the work entailed by multiple agents considering doing the same action. They may decide differently about the Elaborate_Group processes to use, depending on tradeoffs in areas such as communication demands, negotiating time, and centrality of control. The SharedPlan definition serves to constrain the range of possibilities.

6.5. *Capabilities to perform actions in SharedPlans*

Just as individual agents must believe they are capable of performing an action in order to have an intention to do that action, groups of agents must have a certain level of belief in their ability to perform actions for which they form SharedPlans. As in the case of individual plans, the requirements on agents' beliefs about their capabilities vary depending on whether their plan is complete or partial. The definitions of FSP and PSP only explicitly state some of the requisite beliefs; others are implicit in the definitions of these meta-predicates and their interactions with the definitions of Int.To, FIP, and PIP. To understand the kinds of collaborative behavior FSP and PSP yield and to guide agent design, it is useful to extract this information and specify it separately. Formal definitions and theorems establishing the requisite ability constraints are given in Appendix A. In this section we briefly describe the constraints the plan definitions place on ability knowledge.

For a group to have a complete SharedPlan to do the complex, multi-agent action α (i.e., for FSP to hold), the members must have determined a recipe for α, and they must believe that group members or contractors can perform all of the basic-level actions entailed by that recipe and that group members can perform all of the basic-level actions required for any contracting. Because knowledge of how to perform α is distributed in SharedPlans, no single agent may know the complete extended recipe tree for α. The nature of the particular beliefs agents must have about subactions in the recipe for α depends on whether the subaction is single- or multi-agent and on whether it is to be done by a group member or contracted out. In all cases, however, the individual agent(s) performing an action must have strong beliefs that they can perform the action; other group members must have both the weaker beliefs that the performing agents can bring about the action (representable by CBA or CBAG embedded in one of the belief operators), and a belief that the performing agents themselves have the requisite stronger beliefs in their own abilities.

A constituent single-agent subaction β_i performed by a group member G_k engenders the same belief constraints on G_k as FIP would; in particular, BCBA($G, \beta_i, \ldots$) must hold. A constituent multi-agent subaction β_i performed by subgroup GR_k requires that GR_k have ability beliefs for β_i analogous to those the original SharedPlan for α requires the full group to have for α; thus, the requisite ability beliefs for this case can be obtained by recursion. In addition, the full group must mutually believe both that G_k (or GR_k) will be able to do β_i (i.e., that CBA or CBAG respectively hold) and that the G_k (or GR_k) has a recipe it believes it will be able to carry out. For actions contracted out, analogous beliefs must hold about the contracting action; in addition, the group must mutually believe that the contracting action will have its intended effect.

CBAG embedded in MB is too weak a constraint on the performing agents' beliefs; it lacks a requirement that they know the relevant recipes. Unembedded CBAG is too strong; it presumes correct beliefs. The subsidiary meta-predicate MBCBAG ("mutually believe can bring about group") defined in Appendix A, Fig. A.3, represents the appropriate intermediate level of belief. As was the case for BCBA, the recipe for α, R_α, is an argument of the MBCBAG meta-predicate, reflecting the fact that a particular recipe is known to the subgroup members and not just the existence of some recipe. Only the mutual belief that R_α is a recipe for α is part of the definition. Within its definition, MBCBAG appears recursively for β_i that are multi-agent actions, and BCBA is stipulated for β_i that are single-agent actions. For each of these recursive uses, the existence of the recipe R_{β_i} is stipulated outside any belief context, reflecting the agent(s)'s having identified a particular recipe. Thus, MBCBAG is stronger than CBAG embedded in MB; however, MBCBAG is weaker than unembedded CBAG because it does not presume the agent's beliefs about the recipes are correct.

As Theorem (T4) states, FSP entails MBCBAG. The belief requirements for single-agent β_i done by a group member [MBCBAG Clause (2)] follow from the FIP in FSP Clause (2a3a) and Theorem (T2) or are in FSP Clauses (2a2a) and (2a3b1). The belief requirements for multi-agent β_i done by a subgroup [Clause (3)] are entailed by FSP Clause (2b1) and recursion. The full group's belief that the subgroup GR_k has a complete recipe for β_i and believes it is capable of doing the actions in the recipe follows from GR's mutual belief that GR_k has a complete plan [Clause (2b2b)]. The belief requirements for actions that are contracted out are entailed by FSP Clause (3) and either FIP [Clauses (3a2a) and (3b2a)] or FSP [Clauses (3a2b) and (3b2b)] applied recursively to the contracting action. The proof of Theorem (T4) is similar to that of Theorem (T2).

When a group has only a partial SharedPlan for α, its beliefs about capabilities of agents to carry out the necessary subactions may be quite limited, because its knowledge of the recipe it will use is incomplete. The group may only believe that there is some way to find a recipe that it can use to perform α. Until it knows the constituent actions in the recipe for α, the group cannot make any determination about which individual or subgroup will perform the subactions or about their abilities to perform these subactions. While the lack of a recipe makes the agents' knowledge in this situation weaker than in the FSP, the group's beliefs are still stronger than CBAG embedded in MB yields. In particular, the agents must believe that they can determine a complete recipe and will be able to find members able to perform, or to contract out, each of the actions δ_v in the complete recipe once it is determined. As the agents identify portions of the recipe, they must also establish that they will be able to perform the actions in that portion or contract them out.

The subsidiary meta-predicate WMBCBAG ("weakly mutually believe can bring about group") defined in Appendix A, Fig. A.4, represents the beliefs the group GR must have about its abilities to select an appropriate recipe, and perform or contract out each of the constituent subactions in this recipe. This set of beliefs follows from the PSP definition, as stated in Theorem (T5). Clause (1) of WMBCBAG is established by PSP Clause (1); Clause (1b) follows from the FSP in PSP Clause (1b) and Theorem (T4); as shown in Fig. 26, the constraints component of the FSP context, constr($C_{select.rec/\alpha}$), contains the constraints in Clauses (1b1a)-(1b1h) of WMBCBAG.

Clauses (2)-(4) specify the beliefs in capabilities required for the subactions β_i of the portion of the recipe that has been identified. The mutual beliefs of Clauses (2a2) and (2b2) are established respectively by Clauses (2a2) and (2b3) of PSP. If a group member is going to perform β_i itself the Clause (2a1a) of PSP holds. If β_i is basic-level action, then Clause (2a1) is established by Clause (1) of Int.To. If β_i is a complex action, then the application of Theorem (T3) to Clause (2b1) of Int.To yields Clause (2a1) of WMBCBAG.

If a subgroup is going to perform β_i, then Clause (2b2) of PSP holds and Theorem (T5) applied recursively to Clause (2a) of SP yields Clause (2b1) of WMBCBAG. Clauses (3) and (4) of WMBCBAG are established similarly by Clauses (3) and (4) of PSP.

6.6. *An example: SharedPlan for dinner*

In this section we illustrate the SharedPlan formalization of collaborative activity by presenting the full shared plan for the example of Kate and Dan making dinner together. As presented in Section 2, in forming their collaborative plan, Kate and Dan decide that Kate will make the appetizer, Dan will make lasagna for the main course, and the two of them together will make the dessert. To flesh out the example, we add some additional details. Kate and Dan will cook at Dan's house. There is one constraint from the context in which they decide to form their joint plan, that the dinner making be done indoors. There is also one constraint in the recipe they choose for making dinner, that the kitchen of Dan's house be clean.

To simplify the example, we presume that the intentional context is the same for Kate and Dan. They both want to entertain their best friends, Joan and Max. However, more realistically, the motivations that lead Kate and Dan to decide to make dinner together might be different (e.g., Kate may decide to make dinner with Dan because she intends that she will not be hungry or tired whereas Dan's motivation is his intention that they have fun together).

We use the following representation in the formalization:

- *md* represents the group action of Kate and Dan making dinner together on May 8, between 19:00 and 21:00 at Dan's house; *md* is the top-level action, i.e., the binding of α in the SP definition;
- T_{md} denotes the time of *md*; $T_{md} = May8.19\text{–}21$;
- make-dinner denotes the action type of *md*;
- *D.house* denotes Dan's house, the location of *md*;
- T_p, the time of their collaborative plan, is May 8, 17:00;
- $entertain(\{K, D\}, \{J, M\}, T_{entertain})$ denotes the proposition that Kate and Dan entertain Joan and Max; $T_{entertain}$ is the time over which this proposition holds;
- C_{md} denotes the context of *md*;
- the intentional component of C_{md} contains Kate's and Dan's (individual) intentions that they entertain friends; i.e., the intentional context specifies that the following hold:

  (a) for Kate,

  $$\text{Int.Th}(K, entertain(\{K, D\}, \{J, M\}, T_{entertain}), T_i, T_{entertain}, C_{entertainK});$$

  (b) for Dan,

  $$\text{Int.Th}(D, entertain(\{K, D\}, \{J, M\}, T_{entertain}), T_i, T_{entertain}, C_{entertainD});$$

- $\text{constr}(C_{md}) = indoors(md)$; i.e., this is the constraint from the planning context, namely that *md* be done indoors;
- *ma* denotes Kate's making the appetizer; the action type of *ma* is making-appetizer;
- *mmc* denotes Dan's making the main course; the action type of *mmc* is making-main-course;
- *gmd* denotes the group action of Kate and Dan making the dessert; the action type of *gmd* is g-making-dessert;
- R_{md} denotes a particular recipe for *md*; this recipe has 3 constituent actions (*ma*, *mmc*, and *gmd*) and the constraint on Dan's kitchen. In particular,

  $$\begin{aligned} R_{md} = \{\{&\text{making-appetizer}(May8.19\text{–}19{:}30, D.house), \\ &\text{making-main-course}(May8.19{:}30\text{–}20, D.house), \\ &\text{g-making-dessert}(May8.20\text{–}21, D.house)\}, \\ &\{\text{clean-kitchen}(D.house)\}\} \end{aligned}$$

- C_{ma}, C_{mmc} and C_{gmd} denote the contexts for the three constituent subactions *ma*, *mmc* and *gmd* respectively. These contexts are all similar; in each case, the intentional context for action is the (individual) intention that they make dinner together and the constraints are $\text{constr}(C_{md}) \cup \{\text{clean-kitchen}(D.house)\}$.

Using this notation, the full shared plan definition specifies that Kate and Dan's complete plan, $\mathrm{FSP}(P, \{K, D\}, md, May8.17, May8.19\text{–}21, R_{md}, C_{md})$, expands to the following conjunction of clauses:

(0) $\mathrm{MB}(\{K, D\}, \mathrm{Int.Th}(K, \mathrm{Do}(\{K, D\}, md, May8.19\text{–}21, \mathrm{constr}(C_{md})),$
$May8.17, May8.19\text{–}21, C_{md}), May8.17) \wedge$
$\mathrm{MB}(\{K, D\}, \mathrm{Int.Th}(D, \mathrm{Do}(\{K, D\}, md, May8.19\text{–}21, \mathrm{constr}(C_{md})),$
$May8.17, May8.19\text{–}21, C_{md}), May8.17) \wedge$

(1) $\mathrm{MB}(\{K, D\}, R_{md} \in Recipes(md), May8.17) \wedge$

(2) (a) Because ma is a single-agent action,
 (1) (a) $\mathrm{Int.To}(K, ma, May8.17, May8.19\text{–}19{:}30, C_{ma}) \wedge$
 (b) $\mathrm{MB}(\{K, D\}, \mathrm{Int.To}(K, ma, May8.17, May8.19\text{–}19{:}30, C_{ma}), May8.17)$
 (3) Because ma is not basic level,
 (a) $(\exists P_{ma}, R_{ma})\mathrm{FIP}(P_{ma}, K, ma, May8.17, May8.19\text{–}19{:}30, R_{ma}, C_{ma}) \wedge$
 (b) $\mathrm{MB}(\{K, D\}, (\exists P_{ma}, R_{ma})$
 (1) $[\mathrm{CBA}(K, ma, R_{ma}, May8.19\text{–}19{:}30, \mathrm{constr}(C_{ma})) \wedge$
 (2) $\mathrm{FIP}(P_{ma}, K, ma, May8.17, May8.19\text{–}19{:}30, R_{ma}, C_{ma})], May8.17) \wedge$
 (4) $\mathrm{MB}(\{K, D\}, \mathrm{Int.Th}(D, (\exists R_{ma})\mathrm{CBA}(K, ma, R_{ma}, May8.19\text{–}19{:}30,$
 (a) $\mathrm{constr}(C_{ma})), May8.17, May8.19\text{–}19{:}30, C_{cba/ma}), May8.17) \wedge$

