Information Needs in Agent Teamwork

John Yen JYEN@IST.PSU.EDU
Xiaocong Fan ZFAN@IST.PSU.EDU
School of Information Sciences and Technology,
The Pennsylvania State University, University Park, PA 16802, USA

Richard A. Volz VOLZ@CS.TAMU.EDU
Department of Computer Science,
Texas A&M University, College Station, TX 77843, USA

Abstract

Members of effective human teams can often anticipate information needs of teammates and offer relevant information to them proactively. Such capabilities are highly desirable for agent teams to achieve better teamwork processes for supporting information gathering, information fusion, and decision makings of teammates. However, there is a lack of agent theories for specifying such proactive agent behavior. The starting point of establishing such a theory is to formally characterize the concept of "information-need" and provide a framework for reasoning about others' information-needs. To this end, in this paper we (1) introduce a modal operator to represent agents' information-needs; (2) investigate levels of information-needs using the idea of precondition-tree; (3) identify several types of information-needs prevalent in agent teamwork; (4) provide and justify the axioms for anticipating others' information-needs; and (5) to complete the framework, introduce an axiom for enabling agents to commit to helping others with their information-needs. This paper thus provides a formal basis for developing agent theories about proactive information delivery behavior.

1. Introduction

Psychological studies about human teamwork have shown that members of an effective team can often anticipate needs of other teammates and choose to assist them proactively based on a shared mental model (Rouse, Cannon-Bowers, & Salas, 1992). Anticipating needs of teammates and proactively assisting them regarding their needs are also desirable teamwork behaviors for other types of agent teams. These teams in general share the following characteristics: (1) distributed expertise: the distribution of expertise among team members introduces the needs for information exchange; (2) time pressure: information has to be delivered timely in time-stress domains; (3) limited communication: this necessitates selective information exchange; and (4) information richness: the team has to deal with (filter, fuse, and interpret) overwhelming amount of information continually. For instance, applications for dynamic domains such as Battlespace Infospheres often require a large number of intelligent agents and human agents to form a team to cooperate effectively in information gathering, information fusion, sense-making, information delivering, and group decisions. Such teams require the involved agents to be able to anticipate information needs of teammates and offer relevant information proactively.

It's highly desirable to establish an agent theory about proactive information delivery for several reasons. First, such a theory can provide a guide for the specification and design of agent architectures, algorithms, and applications that support proactive information delivery capabilities. Second, it would be helpful for understanding the mental states of the performers involved in proactive communication actions, as well as uncovering the limitations and necessary assumptions of proactive information exchanges implemented in a multi-agent system that might be overlooked otherwise. Third, it could offer opportunities for exploiting novel agent communication protocols that support proactive teamwork behaviors. However, such a theory cannot be directly derived from any of the existing frameworks.

To establish an agent theory about proactive information delivery, at least three issues need to be addressed. First, the concept of "information needs" should be treated as first-class object. Its properties and possible relationships with agents' mental attitudes should be examined. In particular, context information should be embedded into the notion in order to support the flexibility of needs-shifting. Second, the framework should allow an agent to anticipate teammates' information needs based on either logical axioms, assumptions, heuristic rules, or approximate reasoning, etc. Such anticipation also raises a demand for modeling shared team activities (e.g., team process) and nested epistemic states (e.g., one's belief of other's beliefs). Third, the framework should connect information needs to proactive communications. Such connections should be intuitively simple while sufficiently flexible to enable agents to make the final decisions on whether and how to communicate. To better understand and establish their relations, appropriate intentional semantics should be given to those proactive communicative actions in terms of information-needs as well as other mental attitudes.

As the starting point, this paper focuses on the first issue. The rest of the paper is organized as follows. We prepare our framework in section 2. Section 3 discusses preconditions and the idea of precondition-trees. The concept of "information-need" is introduced in section 4, where the levels and types of information-needs are also covered. Section 5 mainly presents and justifies axioms for anticipating different types of information-needs, and section 6 examines the issue of committing to helping others' information-needs. Some related works are discussed in Section 7, and section 8 concludes the paper.

2. Preliminaries

We adopt the SharedPlans theory (Grosz & Kraus, 1996, 1999) as the cornerstone of our framework. The SharedPlan formalism of collaborative planning originated from Pollack's mental state model of plans (Pollack, 1990), and was further extended with the treatment of partiality and the evolution process of shared plans (Grosz & Kraus, 1996, 1999).

Some notations in (Grosz & Kraus, 1996) are adjusted here for convenience. We use $A, B, \cdots, A', \cdots$ to refer to individual agents; use $\alpha, \beta, \cdots$ to denote actions; use Ω to denote a set of actions; use t with superscripts or subscripts to denote time points (by default, t refers to the current time point)[1]; use R with an action as its subscript to denote a recipe for that action; use character C with a subscript, such as C_α,C_p,C_n,C_1, etc., to refer to a context, and use Θ with an action as its subscript to denote a constraints for that action.

1. Time is treated as an ordered set of discrete points. We assume primitive actions performed at time t will be done by the next time point. The performance of a complex action may span several time points. In such cases, certain desired duration may be specified as a constraint for doing the action.

The components of a constraints may be classified into three types: execution preconditions, recipe-constraints (e.g., time, location or other resources considered in the selection of recipes for the action), and constraints considered in reconciling conflict intentions. The composition of a context will be discussed later.

Shared plans are defined in terms of modal operators, meta-predicates, actions, etc. In addition to Bel (belief) and MB (mutual belief), three modal operators are used to relate agents and actions: Exec, Commit, and Do, and four modal operators are used to specify the attitudes of intention: Int.To, Int.Th, Pot.Int.To, and Pot.Int.Th. Modal operator $\mathsf{Exec}(A, \alpha, t, \Theta_\alpha)$ is used to represent the fact that agent A has the ability to perform basic-level action α at time t under the constraints Θ_α; $\mathsf{Commit}(A, \alpha, t_1, t_2, C_\alpha)$ represents the commitment of agent A at t_1 to perform the basic-level action α at t_2 under the context C_α; $\mathsf{Do}(A, \alpha, t, \Theta_\alpha)$ is used to denote that an agent (or a group of agents) A performs action α at time (beginning at, in the case of an interval) t under constraints Θ_α.

Intention operator $\mathsf{Int.To}(A, \alpha, t, t_\alpha, C_\alpha)$ means that at time t, agent A intends to do α at time t_α in the context C_α. Int.To stimulates means-end reasoning. When the action that an agent intends to do is a basic-level action (and the agent does have the ability of doing that action), the Int.To reduces to Commit; otherwise, the agent will try to compose a recipe for the action before doing it. $\mathsf{Int.Th}(A, p, t, t', C_p)$ means agent A at t intends that p hold at t' under the intentional context C_p. While intentions-to only apply to individual agent actions, intentions-that can be used to initiate team activities involving a group of cooperators. In fact, Int.Th plays essential roles in meshing subplans, helping teammates, and reconciling resource or intention conflicts. $\mathsf{Pot.Int.To}(A, \alpha, t, t_\alpha, C_\alpha)$ means agent A has a potential intention to do α. Int.To is used to represent intentions committed by agents, while Pot.Int.Tos refer to intentions not committed yet. An agent may convert a Pot.Int.To to an Int.To if the potential intention does not contradict the already adopted intentions. An Pot.Int.To has to be dropped should there be any conflicts. Similarly, $\mathsf{Pot.Int.Th}(A, p, t, t', C_p)$ refers to potential intention-that. A Pot.Int.Th needs go through similar deliberation process before it can be adopted as a committed intention.

Several meta-predicates were defined. Among others, $\mathsf{CBA}(A, \alpha, R_\alpha, t_\alpha, \Theta_\alpha)$ means agent A's ability to bring about single-agent action α at t_α under constraints Θ_α by following recipe R_α. It represents the knowledge an agent has about its ability to perform an action in a plan. Shared mental states among a team of agents are reflected in their partial shared plans (denoted by PSP) or full shared plans (denoted by FSP).