(a) Because mmc is single-agent,
 (1) (a) $\mathrm{Int.To}(D, mmc, May8.17, May8.19{:}30\text{–}20, C_{mmc})$
 (b) $\mathrm{MB}(\{K, D\}, \mathrm{Int.To}(D, mmc, May8.17, May8.19{:}30\text{–}20, C_{mmc}), May8.17)$
 (3) Because mmc is not basic level,
 (a) $(\exists P_{mmc}, R_{mmc})\mathrm{FIP}(P_{mmc}, D, mmc, May8.17, May8.19{:}30\text{–}20, R_{mmc}, C_{mmc}) \wedge$
 (b) $\mathrm{MB}(\{K, D\}, (\exists P_{mmc}, R_{mmc})$
 (1) $[\mathrm{CBA}(D, mmc, R_{mmc}, May8.19{:}30\text{–}20, \mathrm{constr}(C_{ma})) \wedge$
 (2) $\mathrm{FIP}(P_{mmc}, D, mmc, May8.17, May8.19{:}30\text{–}20, R_{mmc}, \mathrm{constr}(C_{mmc}))],$
 $May8.17) \wedge$
 (4) $\mathrm{MB}(\{K, D\}, \mathrm{Int.Th}(K, (\exists R_{mmc})\mathrm{CBA}(D, mmc, R_{mmc}, May8.19{:}30\text{–}20,$
 $\mathrm{constr}(C_{mmc})), May8.17, May8.19{:}30\text{–}20, C_{cba/mmc}), May8.17) \wedge$

(b) Because gmd is multi-agent (and therefore complex),
 (1) $(\exists P_{gmd}, R_{gmd})\mathrm{FSP}(P_{gmd}, \{K, D\}, gmd, May8.17, Tgmda, R_{gmd}, C_{gmd}) \wedge$
 (2) $\mathrm{MB}(\{K, D\}, (\exists P_{gmd}, R_{gmd})$
 (a) $[\mathrm{CBAG}(\{K, D\}, gmd, R_{gmd}, May8.20\text{–}21, \mathrm{constr}(C_{gmd})) \wedge$
 (b) $\mathrm{FSP}(P_{gmd}, \{K, D\}, gmd, May8.17, May8.20\text{–}21, R_{gmd}, C_{gmd})], May8.17)$

7. Implications of the formalization

The major goal of the formalization presented in this paper is to provide a specification of the mental state of the participants in a collaborative activity that handles complex actions and is comprehensive in its treatment of partiality of belief and intention. The definitions of partial plans provide constraints on agents' beliefs and intentions as they initiate and expand partial plans; they suggest what "snapshots" of the agents' mental state should show during this process. The definitions of complete plans provide the stopping conditions for planning processes. The formalization includes a specification of the minimal mental state requirements needed for the agents to continue to have a plan for collaborative activity. It also ties plans closely to intentions, in particular requiring that an agent have a (partial) plan to do an action when the agent adopts an intention to act.

As promised in the Introduction, the formalization presented in this paper handles more complex relations between actions for both individual and group plans than the original formulation of SharedPlans, provides a means of representing the commitments of agents in collaborations to their group activities, and treats the partiality that naturally arises in most planning situations. Its treatment of individual and collaborative plans is integrated; Lochbaum [43] shows the importance of this integration for modeling collaboration in dialogues. The model also handles contracting out actions. The formalization is more general than alternative approaches in its combined treatment of partiality of recipe and ability knowledge, complex actions with recipes that decompose recursively, and contracting.

The complexities that arose in developing a formalization that handles both complex actions and incremental development of partial plans stem from one of three dimensions along which group activities differ from an individual agent intending to do a basic-level action: commitment to actions, knowledge about how to perform actions, and capabilities to perform actions.

The treatment of commitment becomes more complex both because of partial knowledge and because agents need to form commitments toward others' activities. As a result, we needed to introduce a new type of intention, intending-that a proposition hold, and to define its interactions with intending-to do an action. Finally, to have a reasonable account of intending-to do a complex act in the state of partial recipe knowledge, the formalization provides a treatment of commitment to means–ends reasoning.

Agents' knowledge of how to perform the group activity may be partial and distributed. No single agent may have the complete recipe to all levels of detail for the group activity. As a result, the processes for establishing partial plans and for elaborating them to form complete plans are more complex; they must incorporate capabilities for group decision making and reaching consensus. In addition, agents' assessments of their own and other agents' capabilities to perform actions is more complex.

The representation of agents' states of knowledge about abilities to act is thus also more complex. Were it not for complex actions and partial knowledge, the formalization would require only a single ability predicate. The predicates we define allow us to distinguish among different degrees of recipe knowledge and the corresponding assessment of abilities to act. In particular, they distinguish an agent's assessment when it knows the full recipe (BCBA) from when it knows only a partial recipe (WBCBA), and likewise for groups of agents (MBCBAG and WMBCBAG respectively). The formalization also enables distinguishing what an agent can reasonably know about another agent's abilities from what it knows about its own. Although only one pair of meta-predicates (CBA and CBAG) are needed for the plan definitions themselves, the auxiliary predicates (provided in Appendix A) provide explicit descriptions of the belief and knowledge that is required when agents perform complex actions and may have partial plans and are thus useful for agent design.

Searle [58] and Grosz and Sidner [23] argued that the propositional attitudes of belief, mutual belief, and individual intention to act were insufficient for representing the mental state of participants in collaborative activity. Collaborative plans were more than a simple combination of individual plans. Some means of representing that the actions within a collaborative plan to act were being done in the context of collaboration was needed. Cohen, Levesque, and Nunes [40] subsequently argued that collaborative activity gave rise to communication demands to deal with execution problems and termination of collaboration. Bratman [9] established three criteria that a multi-agent activity must meet to be a "shared cooperative activity": (a) mutual responsiveness; (b) commitment to the joint activity; (c) commitment to mutual support. Furthermore, he argued that as a result of (a) and (b), the agents must form meshing subplans. In the remainder of this section we discuss the way in which our formalization addresses these claims about properties of collaborative or cooperative activity. In the next section, we compare our approach with alternative accounts.

To address the concerns raised by Searle and Grosz and Sidner, the formalization introduces the attitude of intending-that and a context parameter for all intentions; the definitions require intentions-that toward the overall group action and the actions of others. The formalization does not require a separate attitude of joint intention.[33] Although the SharedPlan meta-predicates can be viewed as representing a kind of joint intention, they are not new modal operators. Each reduces to individual intentions plus beliefs and mutual beliefs. The SharedPlan definitions entail that individuals in a collaborating group have certain mental properties; mutual belief ensures that group members all know when they have these attitudes.

To provide other properties, the formalization imposes several constraints on agent design. Some of these are on the agent's mental states and consistency of its beliefs and intentions; others are requirements on agents' planning and communication processes. In particular,

(i) As a minimal constraint on intending to do an action, the formalization requires that the agent will do means–ends reasoning. The definition of Int.To requires that agents not only reconcile intentions, but also have some knowledge about how to do, or to find out how to do, the action. This constraint is essential if agents are to be able to rely on others to do their part in the group activity.

(ii) As a minimal constraint on a group having a plan, the formalization requires that they have agreed about certain decision procedures. In particular, they need procedures for selecting recipes and for identifying agents to perform subactions.

(iii) The formalization requires that groups have procedures for establishing mutual belief and reaching consensus; these form essential components of the plan elaboration processes. Agents must communicate sufficient information for other agents to know what they are able to do and to ascertain what they intend to do. Communication actions result from this constraint.

(iv) The definitions of the ability predicates require that agents compute the context in which they are planning actions.

(v) The formalization specifies that agents cannot hold intentions-to and intentions-that that they believe conflict with each other or one another.

[33] Searle argued that the requisite properties of collaborative activity could not be achieved without such an attitude. Our formalization meets all the criteria he sets.

Bratman's meshing subplans criteria is accomplished in the SharedPlans formalization by the intentions-that each agent forms toward the other group members being able to perform their subactions (Clauses (2a4) and (2b3) of the FSP definition and Clauses (2a4) and (2b5) in the PSP definition); these intentions-that, together with the axioms to avoid conflicting intentions (Fig. 4), ensure meshing subplans. Our formalization treats the case of subplans for multi-agent actions as well as those for individual actions which Bratman discusses.

In discussing mutual responsiveness, Bratman distinguishes between mutual responsiveness of intentions and mutual responsiveness in action. The SharedPlans formalization treats mutual responsiveness of intentions similarly to meshing subplans, through intentions-that. It also handles one aspect of mutual responsiveness in action, namely the responsiveness required when plans must be modified to cope with problems in execution. Its treatment of partial plans provides a basis for interleaving of planning and action. Plans may become more partial as well as more complete; failures lead to increased partiality. Thus, this aspect of responsiveness in action is accommodated by the constraints on processes for elaborating partial plans (cf. discussion below of communication obligations). That aspect of mutual responsiveness in action which has to do with continual monitoring of another agent's actions is beyond the scope of this paper. However, we believe it should be treated as part of the interleaving of planning and acting rather than by making a sharp distinction between planning and execution.

Commitment to the joint activity (Bratman's second criterion) is directly represented by the Int.Th in Clause (0) of the FSP's definition. Commitment to mutual support (Bratman's third criterion) is realized in a more complex way. It requires a combination of the intentions-that the agents form and the axioms for helpful behavior that originate from intentions-that.

Each of the three basic roles of intention that Bratman describes in earlier work [7] also plays a significant role in the formalization. That an intention-to engenders means-ends reasoning is built into the definition of Int.To; the FIP to Elaborate_Individual represents this commitment. Conflict avoidance is also explicitly represented, in Axiom (A1) of Fig. 4. The use of intentions in replanning is only implicit; the context parameter that is associated with each plan encodes the reason for doing the plan's action and thus is available for use in replanning.

Other researchers [12,29,40,65] have discussed the communication obligations that arise from failure (or success) in performing the actions a cooperative or collaborative activity comprises. As described above, the plan elaboration processes required by SharedPlans entail additional communication requirements. Intentions-that play a central role in the way our formalization addresses these communication obligations. We examine briefly their use in dealing with communication requirements stemming from action failures, intention reconciliation decisions, and resource conflicts.

First, we consider the situation in which a member G_k of a collaborating group finds itself unable to successfully complete an action β_k that it initially agreed to do. This might occur for several reasons. For example, G_k might have failed to successfully carry out some of the subactions in its recipe for β_k. Alternatively, in the process of reconciling some other intention, it might have decided to drop its Int.To do β_k. Once this occurs G_k no longer has an intention or a full individual plan to do β_k. G_k knows that the group's shared plan is now less complete. Indeed if the group had a FSP it no longer does. But, until G_k communicates with the other group members, they will not know.

Intentions-that in the SharedPlan will lead G_k to communicate this changed status as follows. G_k still has an intention-that of the form in Clause (0) of PSP (or FSP). Because G_k has dropped an intention to do a subaction, there is now a proposition embedded in this intention-that that is false (i.e., the Do-proposition). As a result of this false proposition and the Clause (0) intention-that, Axiom (A5) for intending-that (Fig. 23) will yield G_k's having a potential intention to do any actions it believes will contribute toward making the proposition true. If G_k believes that communicating its failure to the group will help re-establish the future performance of the action (and hence make the Do-proposition true), it will attempt to communicate. Hence, so long as G_k maintains its intention-that the group do the activity of which β_k is a constituent all is well.