Grosz and Kraus proposed several axioms for deriving helpful behaviors (Grosz & Kraus, 1996, 1999). The following one simplifies the axiom in (Grosz & Kraus, 1999) without considering the case of multiple-agent actions (we assume communicative acts to be examined are single-agent actions) and the case of action-intention conflicts.

Axiom 1 $\forall A, p, t, \beta, t_\beta, t' > t_\beta, C_p\cdot$
$\mathsf{Int.Th}(A, p, t, t', C_p) \wedge \neg \mathsf{Bel}(A, p, t) \wedge \mathsf{Lead}(A, \beta, p, t, t_\beta, \Theta_\beta) \Rightarrow$
$\quad \mathsf{Pot.Int.To}(A, \beta, t, t_\beta, \Theta_\beta \wedge C_p)$, *where*
$\mathsf{Lead}(A, \beta, p, t, t_\beta, \Theta_\beta) \triangleq \mathsf{Bel}(A, \exists R_\beta \cdot \mathsf{CBA}(A, \beta, R_\beta, t_\beta, \Theta_\beta)), t) \wedge$
$\quad [\mathsf{Bel}(A, (\mathsf{Do}(A, \beta, t_\beta, \Theta_\beta) \Rightarrow p), t) \vee \mathsf{Bel}(A, \mathsf{Do}(A, \beta, t_\beta, \Theta_\beta) \Rightarrow$
$\quad\quad [\exists B, \alpha, R_\alpha, t_\alpha, t''\cdot (t_\alpha > t_\beta) \wedge (t_\alpha > t'') \wedge \mathsf{CBA}(B, \alpha, R_\alpha, t_\alpha, \Theta_\alpha) \wedge$
$\quad\quad\quad \mathsf{Pot.Int.To}(B, \alpha, t'', t_\alpha, \Theta_\alpha) \wedge (\mathsf{Do}(B, \alpha, t_\alpha, \Theta_\alpha) \Rightarrow p)], t)]$.

Axiom 1 says that if an agent does not believe p is true now, but has an intention that p be true at some future time, it will consider doing some action β if it believes the performance of β could contribute to making p true either directly or indirectly through the performance of another action by another agent.

2.1 Assumptions on Mental Attitudes

We adopt the typical treatment of the belief attitude and assume operator Bel conforms to the K-,D-,4- and 5- axioms of modal logic (Fagin, Halpern, Moses, & Vardi, 1995). In addition, we assume idempotence property holds for Bel, i.e., $\mathsf{Bel}(A, \mathsf{Bel}(A, p, t), t) \Leftrightarrow \mathsf{Bel}(A, p, t)$ (the $\Leftarrow$ part corresponds to axiom '4'). We adopt the K-axiom and D-axiom of modal logic for the intentional attitudes Int.To and Int.Th, and adopt the K-axiom for Pot.Int.To and Pot.Int.Th.

Intentions and beliefs persist by default until they conflict the new acquired information or the contexts for keeping the intentions no longer hold. For Int.Th, as well as the associated context, the second time argument also serves as a constraints for holding the intention-that. More specifically, suppose agent A has an intention $\mathsf{Int.Th}(A, p, t, t_3, C_p)$, and C_p keeps to be true before t_3. As time goes on from t to some time t_1 ($< t_3$), the intention will become $\mathsf{Int.Th}(A, p, t_1, t_3, C_p)$. Now suppose at t_1 agent A comes to believe p. In considering that p might get changed in between t_2 and t_3, A should continue to hold the intention until t_3. Of course, in some cases achievement goals can be reduced to maintenance goals. For instance, if p is maintainable for A (e.g., the change of p is completely under the control of A), A could replace $\mathsf{Int.Th}(A, p, t_1, t_3, C_p)$ with $\mathsf{Int.To}(A, \mathit{maintain}(p), t_1, t_3, C_p)$, by which the agent is committed to maintaining its belief about p until t_3.