A different situation arises if G_k has reason to drop the Clause (0) intention-that and opt out of the collaborative plan. If G_k has some other intention-that, one which obligates it to some or all members of the group, then that intention-that will, in a manner similar to that just described, lead G_k to communicate this fact to the member(s). For instance, if G_k has an intention-that the other agents will want to agree to collaborate with it at some later time, and a belief that opting out without explanation will cause this not to be the case, then G_k will adopt a potential intention to explain its opting out of this collaboration. Thus, whether or not an agent communicates when it opts out depends on its other intentions (Int.Th's) and its beliefs. Some people communicate in such situations; others do not. We can design computer agents to exhibit either behavior.

Finally, our plan definitions and axioms also entail that an agent communicate when it recognizes a resource conflict. If one agent G_1 intends-that another agent G_2 be able to perform an action β and G_1 detects a resource conflict affecting G_2's ability to do β, then the intentions-that axioms will lead G_1 to consider doing all of the actions it can to prevent this resource conflict. Typically, the act of informing G_2 and discussing the problem will help resolve the problem. Again, one implication of deciding not to take action is that there is no longer a SharedPlan. And again, this constraint provides for a range of agent designs, depending on the tradeoffs between extra communication costs and the costs of error recovery.

8. Comparison with alternative approaches

In this section, we compare our formalization to three alternative approaches to specifying cooperative activity, those of Cohen, Levesque, and Nunes [11,12,40] Sonenberg et al. [61], and Jennings [29].

Cohen, Levesque and Nunes (henceforth, CLN) study notions of joint commitment (represented by a modal operator for joint persistent goal, JPG) and joint intentions (JI) and the ways in which they relate to individual commitments of group members;

they address in particular the need for agents to inform one another whenever they drop a joint commitment.

Although not defined in terms of individual intentions, joint intentions entail individual intentions and commitments. For both individual and joint intentions, CLN employ a strong notion of commitment. JPG stipulates that agents are committed to their joint goal until they believe the goal is satisfied or come to the conclusion that it is impossible to reach the goal. The individual persistent goal, PGOAL, has a similar constraint. In both cases, CLN also allow for a commitment to be relative to a formula q that can serve as an "escape clause"; agents may drop their commitments if they believe q is false. The definitions of joint and individual intentions utilize JPG and PGOAL, and thus intentions entail commitment in this strong sense.

CLN's model also embodies a strong and rigid communication requirement. According to the definition of JPG, if a participant in a collaboration comes to believe privately that the joint goal is satisfied or is impossible to achieve, it incurs a commitment to make this mutually known (i.e., to inform the other agents). That is, an agent may not abandon a joint commitment without communicating with its partners.

Our formalization allows for more flexible behavior. Agents reconcile their intentions when considering adopting new intentions. They may decide to drop intentions for a variety of reasons (e.g., resource limitations). Although communicating with partners in a collaboration plays an important role in the Elaborate_Group process, agents are not required to communicate when they drop intentions. Instead, communication is only one option in such situations. As was explained in the previous section, the intentions-that and mutual belief components of the SharedPlan definitions yield the commitment properties CLN argue for in the case that the SharedPlan is maintained; however, we also allow the possibility of agents opting out of collaborations.

CLN consider actions and plans only at a high level of abstraction and do not address partiality in a significant way. They do not discuss or represent in detail partial plans for individual or joint action. In two papers [11,40], only actions with all constituent subactions specified are considered. A subsequent paper [12] allows some partiality in the definition of individual and joint intention by allowing the operators INTEND* and IJ* to take open action expressions as arguments.

In contrast, we examine in detail various types of partiality. As described in Section 7, doing so complicates our definitions. However, it also yields detailed specification of the possible mental states of agents that have partial individual plans or SharedPlans. As a result, our definitions provide more precise constraints on the design of elaboration processes. Our formalization also includes more detailed specification of recipes for actions, the context of planning and acting, and the information that collaborating agents must have about their joint activity. Because CLN do not provide detailed specification of subactions of complex actions, they also do not consider issues related to the processes of recipe selection and subaction allocation for participants and subgroups. In addition, they do not discuss in detail the ways in which helpful behavior is generated.

Sonenberg et al. [61] (henceforth STWKLR) do provide detailed specifications of the various plan constructs that arise in modeling collaborative activity. Their work differs from ours most in that they do not handle partial plans. In addition, they typically provide a single mechanism in cases in which we give general restrictions on agents' mental states and their elaboration processes. Although Sonenberg et al. have more detailed semantics than we do, they also do not address issues of soundness and completeness.

As in our framework, STWKLR assume that each individual agent is supplied with a set of plan structures (similar to our recipes) known as its plan library. The plan library of a group of agents is the intersection of the plan libraries of its participants. However, they do not consider situations in which agents have partial plans. Their formalization does not allow for a group to construct a new plan structure using elements from individual members' libraries, as we do with recipes. It also does not allow for subcontracting of actions to agents outside the group.

In place of our ability meta-predicates (Section 4.4 and Appendix A), STWKLR define the notion of *skills* of teams. [34] The skills of a team are the set of primitive actions that can be performed by that team; team skills include skills of the individuals the team comprises and skills of the composite. These skills are statically determined; the formalization does not allow for them to vary with time nor to depend on the situation of use. Thus, their agents can reason in what STWKLR call compile-time [35] about the potential of a given team to successfully execute a plan [36] (i.e., perform a complex action) and compute the groups that might succeed in achieving a goal. This information can be used to guide the process of team formation at run-time. STWKLR provide a detailed algorithm for team formation that depends on centralized control. They also provide a general mechanism for agents to synchronize their group activity, an issue we do not address.

The ability meta-predicates we define take into account the context in which an agent is performing an action. They represent an agent's ability to perform an action using a particular recipe at a given time and under constraints. The formalization considers constraints from the context in which the action is being performed (e.g., the complex action of which this action is a constituent) as well as the recipe. In the group ability predicates (e.g., CBAG), agents' beliefs about the capabilities of other agents also play a role. [37] Thus our definitions provide more flexibility in plan formation.

STWKLR, like CLN, build into their definition of joint intention a requirement that agents communicate. In particular, the method they use for transforming a general plan structure to a plan in which teams are assigned to specific actions (i.e., a role plan in their terminology) adds communication actions so that a message reporting the failure or success of a subaction is broadcast to the members of the group after the subaction is executed (or, attempted, in the case of failure).

---

[34] In their formalization an individual agent is also a team.

[35] They call the time prior to engaging in time-bounded activity *compile-time* and the time of collaborative real-time activity *run-time*. We do not distinguish between these time periods.

[36] STWKLR's terminology differs from ours. In particular, their "plan" corresponds approximately to our "recipe"; executing a plan is executing a particular instantiation of a recipe.

[37] Although STWKLR briefly mention a notion of capabilities that takes into account not only skills, but also compatibility with other goals and intentions and constraints on recipes, their detailed discussion and algorithm refer only to skills.

STWKLR argue that their formalization meets Bratman's conditions for shared cooperative activity. They treat mutual responsiveness by providing for a team to backtrack and choose a different subaction or a different role assignment if either is available. However, it appears they can ensure meshing subplans only insofar as the meshing is encoded into plan structures in the agents' libraries. They do not discuss either mutual support or helpful behavior.

Jennings [29] used a formal model for joint intention in the design of a testbed environment for constructing cooperative multi-agent systems. He tested this design and the role of explicit models of intention by implementing a system for electricity transportation management. The formal specification in modal logic was used to develop a system of production rules that yielded agents' behaving according to the formalization. The transformation from formal model to agent design is thus a major contribution of this work.

Jennings' formal responsibility model is a refinement of the Cohen, Levesque, and Nunes [40] formalization of joint intentions. It adds two elements to this formalization: recipe selection and a notion of joint recipe commitment, including a specification of the conditions under which joint recipe commitment can be dropped. Although the responsibility model treats recipes for complex group actions, these decompose immediately into single-agent actions. It appears that the constituent single-agent actions may be complex, and that the agent may form a team for a joint plan to assist in performing the action, but there is no connection in the formalization between the team and plan for this subsidiary action and the original one.

Jennings's system handles task allocation by having a central organizer that uses information it has about the abilities of all other agents to assign tasks. Recipe selection for the complex group action is also managed by this central organizer. The team members select their own recipes for the single-agent constituent actions; there is no description of which information about an individual recipe needs to be communicated to others. Recipes are partial only in allowing delayed specification of the agent and time of action; there is no partiality of the decomposition into constituent actions.

A major emphasis of this work is an examination of the role of communication in assisting collaboration. In particular, Jennings ran several experiments comparing the performance of a system based on the responsibility model with alternatives that did not utilize an explicit model of joint-intention. These experiments focused on the execution phase of collaborative activity and the benefits of communicating to the group when an agent cannot carry out actions it committed to doing as part of the group activity. The results suggest that communication lessens wasted work. Jennings argues further that the explicit model of collaboration enables this savings without a system designer needing to determine in advance all of the problems that might arise during execution and the information that needs to be communicated once one of them does.

9. Conclusions and future work

To provide an account of collaborative activity, Searle [58] introduced the notion of "we-intention". Grosz and Sidner [23] argued that such a notion should not be necessary and their initial formulation of SharedPlans avoids use of one. However, the definitions provided in that formulation could only accommodate group activity that directly decomposed into actions of individual agents. In this paper, we have provided a formalization that handles complex actions and that allows for plans to be partial. In this work, SharedPlans serve two major roles. They summarize the set of beliefs and intentions needed for collaborative activity, and they provide the rationale for the process of revising beliefs and intentions. Consequently, they motivate the collaborative correlate of means-ends reasoning in the plans of an individual agent. SharedPlans ground out in the individual intentions of individual agents and the individual plans that they engender. Our formalization accommodates the properties of shared cooperative activity proposed by Bratman [9]. Intentions to do constituent actions form the basis of each individual's actions. Intentions-that directed toward other agents' abilities to act and success in acting, as well as toward the success of the joint activity, ensure meshing subplans and helpful behavior.

The development of this formalization uncovered several interesting new problems in designing agents for collaborative work. These include:

(i) The need to develop more flexible methods for reasoning about resources and resource conflicts in the context of collaborative activity, and to examine the tradeoffs among them.

(ii) The need to develop more complex methods for groups to construct and agree on recipes.

(iii) The need to understand more fully the ways in which communication supports collaboration and to develop a more complete set of communication axioms. Agents need to communicate about more than the completion of subtasks or errors, the two situations for which alternative approaches (cf. Section 8) have built in the need to communicate. For instance, communication actions play a central role in establishing requisite mutual beliefs and ensuring the satisfaction of intentions-that.

The formalization underspecifies several aspects of collaborative planning. First, the plan definitions entail that the reasoning mechanisms individual agents utilize for elaborating partial plans have certain properties, but a complete specification of these elaboration processes has not yet been provided. Second, the formalization includes only the most basic axioms for the attitude of intending-that (i.e., the Int.Th modal operator). Third, we do not provide specific mechanisms for reconciling intentions. In addition, we do not formalize commitment in this paper. This issue is enormous in its own right; it is most relevant to SharedPlans in the way it affects the development of procedures for reconciling intentions.

Each of these poses a significant research problem which we have left to future research. The major next steps we envision are to develop strategies and protocols for elaborating partial plans, including mechanisms for combining information possessed by different agents about how to perform a complex action, and strategies for negotiating among competing approaches, handling resource conflicts, and reaching consensus on how to allocate portions of the activity among different participants.

Acknowledgements

This paper is an extension of our paper in IJCAI-93 [21]. We thank Joyce Friedman for many thought-provoking questions, and Karen Lochbaum for the same and for helpful comments on many drafts. We also thank Michael Bratman, David Israel, and Martha Pollack for comments on earlier versions, and the anonymous reviewers for their helpful suggestions and illuminating comments. This research was initiated when the first author was a Harold Perlman Visiting Professor, Hebrew University, Jerusalem. Partial support for the first author was provided by US WEST Advanced Technologies and by NSF Grant Number IRI-9525915. The second author was supported in part by NSF Grant Number IRI-9423967 and the Israeli Science Ministry Grant Number 6288.

Appendix A. Abilities and beliefs

This section contains the formal definitions of the meta-predicates BCBA, WBCBA, MBCBAG and WMBCBAG representing different kinds of belief in an agent's ability to carry out actions and related theorems.

A.1. *Proof of Theorem* (T2)

To prove Theorem (T2), we formally define the concept of an *extended recipe tree*. As described in Section 4.1, this concept is an extension of a recipe tree that takes into account contracting actions. Without loss of generality, we assume that a recipe consists of at least two actions.

Theorem (T2):

$(\forall \alpha, T_\alpha, T_p, G, R_\alpha, P)$
$[\mathrm{FIP}(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha) \Rightarrow$
$\mathrm{BCBA}(G, \alpha, R_\alpha, T_\alpha, T_p, \mathrm{constr}(C_\alpha))]$

$\mathrm{BCBA}(G, \alpha, R_\alpha, T_\alpha, T_{bel}, \Theta)$

(1) $[\mathrm{basic.level}(\alpha) \wedge \mathrm{Bel}(G, \mathrm{Exec}(G, \alpha, T_\alpha, \Theta), T_{bel})] \otimes$

(2) $[\neg \mathrm{basic.level}(\alpha) \wedge$

(a) $R_\alpha = \{\beta_i, \rho_j\} \wedge \mathrm{Bel}(G, R_\alpha \in \mathit{Recipes}(\alpha), T_{bel}) \wedge$

(b) $(\forall \beta_i \in R_\alpha, \exists T_{\beta_i})$

(1) $[[(\exists R_{\beta_i})\mathrm{BCBA}(G, \beta_i, R_{\beta_i}, T_{\beta_i}, T_{bel}, \Theta \cup \{\rho_j\})] \vee$

(2) $[(\exists G_c, \gamma, T_\gamma, R_\gamma)[$

(a) $\mathrm{BCBA}(G, \gamma, R_\gamma, T_\gamma, T_{bel}, \Theta \cup \{\rho_j\}) \wedge$

(b) $\mathrm{Bel}(G, \mathrm{GTD}(G, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]]]]$

Fig. A.1. Beliefs and capabilities to perform actions in FIP and the definition of BCBA (believe can bring about) and related theorem.

Definition of Extended Recipe Tree. If α is a basic-level action, and Int.To$(G, \alpha, T_i, T_\alpha, C_\alpha)$ holds, then α's extended recipe tree with respect to that intention is a tree with one node labeled by α.

Suppose α is a complex action such that

$$(\exists R_\alpha, P)\text{FIP}(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$$

holds where $R_\alpha = \{\beta_i, \rho_j\}$.

An extended recipe tree for α, *TR*, with respect to the full individual plan[38] FIP$(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$ is a tree that satisfies the following conditions:

- The root *TR* is labeled by α.
- If according to P, G intends to do β_i by itself, then
  - if β_i is a basic-level action, the ith child of the root of *TR* is the extended recipe tree for β_i with respect to Int.To$(G, \beta_i, T_p, T_{\beta_i}, C_{\beta_i/\alpha})$;
  - otherwise, the ith child of the root of *TR* is the extended recipe tree for β_i with respect to FIP$(P_{\beta_i}, G, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})$.
- If G subcontracts β_i according to P, then the ith child of the root of *TR* is the root of a subtree that has its root labeled by β_i and the child of this β_i node is the root of the extended tree for γ with respect to Int.To$(G, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha})$ if γ is a basic-level action, and with respect to FIP$(P_\gamma, G, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta_i/\alpha})$ otherwise.

Proof of Theorem (T2). Suppose FIP$(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$ holds for some R_α, P, α, G, T_p, T_α and C_α. If G has a FIP for α then α is a complex action. However, it is important to observe that from the definition of Int.To (Fig. 3) and the definition of BCBA, if an agent intends to do a basic-level action α, BCBA holds for α, i.e.,

$$\begin{aligned} &\text{basic.level}(\alpha) \wedge \text{Int.To}(G, \alpha, T_i, T_\alpha, C_\alpha) \\ &\quad \Rightarrow \text{BCBA}(G, \alpha, R_{Empty}, T_\alpha, T_i, \text{constr}(C_\alpha)). \end{aligned} \tag{A.1}$$

We now prove (T2) by induction on the height of the extended recipe tree of α with respect to FIP$(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$. Since α is a complex action, the height of the extended recipe tree for α is at least 1.

Base case: If the height of the extended recipe tree for α with respect to FIP$(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$ is 1 and $R_\alpha = \{\beta_i, \rho_j\}$, then all β_i are basic-level actions and G intends to do each of the β_i by itself.

From Clause (1) of the FIP definition, Bel$(G, R_\alpha \in \textit{Recipes}(\alpha), T_p)$ and therefore Clause (2a) of the BCBA definition holds. Since all the subactions in R_α are basic-level actions and G intends to do each subaction by itself, Clause (2b) of BCBA follows from Clause (2a) of the FIP definition and (A.1) above. Thus, we can conclude that

$$\text{BCBA}(G, \alpha, R_\alpha, T_\alpha, T_p, \text{constr}(C_\alpha)).$$

[38] We will drop the FIP when it is clear from the context.

Induction case: Suppose (T2) holds when the height of the extended recipe tree is less than k. We consider the case in which the height of the extended recipe tree with respect to the full individual plan is k.

As in the base case, Clause (2a) of the BCBA is established from Clause (1) of the FIP definition. For any $\beta_i \in R_\alpha$ we need to show that Clause (2) holds.

- If, according to plan P, G intends to do β_i by itself, then
  - If β_i is a basic-level action, then the proof that Clause (2b1) of BCBA holds proceeds as in the base case.
  - If β_i is a complex action, then Clause (2b1) of FIP specifies that

    $$(\exists P_{\beta_i}, R_{\beta_i})\text{FIP}(P_{\beta_i}, G, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha}),$$

    where $\text{constr}(C_{\beta_i/\alpha}) \supseteq \text{constr}(C_\alpha) \cup \{\rho_j\}$; the height of the extended recipe tree of β_i with respect to $\text{FIP}(P_{\beta_i}, G, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})$ is less than k (since

---

Theorem (T3):

$(\forall \alpha, T_\alpha, T_p, G, P)$

$[\text{PIP}(P, G, \alpha, T_p, T_\alpha, C_\alpha) \Rightarrow$

$(\exists R^p_\alpha)\text{WBCBA}(G, \alpha, T_\alpha, R^p_\alpha, T_{bel}, \text{constr}(C_\alpha))]$

$\text{WBCBA}(G, \alpha, T_\alpha, R^p_\alpha, T_{bel}, \Theta)$

(0) $[\text{basic.level}(\alpha) \wedge \text{Bel}(G, \text{Exec}(G, \alpha, T_\alpha, \Theta), T_{bel})] \otimes$

(a) $[\neg\text{basic.level}(\alpha) \wedge R^p_\alpha = \{\beta_i, \rho_j\} \wedge$

(1) $\text{Bel}(G, (\exists R_\alpha \in \mathit{Recipes}(\alpha))\{\beta_i, \rho_j\} \subseteq R_\alpha, T_{bel}) \wedge$

(a) $[\text{Bel}(G, (\exists R_\alpha)[R_\alpha \in \mathit{Recipes}(\alpha) \wedge \{\beta_i, \rho_j\} \subset R_\alpha], T_p) \Rightarrow [$

(b) $(\exists T_{select.rec}, R_{select.rec})$

(1) $\text{BCBA}(G, \text{Select\_Rec}(G, \alpha, \{\beta_i, \rho_j\}, T_{select.rec}), R_{select.rec}, T_{select.rec}, T_{bel},$

(a) $(\exists R_\alpha = \{\delta_r, \kappa_e\})$

(b) $[\text{Bel}(G, R_\alpha \in \mathit{Recipes}(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha, T_{select.rec}) \wedge$

(c) $(\forall \delta_r \in R_\alpha)$

(d) $\text{Bel}(G, (\exists T_{\delta_r}, R_{\delta_r})\text{CBA}(G, \delta_r, R_{\delta_r}, T_{\delta_r}, \Theta \cup \{\kappa_e\}) \vee$

(e) $(\exists T_{\delta_r})\text{CC}(G, \delta_r, T_{\delta_r}, \Theta), T_{select.rec})])]] \wedge$

$(\forall \beta_i \in R^p_\alpha, \exists T_{\beta_i})[$

(2) $(\exists R^p_{\beta_i})\text{WBCBA}(G, \beta_i, T_{\beta_i}, R^p_{\beta_i}, T_{bel}, \Theta \cup \{\rho_j\}) \vee$

(3) $(\exists G_c, \gamma, T_\gamma, R^p_\gamma)$

(a) $[\text{Bel}(G, \text{GTD}(G, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel}) \wedge$

(b) $\text{WBCBA}(G, \gamma, T_\gamma, R^p_\gamma, T_{bel}, \Theta \cup \{\rho_j\})] \vee$

(4) (a) $\text{Bel}(G, (\exists R_{\beta_i})\text{CBA}(G, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel}) \vee$

(b) $(\exists T_{\beta_i}, \gamma, T_\gamma, G_c)$

(1) $[\text{Bel}(G, (\exists R_\gamma)\text{CBA}(G, \gamma, R_\gamma, T_\gamma, \Theta \cup \{\rho_j\}) \wedge$

(a) $\text{GTD}(G, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]]$

Fig. A.2. Beliefs and capabilities to perform actions in PIP and the definition of WBCBA (weakly believe can bring about).

this extended tree is a subtree of the extended tree for α). By the induction hypothesis, we can conclude that

$$(\exists R_{\beta_i})\mathrm{BCBA}(G, \beta_i, R_{\beta_i}, T_{\beta_i}, T_i, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}).$$

- If G contracts out β_i, then Clause (2b2b) of BCBA follows from Clause (3a) of the FIP definition. Since the height of the extended recipe tree for γ is less than k (it is a subtree of the extended recipe tree for α), the proof that Clause (2b2a) of BCBA holds is similar to the proof that Clause (2b1) holds when G intends to do β_i by itself.

Thus, we can conclude that $\mathrm{BCBA}(G, \alpha, R_\alpha, T_\alpha, T_p, \mathrm{constr}(C_\alpha))$. □

Theorem (T4):

$(\forall \alpha, T_\alpha, T_p, \mathrm{GR})$
$[(\exists R_\alpha, P)\mathrm{FSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha) \Rightarrow$
$(\exists R_\alpha)\mathrm{MBCBAG}(\mathrm{GR}, \alpha, R_\alpha, T_\alpha, T_p, \mathrm{constr}(C_\alpha))]$

$\mathrm{MBCBAG}(\mathrm{GR}, \alpha, R_\alpha, T_\alpha, T_{bel}, \Theta)$

(1) $R = \{\beta_i, \rho_j\} \wedge \mathrm{MB}(\mathrm{GR}, R_\alpha \in \mathit{Recipes}(\alpha), T_{bel}) \wedge$
(b) $(\forall \beta_i \in R_\alpha, \exists T_{\beta_i})[$

(2) $[[\mathrm{single.agent}(\beta_i) \wedge$
(a) $(\exists G_{\beta_i} \in \mathrm{GR})[$
(1) $(\exists R_{\beta_i})\mathrm{BCBA}(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, T_{bel}, \Theta \cup \{\rho_j\}) \wedge$
(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_i})\mathrm{CBA}(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{b_i}, \Theta \cup \{\rho_j\}), T_{bel})]]\otimes$

(3) $[\mathrm{multi.agent}(\beta_i) \wedge$
(a) $(\exists \mathrm{GR}_{\beta_i} \subseteq \mathrm{GR})[$
(1) $(\exists R_{\beta_i})\mathrm{MBCBAG}(\mathrm{GR}_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, T_{bel}, \Theta \cup \{\rho_j\}) \wedge$
(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_i}), \mathrm{CBAG}(\mathrm{GR}_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]]] \vee$

(4) $[$
(a) $(\exists G_c, \gamma)$ $[\mathrm{single.agent}(\gamma) \wedge$
(1) $(\exists G_\gamma \in \mathrm{GR}, T_\gamma)[$
(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_\gamma), \mathrm{CBA}(G_\gamma, \gamma, R_\gamma, T_\gamma, T_{bel}, \Theta \cup \{\rho_j\}), T_{bel}) \wedge$
(3) $\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(G_\gamma, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel}))$
(4) $(\exists R_\gamma)\mathrm{BCBA}(G_\gamma, \gamma, R_\gamma, T_\gamma, T_{bel}, \Theta \cup \{\rho_j\})] \vee$
(b) $(\exists G_c, \gamma)$ $[\mathrm{multi.agent}(\gamma) \wedge$
(1) $(\exists \mathrm{GR}_\gamma \subseteq \mathrm{GR}, T_\gamma)[$
(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_\gamma)\mathrm{CBAG}(\mathrm{GR}_\gamma, \gamma, R_\gamma, T_\gamma, T_{bel}, \Theta \cup \{\rho_j\}), T_{bel}) \wedge$
(3) $\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(G_\gamma, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})) \wedge$
(4) $(\exists R_\gamma)\mathrm{MBCBAG}(\mathrm{GR}_\gamma, \gamma, R_\gamma, T_\gamma, T_{bel}, \Theta \cup \{\rho_j\})]]]$

Fig. A.3. Beliefs and capabilities to perform actions in FSP and the definition of MBCBAG (mutually believe can bring about group).