For the relationships between Bel and intentions (Int.To and Int.Th), in addition to those in (Grosz & Kraus, 1996), we also assume "goals are known" (Cohen & Levesque, 1990a). That is,

$$\mathsf{Int.To}(A, \alpha, t, t', C_\alpha) \Rightarrow \mathsf{Bel}(A, \mathsf{Int.To}(A, \alpha, t, t', C_\alpha), t),$$
$$\mathsf{Int.Th}(A, p, t, t', C_p) \Rightarrow \mathsf{Bel}(A, \mathsf{Int.Th}(A, p, t, t', C_p), t).$$

2.2 Actions and Context

An action is either primitive, or complex. A complex action (expression) can be built from primitive actions by using the constructs of dynamic logic: $\alpha; \beta$ for sequential composition, $\alpha|\beta$ for nondeterministic choice, $p?$ for testing (where p is a logical formula), and α^* for repetition. Let $\mathit{post}(\alpha)$ return a conjunction of propositions that describe the effects of α.

A recipe for a complex action γ is a specification of a group of subsidiary actions at different levels of abstraction, the doing of which under certain constraints constitutes the performance of γ. Thus, a recipe is in *per se* composed of an action expression and a set of constraints on the action expression. A set of recipes may be specified for an action. Let $\mathit{recipe}_A(\alpha)$ be the set of recipes of α specified for agent A. $\mathit{recipe}_A(\alpha)$ and $\mathit{recipe}_B(\alpha)$ may be the same, or overlapped, or even disjoint.

As mentioned above, the SharedPlans theory assumes that all actions are intended, committed and performed in some specific context (Grosz & Kraus, 1996). A context might specify certain constraints on the performance of α (e.g., the deadline of doing α, dependencies of α on other actions, etc.), the reason of doing α (e.g., α is part of the recipe adopted to do some higher-level action, the doing of which is part of doing yet another higher-level action, and so on, until the top-level task or goal.), etc. Contexts can be used for other means. For instance, there is also a context parameter in the modal operator InfoNeed to be introduced later.

We use C or $\mathcal{C}$ with subscripts or superscripts to refer to contexts. Nevertheless, the subscript (or superscript) on a context does not impart any meanings to the context; the meaning of a context only depends on where the context occurs. For instance, when C_1 occurs as an argument of $Int.To$ (intention-to), it refers to the context in which the action (another argument of $Int.To$) is being done; when C_1 occurs as an argument of $Int.Th$ (intention-that), it refers to the context in which the proposition (another argument of $Int.Th$) is intended. However, to make notations more consistent, we take it as a convention by using C_α (or C_p) to refer to the context in which action α (or proposition p) is concerned.

A context is composed of a set of formulae, which are collectively evaluated as one conjunction. For this reason, if $C_1 = \{p_1, p_2\}$ and p_3 is some other formula, $C_1 \cup \{p_3\}$ and $p_1 \wedge p_2 \wedge p_3$ are equivalent when used as contexts. Conveniently, $C_1 \wedge p_3$ is also used to refer to $C_1 \cup \{p_3\}$.

The componential formulae of a context may play different roles. Some may serve as constraints; some may serve as traces of explanation; some may serve as criteria for attention management such as goal reconciliation or task delegation; and some may serve as specification of agent social relationships such as the agent's social roles, sincerity, etc. Thus, we assume associated with each formula of a context there is certain meta-level information indicating its roles, and there are functions defined for obtaining those components related to a specific role. In particular, we assume $Constr(C_1)$ returns the constraints component of context C_1.

3. Preconditions and Precondition Trees

Prior to performing a plan or an action, an agent typically needs to check whether the plan or action is both physically and epistemically feasible (Davis, 1994). In other words, obstacles to plans or actions come in one of two varieties: physical and informational. Accordingly, we distinguish physical preconditions and informational preconditions.

For instance, suppose in a battlefield domain there is a complex action called *RemoveThreat*: upon knowing a threat from an enemy unit, the performers of this action may either choose to attack the enemy from the flank, or wait at ease for the exhausted enemy. This action, $\mathit{RemoveThreat}(?e, ?loc, ?dir, ?num)$, can be represented as:

$(\mathit{MoveToFlank}(?e, ?loc, ?dir); \mathit{Fire}(?e, ?num))|$

$((\mathit{FarAway}(?e, \mathit{Self})?;\ \mathit{Wait}(\mathit{Self}))^*; \mathit{Fire}(?e, ?num))$.

Assume the preconditions of *RemoveThreat* involve three pieces: *CanFight* (*Self*): the agent can fight (e.g., have enough fighting power, can move, etc.); $\mathit{Threat}(?e, ?loc, ?dir, ?num)$: the agent knows the threat to be removed; and $\mathit{Outmatch}(?e, ?num)$: the agent knows its own team outmatches the enemy unit. Here, as far as the action *RemoveThreat* is concerned, *CanFight*(*Self*) is a kind of physical preconditions, while $\mathit{Threat}(?e, ?loc, ?dir, ?num)$ and $\mathit{Outmatch}(?e, ?num)$ are kinds of informational preconditions.

As far as helping behavior is concerned, the other agents can help the performers of *RemoveThreat* with the physical obstacles: if a performer cannot fight, they could enable the performer by delivering supplies or remove other potential barriers. The other agents can also help the performers of *RemoveThreat* with the informational obstacles: if a performer does not know the approaching threat, they would inform the threat information to the performer proactively. Since proactive information delivery is our concern, we will focus on informational preconditions only.

Informational preconditions may also have different varieties. A complex action or plan may have associated *constraints* which have to be satisfied prior to its being performed. For instance, prior to removing a threat an agent may have to know whether its own team outmatches the threatening enemies. In addition, a complex action or plan may have *knowledge preconditions* (Morgenstern, 1987; Davis, 1994): knowing enough to carry out a plan. Lochbaum (Lochbaum, 1995) recasted the observations on knowledge preconditions made by Morgenstern (Morgenstern, 1987) in the terminology of the SharedPlan framework, where predicates *has.recipe* and *id.params* are used respectively to represent (1) agents need to know recipes (*know-how* information) for the act to be performed, and (2) agents must be able to identify the parameters of the act to be performed.

In our pursuit of this research, without loss of generality, we assume agents already have relevant recipes for their actions and plans (to withdraw this assumption, we may allow agents to exchange meta-level information about recipes). To further simplify the issue of parameter identification, we also assume the parameters of an action either have constant values, or their values are passed by from a higher-level action (plan), or can be determined as soon as the preconditions are satisfied. Consequently, the task of parameter identification for an action (or plan) is reduced to satisfying the preconditions of the action (or plan). For example, the parameters of *RemoveThreat* are determined as soon as the predicate *Threat* can be unified successfully with a fact from the agent's belief base.

Now we formally characterize action preconditions. As we mentioned in section 2.2, several recipes may be specified for an action. Thus, different from the treatment in (Grosz & Kraus, 1996), we assume action preconditions depends on recipes. This is reasonable. For instance, another recipe for *RemoveThreat* can be specified as: recruit an echelon unit to induce the approaching enemy to move away from the crucial area. To carry out this recipe, the agents recruited to remove the threat also need to know the location of the crucial area, as well as the pre-requisite information about the approaching enemy. Let R_α be a recipe for action α, $pre(R_\alpha)$ be the preconditions specified for R_α, then we write $pre(\alpha)$ to be the set $\cup_{R_\alpha \in recipe(\alpha)} pre(R_\alpha)$. By $I \in pre(\alpha)$ we mean I is a conjunct of $pre(\alpha)$.