Theorem (T5):

$(\forall\alpha, T_\alpha, T_p, \mathrm{GR})$

$[(\exists P)\mathrm{PSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha) \Rightarrow$

$(\exists R^p_\alpha)\mathrm{WMBCBAG}(\mathrm{GR}, \alpha, T_\alpha, R^p_\alpha, T_p, \Theta)]$

$\mathrm{WMBCBAG}(\mathrm{GR}, \alpha, T_\alpha, R^p_\alpha, T_{bel}, \Theta)$

(1) $R^p_\alpha = \{\beta_i, \rho_j\} \wedge \mathrm{MB}(\mathrm{GR}, (\exists R_\alpha \in \mathit{Recipes}(\alpha))\{\beta_i, \rho_j\} \subseteq R_\alpha, T_{bel}) \wedge$

(a) $[\mathrm{MB}(\mathrm{GR}, (\exists R_\alpha)[\{\beta_i, \rho_j\} \subset R_\alpha \wedge R_\alpha \in \mathit{Recipes}(\alpha)], T_{bel}) \Rightarrow [$

(b) $(\exists T_{select.rec.g}, P_{select.rec.g}, R_{select.rec.g})$

(1) $\mathrm{MBCBAG}(\mathrm{GR}, \mathrm{Select\_Rec\_GR}(\mathrm{GR}, \alpha, \{\beta_i, \rho_j\}, T_{select.rec.g}), R_{select.rec.g}, T_{select.rec.g}, T_{bel},$

(a) $(\exists R_\alpha = \{\delta_r, \kappa_e\})[$

(b) $\mathrm{MB}(\mathrm{GR}, R_\alpha \in \mathit{Recipes}(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha, T_{select.rec.g}) \wedge$

(c) $(\forall\delta_r \in R_\alpha)\mathrm{MB}(\mathrm{GR}, (\exists T_{\delta_r})[$

(d) $[[\mathrm{single.agent}(\delta_r) \wedge$

(e) $(\exists G_{\delta_r} \in \mathrm{GR}, R_{\delta_r})\mathrm{CBA}(G_{\delta_r}, \delta_r, R_{\delta_r}, T_{\delta_r}, \Theta \cup \{\kappa_e\})]\otimes$

(f) $[\mathrm{multi.agent}(\delta_r) \wedge$

(g) $(\exists \mathrm{GR}_{\delta_r} \subseteq \mathrm{GR}, R_{\delta_r})\mathrm{CBAG}(\mathrm{GR}_{\delta_r}, \delta_r, R_{\delta_r}, T_{\delta_r}, \Theta \cup \{\kappa_e\})]] \bigvee$

(h) $[\mathrm{CCG}(\mathrm{GR}, \delta_r, T_{\delta_r}, \Theta \cup \{\rho_l\})]], T_{select.rec.g})])] \wedge$

$(\forall\beta_i \in R^p_\alpha, \exists T_{\beta_i})[$

(2) (a) $[[\mathrm{single.agent}(\beta_i) \wedge (\exists G_k \in \mathrm{GR})$

(1) $[(\exists R^p_{\beta_i})\mathrm{WBCBA}(G, \beta_i, T_{\beta_i}, R^p_{\beta_i}, T_{bel}, \Theta \cup \{\rho_j\}) \wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_r})\mathrm{CBA}(G_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]]\otimes$

(b) $[\mathrm{multi.agent}(\beta_i) \wedge (\exists \mathrm{GR}_k \subset GR)$

(1) $[(\exists R^p_{\beta_i})\mathrm{WMBCBAG}(\mathrm{GR}_k, \beta_i, T_{\beta_i}, R^p_{\beta_i}, T_{bel}, \Theta \cup \{\rho_j\}) \wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_r})\mathrm{CBAG}(\mathrm{GR}_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]]] \vee$

(3) $[\mathrm{WMBCC}(\mathrm{GR}, \beta_i, T_{\beta_i}, T_{\beta_i} \cup \{\rho_j\})] \wedge$

(4) (a) (1) $[[\mathrm{single.agent}(\beta_i) \wedge$

(a) $\mathrm{MB}(\mathrm{GR}, (\exists G_k \in \mathrm{GR}, R_{\beta_i}, T_{\beta_i})\mathrm{CBA}(G_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]\otimes$

(2) $[\mathrm{multi.agent}(\beta_i) \wedge$

(a) $\mathrm{MB}(\mathrm{GR}, (\exists \mathrm{GR}_k \subseteq \mathrm{GR}, R_{\beta_i}, T_{\beta_i})\mathrm{CBAG}(\mathrm{GR}_k, \beta_i, R_{\beta_i}, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]] \vee$

(b) (1) $[\mathrm{MB}(\mathrm{GR}, (\exists T_{\beta_i})\mathrm{CCG}(\mathrm{GR}, \beta_i, T_{\beta_i}, \Theta \cup \{\rho_j\}), T_{bel})]]$

{COMMENT: WMBCC includes all the belief and mutual belief required for contracting out β_i when the group has PSP. }

Fig. A.4. Beliefs and capabilities to perform actions in PSP and the definition of WMBCBAG (weakly mutually believe can bring about group).

Appendix B. Formal plan definitions and additional intention-that axiom

This appendix contains the full formal definitions for the individual plan meta-predicates FIP and PIP and the group plan meta-predicates FSP and PSP.

$\mathrm{FIP}(P, G, \alpha, T_p, T_\alpha, R_\alpha, C_\alpha)$

(1) $R_\alpha = \{\beta_i, \rho_j\} \wedge \mathrm{Bel}(G, R_\alpha \in \mathit{Recipes}(\alpha), T_p) \wedge$

$(\forall \beta_i \exists T_{\beta_i})[$

(2) (a) $[\mathrm{Int.To}(G, \beta_i, T_p, T_{\beta_i}, C_{\beta_i/\alpha}) \wedge$

(b) $\neg\mathrm{basic.level}(\beta_i) \Rightarrow$

(1) $(\exists P_{\beta_i}, R_{\beta_i})\mathrm{FIP}(P_{\beta_i}, G, \beta_i, T_p, T_{\beta_i}, R_{\beta_i}, C_{\beta_i/\alpha})] \otimes$

(3) $[\exists(G_c, \gamma, T_\gamma)$

(a) $[\mathrm{Bel}(G, \mathrm{GTD}(G, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(b) $\mathrm{Int.To}(G, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}) \wedge$

(c) $\mathrm{Int.Th}(G, (\exists R_{\beta_i})\mathrm{CBA}(G_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_i}, C_{cba/\beta_i/\alpha}) \wedge$

(d) $\neg\mathrm{basic.level}(\gamma) \Rightarrow$

(1) $(\exists R_\gamma, P_\gamma)\mathrm{FIP}(P_\gamma, G, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta_i/\alpha})]]]\}$

{COMMENT: $\mathrm{constr}(C_{\beta_i/\alpha}) = \mathrm{constr}(C_\alpha) \cup \{\rho_j\}$,
$\mathrm{constr}(C_{\gamma/\beta_i/\alpha}) = \mathrm{constr}(C_\alpha) \cup \{\rho_j\}$.}

Fig. B.1. Definition of full individual plan.

$\mathrm{PIP}(P, G, \alpha, T_p, T_\alpha, C_\alpha)$

$(\exists\{\beta_i, \rho_j\})[$

(1) $\mathrm{Bel}(G, (\exists R_\alpha)[R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha], T_p) \wedge$

(a) $[\mathrm{Bel}(G, (\exists R_\alpha)[R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subset R_\alpha], T_p) \Rightarrow [$

(b) $(\exists T_{select.rec}, P_{select.rec}, R_{select.rec})$

(1) $[\mathrm{Int.To}(G, \mathrm{Select\_Rec}(G, \alpha, \{\beta_i, \rho_j\}, T_{select.rec}), T_p, T_{select.rec}, C_{select.rec/\alpha}) \wedge$

(2) $\mathrm{FIP}(P_{select.rec}, G, \mathrm{Select\_Rec}(G, \alpha, \{\beta_i, \rho_j\}, T_{select.rec}), T_p, T_{select.rec},$

$R_{select.rec}, C_{select.rec/\alpha})]]] \bigwedge$

$set(\beta_i) = (set[\beta_k] \cup set[\beta_r]) \wedge (\forall \beta_r \in set[\beta_r][\exists T_{\beta_r})[$

(2) $[\mathrm{Int.To}(G, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha})] \bigotimes$

(3) $[(\exists G_c, \gamma, T_\gamma)$

(a) $[\mathrm{Bel}(G, \mathrm{GTD}(G, \gamma, T_\gamma, G_c, \beta_r, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(b) $\mathrm{Int.To}(G, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha}) \wedge$

(c) $\mathrm{Int.Th}(G, (\exists R_{\beta_r})\mathrm{CBA}(G_c, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_r}, C_{cba/\beta_r/\alpha})]]] \bigwedge$

(4) $(\forall \beta_k \in set[\beta_k])[$

(a) (1) $[(\exists T_{\beta_k})[\mathrm{Pot.Int.To}(G, \beta_k, T_p, T_{\beta_k}, C_{\beta_k/\alpha}) \wedge$

(2) $\mathrm{Bel}(G, (\exists R_{\beta_k})\mathrm{CBA}(G, \beta_k, R_{\beta_k}, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)] \vee$

(b) (1) $[(\exists T_{\beta_k}, \gamma, T_\gamma)[\mathrm{Pot.Int.To}(G, \gamma, T_p, T_\gamma, C_{\beta_k/f(\beta_k,\gamma)/\alpha}) \wedge$

(2) $\mathrm{Bel}(G, (\exists G_c, R_\gamma)\mathrm{CBA}(G, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}) \wedge$

(a) $\mathrm{GTD}(G, \gamma, T_\gamma, G_c, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)]]]]]$

{COMMENT:

$\mathrm{constr}(C_{select.rec/\alpha}) \supseteq$

$\exists R_\alpha = \{\delta_e, \kappa_e\})[\mathrm{Bel}(G, R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha, T_{select.rec}) \wedge$

$(\forall \delta_e \in R_\alpha)$

$\mathrm{Bel}(G, (\exists T_{\delta_e})[(\exists R_{\delta_e})\mathrm{CBA}(G, \delta_e, T_{\delta_e}, R_{\delta_e}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\}) \vee$

$\mathrm{CC}(G, \delta_e, T_{\delta_e}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\})], T_{select.rec})]$

$\mathrm{constr}(C_{\beta_r/\alpha}) \supseteq \mathrm{constr}(C_{\gamma/\beta_r/\alpha}) = \mathrm{constr}(C_\alpha) \cup \{\rho_j\}$

}

Fig. B.2. Definition of partial individual plan.