We view preconditions as having a tree structure capturing the hierarchical compositions reflected in agents' inference knowledge. For the purpose of concept illustration, we use Horn-clauses as the way to represent such inference knowledge. For instance, *Threat* may be the head predicate of the Horn-clause:

$$Threat(?e, ?loc, ?dir, ?num) \leftarrow IsEnemy(?e), At(?e, ?loc, NOW), Dir(?e, ?dir), Number(?e, ?num),$$

that is, an agent could deduce the existence of threat if it has beliefs about the identified enemy unit (*IsEnemy*), the location of the enemy unit(*At*), the moving direction of the enemy unit (*Dir*), and the number of enemies in the unit (*Number*). *Dir* may be the head of Horn-clause: $Dir(?e, ?dir) \leftarrow At(?e, ?l1, NOW-1), At(?e, ?l2, NOW), Compass(?l1, ?l2, ?dir)$, that is, to deduce the moving direction, an agent needs to know the change of location,

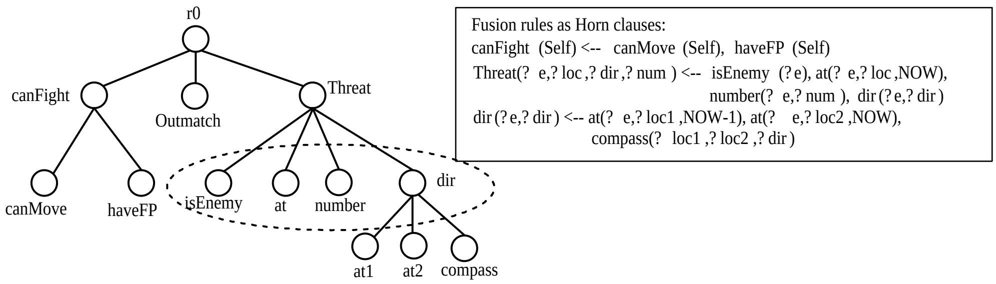

Figure 1: The Precondition Tree of *RemoveThreat*

from which to infer the direction. And *CanFight* may be the head of Horn-clause:
$CanFight(Self) \leftarrow HaveFP(Self), CanMove(Self)$,
that is, to be able to fight, an agent needs to have enough fighting power, and it can move to the targets. For this example, a precondition tree of *RemoveThreat* can be constructed as shown in Fig.1.

In general, precondition trees can be generated as follows. If the precondition of plan p_i is a single predicate, create a root node labeled with the predicate; if the precondition is composed of a set of predicates, create a virtual root node, then create a node for each of the predicates and make these nodes sons of the virtual root node. Then, populate the tree by applying the following recursive algorithm to the leaf nodes.

Algorithm 1: populateTree(Node nd, Predicate pd)
  hc = getHornClause(pd); /* get the Horn Clause with pd as its head*/ (1)
  if (hc is Null) return;
  plist = tail(hc); /* get all the negative literals of hc */
  for each pred in plist
    pn = createNode(pred);
    addSon(nd, pn);
    populateTree(pn, pred);
end.

Note that in Algorithm 1 it is assumed that each predicate can be the head of at most one Horn-clause (see statement (1)). In case that for a predicate there exist multiple ways of decomposition, colored-precondition-tree can be introduced such that analysis can be carried out by following the trunk with certain color whenever necessary. Also note that in a precondition tree, the nodes at the same level collectively form a context for each individual. For instance, in Fig.1, as far as threat identification is concerned, $Dir(?e, ?dir)$ is useful only when it is evaluated together with $IsEnemy(?e)$, $At(?e, ?loc, NOW)$, and $Number(?e, ?num)$. Thus,

$$(Dir(?e, ?dir)\ IsEnemy(?e)\ At(?e, ?loc, NOW)\ Number(?e, ?num)) \qquad \text{(pc1)}$$

collectively establishes a context for each of the individual predicate.