FSP(P, GR, α, T_p, T_α, R_α, C_α)

(0) MB(GR, $(\forall G_j \in$ GR)Int.Th(G_j, Do(GR, α, T_α, constr(C_α)), T_p, T_α, C_α), T_p)$\wedge$

(1) $R_\alpha = \{\beta_i, \rho_j\} \wedge$ MB(GR, $R_\alpha \in$ *Recipes*(α), T_p)$\wedge$

(2) [$(\forall \beta_i \in R_\alpha \exists T_{\beta_i})$[

(a) [single.agent(β_i) $\wedge$ $(\exists G_k \in$ GR)[

(1) (a) Int.To(G_k, β_i, T_p, T_{β_i}, $C_{\beta_i/\alpha}$)$\wedge$

(b) MB(GR, Int.To(G_k, β_i, T_p, T_{β_i}, $C_{\beta_i/\alpha}$), T_p)$\wedge$

(2) [[basic.level(β_i)$\wedge$

(a) MB(GR, CBA(G_k, β_i, R_{Empty}, T_{b_i}, constr(C_α) $\cup \{\rho_j\}$), T_p)]$\otimes$

(3) [$\neg$basic.level(β_i)$\wedge$

(a) $(\exists P_{\beta_i}, R_{\beta_i})$FIP($P_{\beta_i}$, G_k, β_i, T_p, T_{β_i}, R_{β_i}, $C_{\beta_i/\alpha}$)$\wedge$

(b) MB(GR, $(\exists P_{\beta_i}, R_{\beta_i})$

(1) [CBA(G_k, β_i, R_{β_i}, T_{β_i}, constr(C_α) $\cup \{\rho_j\}$)$\wedge$

(2) FIP(P_{β_i}, G_k, β_i, T_p, T_{β_i}, R_{β_i}, $C_{\beta_i/\alpha}$)], T_p)]]$\wedge$

(4) MB(GR, $(\forall G_j \in$ GR, $G_j \neq G_k)$Int.Th(G_j, $(\exists R_{\beta_i})$

(a) CBA(G_k, β_i, R_{β_i}, T_{β_i}, constr(C_α) $\cup \{\rho_j\}$), T_p, T_{β_i}, $C_{cba/\beta_i/\alpha}$), T_p)] $\bigotimes$

(b) [multi.agent(β_i) $\wedge$ $(\exists \text{GR}_k \subseteq$ GR)[

(1) $(\exists P_{\beta_i}, R_{\beta_i})$FSP($P_{\beta_i}$, GR$_k$, β_i, T_p, T_{β_i}, R_{β_i}, $C_{\beta_i/\alpha}$)]$\wedge$

(2) MB(GR, $(\exists P_{\beta_i}, R_{\beta_i})$

(a) [CBAG(GR$_k$, β_i, R_{β_i}, T_{β_i}, constr(C_α) $\cup \{\rho_j\}$)$\wedge$

(b) FSP(P_{β_i}, GR$_k$, β_i, T_p, T_{β_i}, R_{β_i}, $C_{\beta_i/\alpha}$)], T_p)$\wedge$

(3) MB(GR, $(\forall G_j \in \text{GR} \setminus \text{GR}_k)$Int.Th($G_j$, $(\exists R_{\beta_i})$

(a) CBAG(GR$_k$, β_i, R_{β_i}, T_{β_i}, constr(C_α) $\cup \{\rho_j\}$), T_p, T_{β_i}, $C_{cbag/\beta_i/\alpha}$), T_p)]] $\bigotimes$

(3) [FSPC(GR, β_i, T_{β_i}, T_p, C_α, $\{\rho_j\}$)]]]

{COMMENT: constr($C_{\beta_i/\alpha}$) $\supseteq$ constr(C_α) $\cup \{\rho_j\}$}

Fig. B.3. Definition of full SharedPlan.

(3) $\mathrm{FSPC}(\mathrm{GR}, \beta_i, T_{\beta_i}, T_p, C_\alpha, \{\rho_j\})$

$(\exists \gamma, T_\gamma)[$

(a) $[\mathrm{single.agent}(\beta_i) \wedge (\exists G_c)[$

(1) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Int.Th}(G_j, (\exists R_{\beta_i})$

(a) $\mathrm{CBA}(G_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \Theta), T_p, T_\beta, C_{cba/\beta_i/\alpha}), T_p) \wedge$

(2) (a) $[\mathrm{single.agent}(\gamma) \wedge (\exists G_k \in \mathrm{GR})[$

(1) $\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(G_k, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \mathrm{constr}(c_\alpha) \cup \Theta), T_p) \wedge$

(2) $\mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}) \wedge$

(3) $\mathrm{MB}(\mathrm{GR}, \mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}), T_p) \wedge$

(4) $\mathrm{MP}(\mathrm{GR}, G_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})]] \otimes$

(b) $[\mathrm{multi.agent}(\gamma) \wedge (\exists \mathrm{GR}_k \subseteq \mathrm{GR})[$

(1) $\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(\mathrm{GR}_k, \gamma, T_\gamma, G_c, \beta_i, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \Theta), T_p) \wedge$

(2) $\mathrm{SGP}(\mathrm{GR}, \mathrm{GR}_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})]]]] \bigotimes$

(b) $[\mathrm{multi.agent}(\beta_i) \wedge (\exists \mathrm{GR}_c)$

(1) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Int.Th}(G_j, (\exists R_{\beta_i})$

(a) $\mathrm{CBAG}(\mathrm{GR}_c, \beta_i, R_{\beta_i}, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \Theta), T_p, T_\beta, C_{cbag/\beta_i/\alpha}), T_p) \wedge$

(2) (a) $[\mathrm{single.agent}(\gamma) \wedge (\exists G_k \in \mathrm{GR})[$

(1) $\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(G_k, \gamma, T_\gamma, \mathrm{GR}_c, \beta_i, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \Theta), T_p) \wedge$

(2) $\mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}) \wedge$

(3) $\mathrm{MB}(\mathrm{GR}, \mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_i/\alpha}), T_p) \wedge$

(4) $\mathrm{MP}(\mathrm{GR}, G_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})]] \otimes$

(b) $[\mathrm{multi.agent}(\gamma) \wedge (\exists \mathrm{GR}_k \subseteq \mathrm{GR})[$

(1) $\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(\mathrm{GR}_k, \gamma, T_\gamma, \mathrm{GR}_c, \beta_i, T_{\beta_i}, \mathrm{constr}(C_\alpha) \cup \Theta), T_p) \wedge$

(2) $\mathrm{SGP}(\mathrm{GR}, \mathrm{GR}_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta_i/\alpha})]]]]$

Fig. B.4. FSPC: contracting in FSP.

$\mathrm{MP}(\mathrm{GR}, G_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta/\alpha})$

(1) $[\neg\text{basic.level}(\gamma) \Rightarrow [$

(a) $(\exists P_\gamma, R_\gamma)\mathrm{FIP}(P_\gamma, G_k, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta/\alpha})\wedge$

(b) $\mathrm{MB}(\mathrm{GR}, (\exists P_\gamma, R_\gamma)$

(1) $[\mathrm{CBA}(G_k, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_{\gamma/\beta/\alpha}))\wedge$

(2) $\mathrm{FIP}(P_\gamma, G_k, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta/\alpha})], T_p)]]\wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR}, G_j \neq G_k)\mathrm{Int.Th}(G_j, (\exists R_\gamma)$

(a) $\mathrm{CBA}(G_k, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_{\gamma/\beta/\alpha})), T_p, T_\gamma, C_{cba/\gamma/\beta/\alpha}), T_p)$

$\mathrm{SGP}(\mathrm{GR}, \mathrm{GR}_k, \gamma, T_\gamma, T_p, C_{\gamma/\beta/\alpha})$

(1) $(\exists P_\gamma, R_\gamma)\mathrm{FSP}(P_\gamma, \mathrm{GR}_k, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta_i/\alpha})\wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\exists P_\gamma, R_\gamma)$

(a) $[\mathrm{CBAG}(\mathrm{GR}_k, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_{\gamma/\beta/\alpha}))\wedge$

(b) $\mathrm{FSP}(P_\gamma, \mathrm{GR}_k, \gamma, T_p, T_\gamma, R_\gamma, C_{\gamma/\beta/\alpha})], T_p)\wedge$

(3) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR} \setminus \mathrm{GR}_k)\mathrm{Int.Th}(G_j, (\exists R_\gamma)$

(a) $\mathrm{CBAG}(\mathrm{GR}_k, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_{\gamma/\beta/\alpha})), T_p, T_\gamma, C_{cbag/\gamma/\beta_i/\alpha}), T_p)]]$

Fig. B.5. The performance of the "contracting" action in the contracting cases of FSP: MP-Member of the group performs the contracting action. SGP-subgroup performs the contracting action.

$\mathrm{PSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha)$

$(\exists \{\beta_i, \rho_j\})[$

(0) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Int.Th}(G_j, \mathrm{Do}(\mathrm{GR}, \alpha, T_\alpha, \mathrm{constr}(C_\alpha)), T_p, T_\alpha, C_\alpha), T_p) \wedge$

(1) $\mathrm{MB}(\mathrm{GR}, (\exists R_\alpha)[\{\beta_i, \rho_j\} \subsetneq R_\alpha \wedge R_\alpha \in \mathit{Recipes}(\alpha)], T_p) \wedge$

(a) $[\mathrm{MB}(\mathrm{GR}, (\exists R_\alpha)[\{\beta_i, \rho_j\} \subset R_\alpha \wedge R_\alpha \in \mathit{Recipes}(\alpha)], T_p) \Rightarrow [$

(b) $(\exists T_{select.rec.g}, P_{select.rec.g}, R_{select.rec.g})$

(1) $\mathrm{FSP}(P_{select.rec.g}, \mathrm{GR}, \mathrm{Select\_Rec\_GR}(\mathrm{GR}, \alpha, \{\beta_i, \rho_j\}, T_{select.rec.g}), T_p, T_{select.rec.g},$
$R_{select.rec.g}, C_{select.rec.g/\alpha}) \wedge$

$\mathit{set}(\beta_i) = \mathit{set}[\beta_k] \cup \mathit{set}[\beta_r] \wedge (\forall \beta_r \in \mathit{set}[\beta_r] \exists T_{\beta_r})[$

(2) (a) $[\mathrm{single.agent}(\beta_r) \wedge (\exists G_k \in \mathrm{GR})$

(1) (a) $[\mathrm{Int.To}(G_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha}) \vee$

(b) $\mathrm{MB}(\mathrm{GR}, \mathrm{Int.To}(G_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha}), T_p) \wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_r})\mathrm{CBA}(G_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(3) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR}, G_j \neq G_k)\mathrm{Int.Th}(G_j, (\exists R_{\beta_r})$

(a) $\mathrm{CBA}(G_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_r}, C_{cba/\beta_r/\alpha}), T_p)]] \otimes$

(b) $[\mathrm{multi.agent}(\beta_r) \wedge (\exists \mathrm{GR}_k \subseteq GR)$

(1) $[(\exists P_{\beta_r})\mathrm{SP}(P_{\beta_r}, \mathrm{GR}_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha}) \wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\exists P_{\beta_r})\mathrm{SP}(P_{\beta_r}, \mathrm{GR}_k, \beta_r, T_p, T_{\beta_r}, C_{\beta_r/\alpha}), T_p) \wedge$

(3) $\mathrm{MB}(\mathrm{GR}, (\exists R_{\beta_r})\mathrm{CBAG}(\mathrm{GR}_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(4) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \{\mathrm{GR} \setminus \mathrm{GR}_k\})\mathrm{Int.Th}(G_j, (\exists R_{\beta_r})$

(a) $\mathrm{CBAG}(\mathrm{GR}_k, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_r}, C_{cbag/\beta_r/\alpha}), T_p)]] \otimes$

Fig. B.6. Definition of partial SharedPlan (Part A).

(3) $[\mathrm{PSPC}(\mathrm{GR}, \beta_r, T_{\beta_r}, C_\alpha \cup \{\rho_j\})]] \bigwedge$

(4) $(\forall \beta_k \in set[\beta_k])[$

(a) (1) $[\mathrm{single.agent}(\beta_k) \wedge$

(a) $\mathrm{MB}(\mathrm{GR}, (\exists G_k \in \mathrm{GR}, R_{\beta_k}, T_{\beta_k})\mathrm{CBA}(G_k, \beta_k, R_{\beta_k}, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(b) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Pot.Int.Th}(G_j, (\exists G_k \in \mathrm{GR}, T_{\beta_k})$

(1) $\mathrm{Do}(G_k, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_k}, C_{do/\beta_k/\alpha}), T_p)] \otimes$

(2) $[\mathrm{multi.agent}(\beta_k) \wedge$

(a) $\mathrm{MB}(\mathrm{GR}, (\exists \mathrm{GR}_k \subseteq \mathrm{GR}, R_{\beta_k}, T_{\beta_k})\mathrm{CBAG}(\mathrm{GR}_k, \beta_k, R_{\beta_k}, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(b) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Pot.Int.Th}(G_j, (\exists \mathrm{GR}_k \subseteq \mathrm{GR}, T_{\beta_k})$

(1) $\mathrm{Do}(\mathrm{GR}_k, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_k}, C_{do/\beta_k/\alpha}), T_p)]] \bigvee$

(b) (1) $[\mathrm{MB}(\mathrm{GR}, (\exists T_{\beta_k})\mathrm{CCG}(\mathrm{GR}, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p) \wedge$

(2) $\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR})\mathrm{Pot.Int.Th}(G_j, (\exists \gamma, T_\gamma, G_c, G_k, T_{\beta_k})$

(a) $[[(\mathrm{single.agent}(\gamma) \wedge G_k \in \mathrm{GR}) \otimes (\mathrm{multi.agent}(\gamma) \wedge G_k \subseteq \mathrm{GR})] \wedge$

(b) $\mathrm{Do}(G_k, \gamma, T_\gamma, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}) \wedge$

$\mathrm{GTD}(G_k, \gamma, T_\gamma, G_c, \beta_k, T_{\beta_k}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\})], T_p, T_{\beta_k}, C_{cont/\beta_k/\alpha}), T_p)]]$

{COMMENT:

$\mathrm{constr}(C_{select.rec.g/\alpha}) \supseteq$

$(\exists R_\alpha = \{\delta_v, \kappa_e\})$

$[\mathrm{MB}(\mathrm{GR}, R_\alpha \in Recipes(\alpha) \wedge \{\beta_i, \rho_j\} \subseteq R_\alpha, T_{select.rec.g}) \wedge$

$(\forall \delta_v \in R_\alpha)$

$\mathrm{MB}(\mathrm{GR}, (\exists T_{\delta_v})$

$[[\mathrm{single.agent}(\delta_v) \wedge (\exists G_{\delta_v} \in \mathrm{GR}, R_{\delta_v})\mathrm{CBA}(G_{\delta_v}, \delta_v, T_{\delta_v}, R_{\delta_v}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\})] \otimes$

$[\mathrm{multi.agent}(\delta_v) \wedge (\exists \mathrm{GR}_{\delta_v} \subseteq \mathrm{GR}, R_{\delta_v})\mathrm{CBAG}(\mathrm{GR}_{\delta_v}, \delta_v, T_{\delta_v}, R_{\delta_v}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\})] \vee$

$[\mathrm{CCG}(\mathrm{GR}, \delta_v, T_{\delta_v}, \mathrm{constr}(C_\alpha) \cup \{\kappa_e\})]], T_{select.rec.g})])]]]$

$\mathrm{constr}(C_{\beta_r/\alpha}) = \mathrm{constr}(C_\alpha) \cup \{\rho_j\}$

}

Fig. B.7. Definition of partial SharedPlan (Part B).

$\mathrm{PSP}(P, \mathrm{GR}, \alpha, T_p, T_\alpha, C_\alpha)$

(3) *Contracting case*: $\mathrm{PSPC}(\mathrm{GR}, \beta_r, T_{\beta_r}, C_\alpha, \{\rho_j\})$

(a) *Single-agent subaction*:

The group will get another agent G_c to do the subaction β_r:

(1) The group GR mutually believe that all members of the group are committed to G_c's success:

$$\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR}) \; \mathrm{Int.Th}(G_j, (\exists R_{\beta_r})\mathrm{CBA}(G_c, \beta_r, R_{\beta_r}, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p, T_{\beta_r}, C_{cba/\beta_r/\alpha}), T_p)$$

(a) The "contracting" act γ is single-agent;
there is a member of the group G_k such that,

(1) The group mutually believe that G_k can get G_c to do the subact by doing γ:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(G_k, \gamma, T_\gamma, G_c, \beta_r, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) G_k intends to do the "contracting" action:

$$\mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha})$$

(3) The group mutually believe that G_k intends to do the "contracting" action:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha}), T_p)$$

(4) The group mutually believe that G_k is able to do the "contracting" action.

$$\mathrm{MB}(\mathrm{GR}, (\exists R_\gamma)\mathrm{CBA}(G_k, \gamma, T_p, T_\gamma, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(5) The group mutually believe that all its members are committed to G_k's success:

$$\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \mathrm{GR}, G_j \neq G_k) \; \mathrm{Int.Th}(G_j, (\exists R_\gamma)\mathrm{CBA}(G_k, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_{\gamma/\beta_r/\alpha})), T_p, T_\gamma, C_{cba/\gamma/\beta_r/\alpha}), T_p)$$

(b) The "contracting" act is multi-agent action;
there is subgroup GR_k of the group such that:

(1) The group mutually believe that the subgroup GR_k can get G_c to do the subaction β_r by doing γ:

$$\mathrm{MB}(\mathrm{GR}, \mathrm{GTD}(\mathrm{GR}_k, \gamma, T_\gamma, G_c, \beta_r, T_{\beta_r}, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) The subgroup has SharedPlan for doing γ:

$$(\exists P_\gamma) SP(P_\gamma, \mathrm{GR}_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha})$$

(3) The group mutually believe that the subgroup has SharedPlan for doing γ.

$$\mathrm{MB}(\mathrm{GR}, (\exists P_\gamma) SP(P_\gamma, \mathrm{GR}_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha}))$$

(4) The group mutually believe that GR_k can bring about the "contracting" action.

$$\mathrm{MB}(\mathrm{GR}, (\exists R_\gamma)\mathrm{CBAG}(\mathrm{GR}_k, \gamma, T_p, T_\gamma, \mathrm{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(5) GR mutually believe that all its members are committed to GR_k's success:

$$\mathrm{MB}(\mathrm{GR}, (\forall G_j \in \{\mathrm{GR} \setminus \mathrm{GR}_k\}) \; \mathrm{Int.Th}(G_j, (\exists R_\gamma)\mathrm{CBAG}(\mathrm{GR}_k, \gamma, R_\gamma, T_\gamma, \mathrm{constr}(C_{\gamma/\beta_r/\alpha})), T_p, T_\gamma, C_{cba/\gamma/\beta_r/\alpha}), T_p)$$

Fig. B.8. PSP: Contracting-out case: single-agent.

$\text{PSP}(P, \text{GR}, \alpha, T_p, T_\alpha, C_\alpha)$

(3) *Contracting case*: $\text{PSPC}(\text{GR}, \beta_r, T_{\beta_r}, C_\alpha, \{\rho_j\})$

(b) *Multi-agent action*:

The group will get another group of agents GR_c to do do the subaction β_r:

(1) The group GR mutually believe that all members of the group are committed to GR_c's success:

$$\text{MB}(\text{GR}, (\forall G_j \in \text{GR}) \\ \text{Int.Th}(G_j, (\exists R_{\beta_r})\text{CBAG}(\text{GR}_c, \beta_r, R_{\beta_r}, T_{\beta_r}, \text{constr}(C_\alpha) \cup \{\rho_j\}), \\ T_p, T_{\beta_r}, C_{cbag/\beta_r/\alpha}), T_p)$$

(2) (a) The "contracting" act γ is single-agent; there is a member of the group G_k such that,

(1) The group mutually believe that G_k can get GR_c to do the subact β_r by doing γ:

$$\text{MB}(\text{GR}, \text{GTD}(G_k, \gamma, T_\gamma, \text{GR}_c, \beta_r, T_{\beta_r}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) G_k intends to do the "contracting" action:

$$\text{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha}), T_p)$$

(3) The group mutually believe that G_k intends to do the "contracting" action:

$$\text{MB}(\text{GR}, \text{Int.To}(G_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha}), T_p)$$

(4) The group mutually believe that G_k can bring about the "contracting" action.

$$\text{MB}(\text{GR}, (\exists R_\gamma)\text{CBA}(G_k, \gamma, T_p, T_\gamma, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(5) The group mutually believe that all its members are committed to G_k's success:

$$\text{MB}(\text{GR}, (\forall G_j \in \text{GR}, G_j \neq G_k) \\ \text{Int.Th}(G_j, (\exists R_\gamma)\text{CBA}(G_k, \gamma, R_\gamma, T_\gamma, \text{constr}(C_{\gamma/\beta_r/\alpha})), \\ T_p, T_\gamma, C_{cba/\gamma/\beta_r/\alpha}), T_p)$$

(b) The "contracting" act is multi-agent action; there is subgroup GR_k of the group such that:

(1) The group mutually believe that the subgroup GR_k can get G_c to do the subact by doing γ:

$$\text{MB}(\text{GR}, \text{GTD}(\text{GR}_k, \gamma, T_\gamma, G_c, \beta_r, T_{\beta_r}, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(2) The subgroup has SharedPlan for doing γ:

$$(\exists P_\gamma) SP(P_\gamma, \text{GR}_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha})$$

(3) The group mutually believe that the subgroup has SharedPlan for doing γ:

$$\text{MB}(\text{GR}, (\exists P_\gamma) SP(P_\gamma, \text{GR}_k, \gamma, T_p, T_\gamma, C_{\gamma/\beta_r/\alpha}))$$

(4) The group mutually believe that GR_k can bring about the "contracting" action:

$$\text{MB}(\text{GR}, (\exists R_\gamma)\text{CBAG}(\text{GR}_k, \gamma, T_p, T_\gamma, \text{constr}(C_\alpha) \cup \{\rho_j\}), T_p)$$

(5) GR mutually believe that all its members are committed to GR_k's success:

$$\text{MB}(\text{GR}, (\forall G_j \in \{\text{GR} \setminus \text{GR}_k\}) \\ \text{Int.Th}(G_j, (\exists R_\gamma)\text{CBAG}(\text{GR}_k, \gamma, R_\gamma, T_\gamma, \text{constr}(C_{\gamma/\beta_r/\alpha})), \\ T_p, T_\gamma, C_{cba/\gamma/\beta_r/\alpha}), T_p)$$

Fig. B.9. PSP: contracting-out case: multi-agent.

Axiom (A8):

$(\forall \eta, G_1, T_\eta, T_i)[$

G_1 is committed to G_2's success in doing β as part of G_1's intention to α

$(\exists \alpha, T_\alpha, R_\alpha, \beta, G_2)[$

$\text{Int.To}(G_1, \alpha, T_i, T_\alpha, C_\alpha) \wedge$

$\text{Int.Th}(G_1, (\exists R_\beta)\text{CBA}(G_2, \beta, R_\beta, T_\beta, \text{constr}(C_{\beta/\alpha})), T_i, T_\beta, C_{cba/\beta/\alpha}) \wedge$

It is "cheaper" to G_1 to help G_2 in doing β by doing η

$[\text{cost}(G, \text{Do}(G_1, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_\alpha, C_\alpha, R_\alpha,$

$\neg\text{Do}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})) \wedge \text{Do}(G_2, \beta, T_\beta, \text{constr}(C_{\beta/\alpha}))) -$

$\text{cost}(G_1, \text{Do}(G_1, \alpha, T_\alpha, \text{constr}(C_\alpha)), T_\alpha, C_\alpha, R_\alpha,$

$\text{Do}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})) \wedge \text{Do}(G_2, \beta, T_\beta, \text{constr}(C_{\beta/\alpha}))) >$

$\text{econ}'(\text{cost}(G_1, \text{Do}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})), T_\eta, C_{\eta/cba/\beta/\alpha}, R_\eta))] \wedge$

G_1 believes it can perform η

$\text{Bel}(G_1, (\exists R_\eta)\text{CBA}(G_1, \eta, T_\eta, \text{constr}(C_{\eta/cba/\beta/\alpha})), T_i)] \Rightarrow$

G_1 will consider doing η

$\text{Pot.Int.To}(G_1, \eta, T_i, T_\gamma, C_{\eta/cba/\beta/\alpha})]$

Fig. B.10. Another helpful-behavior axiom for intending-that.

References

[1] J. Allen and C.R. Perrault, Analyzing intention in utterances, *Artif. Intell.* **15** (1980) 143–178; also in: B.J. Grosz, K. Sparck Jones and B.L. Webber, eds., *Readings in Natural Language Processing* (Morgan Kaufmann, Los Altos, CA, 1986) 441–458.

[2] D. Appelt and A. Kronfeld, A computational model of referring, in: *Proceedings IJCAI-87*, Milan (1987) 640–647.

[3] K.J. Arrow, The economics of agency, in: J. Pratt and R. Zeckhauser, eds., *Principals and Agents: The Structure of Business* (Harvard Business School Press, Cambridge, MA, 1985) 37–51.

[4] C.T. Balkanski, Modelling act-type relations in collaborative activity, Tech. Rept. TR-23-90, Harvard University, Cambridge, MA (1990).

[5] C.T. Balkanski, Actions, beliefs and intentions in rationale clauses and means clauses, Ph.D. Thesis, Harvard University, Cambridge, MA (1993).

[6] A.H. Bond and L. Gasser, An analysis of problems and research in DAI, in: A.H. Bond and L. Gasser, eds., *Readings in Distributed Artificial Intelligence* (Morgan Kaufmann, San Mateo, CA, 1988) 4–46.

[7] M.E. Bratman, *Intention, Plans, and Practical Reason* (Harvard University Press, Cambridge, MA, 1987).

[8] M.E. Bratman, What is intention?, in: P.R. Cohen, J. Morgan and M.E. Pollack, eds., *Intentions in Communication* (MIT Press, Cambridge, MA, 1990) Chapter 2, 15–31.

[9] M.E. Bratman, Shared cooperative activity, *Philos. Rev.* **101** (1992) 327–341.

[10] M.E. Bratman, D.J. Israel and M.E. Pollack, Plans and resource-bounded practical reasoning, *Comput. Intell.* **4** (1988) 349–355.

[11] P. Cohen and H. Levesque, Intention is choice with commitment, *Artif. Intell.* **42** (1990) 263–310.

[12] P. Cohen and H. Levesque, Teamwork, *Noûs* **25** (1991) 487–512.

[13] S.E. Conry, K. Kuwabara, V.R. Lesser and R.A. Meyer, Multistage negotiation for distributed satisfaction, *IEEE Trans. Syst. Man Cybern.* **21** (1991) 1462–1477.

[14] R. Davis and R.G. Smith, Negotiation as a metaphor for distributed problem solving, *Artif. Intell.* **20** (1983) 63–109.

[15] K.S. Decker, E.H. Durfee and V.R. Lesser, Evaluating research in cooperative distributed problem solving, in: M.N. Huhns, ed., *Distributed Artificial Intelligence*, Research Notes on Artificial Intelligence **2** (Pitman, London, 1989) 487–519.

[16] E.H. Durfee, *Coordination of Distributed Problem Solvers* (Kluwer Academic Publishers, Boston, MA, 1988).

[17] E. Ephrati and J.S. Rosenschein, Reaching agreement through partial revelation of preferences, in: *Proceedings ECAI-92*, Vienna (1992) 229–233.

[18] E. Ephrati and J.S. Rosenschein, Planning to please: following another agent's intended plan, *J. Group Decis. Negot.* **2** (1993) 219–235.

[19] L. Gasser, Social concepts of knowledge and action: DAI foundations and open systems semantics, *Artif. Intell.* **47** (1991) 107–138.

[20] A.I. Goldman, *A Theory of Human Action* (Princeton University Press, Princeton, NJ, 1970).

[21] B.J. Grosz and S. Kraus, Collaborative plans for group activities, in: R. Bajcsy, ed., *Proceedings IJCAI-93*, Chambery (1993) 367–373.

[22] B.J. Grosz and C. Sidner, Attention, intentions, and the structure of discourse, *Comput. Linguist.* **12** (1986) 175–204.

[23] B.J. Grosz and C. Sidner, Plans for discourse, in: P. Cohen, J.L. Morgan and M.E. Pollack, eds., *Intentions in Communication* (MIT Press, Cambridge, MA, 1990).

[24] B.J. Grosz and C. Sidner, A reply to Hobbs, in: P.R. Cohen, J.L. Morgan and M.E. Pollack, eds., *Intentions in Communication* (MIT Press, Cambridge, MA, 1990) 461–462.

[25] J. Hobbs, Ontological promiscuity, in: *Proceedings 23rd Annual Meeting of the Association for Computational Linguistics*, Stanford, CA (1985).

[26] M.N. Huhns, M.P. Singh, T. Ksiezyk and N. Jacobs, Global information management via local autonomous agents, in: *Proceedings ICOT International Symposium on Fifth Generation Computer Systems: Workshop on Heterogeneous Cooperative Knowledge Bases*, Tokyo (1994) 1–15.

[27] D Israel, J. Perry and S. Tutiya, Actions and movements, in: J. Mylopoulos and R. Reiter, eds., *Proceedings IJCAI-91*, Chambery (1991) 1060–1065.

[28] N. Jennings, On being responsible, in: *Decentralized Artificial Intelligence* **3** (Elsevier Science Publishers, Amsterdam, 1992) 93–102.

[29] N.R. Jennings, Controlling cooperative problem solving in industrial multi-agent systems using joint intentions, *Artif. Intell.* **75** (1995) 1–46.

[30] H. Kautz, A circumscriptive theory of plan recognition, in: P.N. Cohen, J.L. Morgan and M.E. Pollack, eds., *Intentions in Communication* (MIT Press, Cambridge, MA, 1989).

[31] M. Klein, Supporting conflict resolution in cooperative design systems, *IEEE Trans. Syst. Man Cybernet.* **21** (1991) 1379–1390.

[32] K. Konolige and M.E. Pollack, A representationalist theory of intention, in: R. Bajcsy, ed., *Proceedings IJCAI-93*, Chambery (1993) 390–395.

[33] S. Kraus, Agents contracting tasks in non-collaborative environments, in: *Proceedings AAAI-93*, Washington, DC (1993) 243–248.

[34] S. Kraus and D. Lehmann, Knowledge, belief and time, *Theoret. Comput. Sci.* **58** (1988) 155–174.

[35] S. Kraus and D. Lehmann, Designing and building a negotiating automated agent, *Comput. Intell.* **11** (1995) 132–171.

[36] S. Kraus, M. Nirkhe and K. Sycara, Reaching agreements through argumentation: a logical model, in: *Proceedings Workshop on Distributed Artificial Intelligence* (1993) 233–247; also presented in AAAI-93 Workshop on AI Theories of Groups and Organizations: Conceptual and Empirical Research.

[37] S. Kraus and J. Wilkenfeld, Negotiations over time in a multi-agent environment: preliminary report, in: *Proceedings IJCAI-91*, Sydney, Australia (1991) 56–61.

[38] S. Kraus and J. Wilkenfeld, A strategic negotiations model with applications to an international crisis, *IEEE Trans. Syst. Man Cybernet.* **23** (1993) 313–323.

[39] V.R. Lesser, A retrospective view of FA/C distributed problem solving, *IEEE Trans. Syst. Man Cybernet.* **21** (1991) 1347–1362.
[40] H.J. Levesque, P. Cohen and J. Nunes, On acting together, in: *Proceedings AAAI-90*, Boston, MA (1990) 94–99.
[41] D.J. Litman and J.F. Allen, Discourse processing and commonsense plans, in: P.R. Cohen, J.L. Morgan and M.E. Pollack, eds., *Intentions in Communication* (MIT Press, Cambridge MA, 1990).
[42] K. Lochbaum, An algorithm for plan recognition in collaborative discourse, in: *Proceedings 29th Annual Conference of the Association for Computational Linguistics*, Berkeley, CA (1991) 33–38.
[43] K. Lochbaum, Using collaborative plans to model the intentional structure of discourse, Ph.D. Thesis, Harvard University, Cambridge, MA (1994); available as Tech. Rept. TR-25-94.
[44] K.E. Lochbaum, The use of knowledge preconditions in language processing, in: C.S. Mellish, ed., *Proceedings IJCAI-95*, Montreal, Que. (1995) 1260–1266.
[45] K. Lochbaum, B.J. Grosz and C. Sidner, Models of plans to support communication: an initial report, in: *Proceedings AAAI-90*, Boston, MA (1990) 485–490.
[46] T.W. Malone, R.E. Fikes and M.T. Howard, Enterprise: a market-like task scheduler for distributed computing environments, in: B.A. Huberman, ed., *The Ecology of Computation* (North-Holland, Amsterdam, 1988) 177–205.
[47] T. Moehlman, V. Lesser and B. Buteau, Decentralized negotiation: an approach to the distributed planning problem, *Group Decis. Negot.* **2** (1992) 161–191.
[48] L. Morgenstern, Foundations of a logic of knowledge, action, and communication, Ph.D. Thesis, New York University, New York (1988).
[49] E. Osawa and M. Tokoro, Collaborative plan construction for multiagent mutual planning, in: *Decentralized Artificial Intelligence* **3** (Elsevier Science Publishers, Amsterdam, 1992) 169–185.
[50] F. Polat, S. Shekhar and A. Guvenir, Distributed conflict resolution among cooperating expert systems, *Expert Syst.* **10** (1993) 227–236.
[51] M.E. Pollack, Inferring domain plans in question-answering, Ph.D. Thesis, University of Pennsylvania, Philadelphia, PA (1986).
[52] M.E. Pollack, Inferring domain plans in question-answering, Tech. Rept. MS-CIS-86-40, Department of Computer Science and Information Science, Moore School, University of Pennsylvania, Philadephia, PA (1986).
[53] M.E. Pollack, A model of plan inference that distinguishes between the beliefs of actors and observers, in: *Proceedings 24th Annual Meeting of the Association for Computational Linguistics*, New York (1986) 207–214.
[54] M.E. Pollack, Plans as complex mental attitudes, in: P.N. Cohen, J.L. Morgan and M.E. Pollack, eds., *Intentions in Communication* (MIT Press, Cambridge, MA, 1990).
[55] A. Rao and M. Georgeff, Modelling rational agents within bdi architecture, in: *Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning*, Cambridge, MA (1991) 473–484.
[56] S.J. Russell and E.H. Wefald, Principles of meta-reasoning, in: *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Ont. (1989) 400–411.
[57] S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Prentice Hall, Englewood Cliffs, NJ, 1994).
[58] J.R. Searle, Collective intentions and actions, in: *Intentions in Communication* (MIT Press, Cambridge, MA, 1990) Chapter 19.
[59] O. Shehory and S. Kraus, Task allocation via coalition formation among autonomous agents, in: C.S. Mellish, ed., *Proceedings IJCAI-95*, Montreal, Que. (1995) 655–661.
[60] Y. Shoham, Agent-oriented programming, *Artif. Intell.* **60** (1993) 51–92.
[61] E. Sonenberg, G. Tidhar, E. Werner, D. Kinny, M. Ljungberg and A. Rao, Planned team activity, Tech. Rept. 26, Australian Artificial Intelligence Institute (1992).
[62] K.P. Sycara, Resolving goal conflicts via negotiation, in: *Proceedings AAAI-88*, St. Paul, MN (1988) 245–250.
[63] B. Vermazen, Objects of intention, *Philos. Stud.* **71** (1993) 223–265.
[64] E. Werner, Cooperating agents: a unified theory of communication and social structure, in: L. Gasser and M.N. Huhns, eds., *Distributed Artificial Intelligence*, Research Notes on Artificial Intelligence **2** (Pitman London, 1987) 3–36.

[65] E. Werner, Toward a theory of communication and cooperation for multiagent planning, in: *Proceedings Second Conference on Theoretical Aspects of Reasoning about Knowledge*, Pacific Grove, CA (1988) 129–143.

[66] M. Wooldridge and N.R. Jennings, Formalizing the cooperative problem solving process, in: *Proceedings 13th International Workshop on Distributed Intelligence*, Lake Quinalt, WA (1994) 403–417.

[67] G. Zlotkin and J.S. Rosenschein, Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains, *IEEE Trans. Syst. Man Cybernet.* **21** (1991) 1317–1324.