The precondition trees can be used in collaborative constraints satisfaction. Suppose agents A_1, A_2 and A_3 share the precondition tree as shown in Fig.1, and A_3 is the doer of plan *RemoveThreat*. Assume both A_1 and A_2 have identified an enemy unit ($e1$) approaching A_3, who is now unaware of the threat from $e1$. Also assume A_1 can only observe the location, $At(e1, area_4, NOW)$, and moving direction, $Dir(e1, northeast)$, of $e1$; A_2 can only observe the enemy number, $Number(e1, 100)$, of unit $e1$. Obviously, neither A_1 nor A_2 alone can enable A_3 to do *RemoveThreat*. However, they can collaboratively satisfy A_3, because A_1 knows $At(e1, area_4, NOW)$ and $Dir(e1, northeast)$ will be useful for A_3 in the context $pc1$, and A_2 knows $Number(e1, 100)$ will be useful for A_3 in the same context.

4. Information Needs

4.1 Information and Incomplete Information

Information is defined in WordNet Dictionary as a message received and understood that reduces the recipient's uncertainty. We adopt the definition prescribed in the Open Archival Information System (OAIS) (OAIS, 1999): information is "any type of knowledge that can be exchanged, and it is always represented by some type of data".

To represent information, we start with the identifying reference expression (IRE), which is used to identify objects in appropriate domain of discourse(FIPA, 2002). For any predicate symbol p with arity n, it will be written in the form $p(\vec{?x}, \vec{c})$, where $\vec{?x}$ is a set of variables, $\vec{c}$ is a set of constants in appropriate domains, and their sizes sum to n. IRE is written using one of three referential operators defined in FIPA specification. $(iota\ \vec{?x}\ p(\vec{?x}, \vec{c}))$ refers to "the collection of objects, which maps one-to-one to $\vec{?x}$ and there is no other solution such that p is true of the objects"; it is undefined if for any variable in $\vec{?x}$ no object or more than one object can satisfy p (together with substitutions for other variables). $(all\ \vec{?x}\ p(\vec{?x}, \vec{c}))$ refers to "the collection of sets of all objects that satisfy p, each set (could be an empty set) corresponds one-to-one to a variable in $\vec{?x}$". $(any\ \vec{?x}\ p(\vec{?x}, \vec{c}))$ refers to "any collection of objects, which maps one-to-one to $\vec{?x}$, such that p is true of the objects"; it is undefined if for any variable in $\vec{?x}$ no object can satisfies p (together with substitutions for other variables). We will omit operator *any* if possible. Hence, expressions of form $(any\ \vec{?x}\ p(\vec{?x}, \vec{c}))$ can be simplified as $p(\vec{?x}, \vec{c})$.

Throughout this paper, we consider two forms of information: factual information and referential information. Factual information is represented as a proposition (predicate with constant arguments), and referential information is represented in terms of a special predicate $Refer(ire, obj)$, where ire is an identifying reference expression, and obj is the result of the reference expression ire evaluated with respect to a certain theory.

In the following we will use I $(I', I_1, \cdots)$ to represent the information to be communicated: when I refers to a proposition, the sender is informing the receivers that the predicate is true; when I refers to $Refer(ire, obj)$, the sender is informing the receivers that those objects in obj are what satisfy ire evaluated with respect to the sender's belief base.

Information can be classified along several dimensions. For instance, we can distinguish static information (which seldom changes once acquired, e.g., recipes for actions) from dynamic information. Depending on the ways of information acquisition, there are *observable* information, *computable* information (e.g., by inference rules), *a priori* information (common domain knowledge), etc. We focus on dynamic observable information, rather than static information.

4.1.1 Incomplete Information

Normally, in a referential information all the variables are bound with values. For example, $Refer(threat(e1, ?loc, ?dir, ?num), (area_1, south, 100))$. In multi-agent systems information exchange also involve incomplete information (with unbound variables), which is of special significance in teamwork settings. For instance, when a group of agents generate shared plans (Grosz & Kraus, 1999), parameter identification for team activities highly depends on the exchange of incomplete information.

Hence, we assume agents are capable of handling (recording and manipulating) incomplete information of the form $Q(\overrightarrow{v})$, where $\overrightarrow{v}$ is a vector of terms, each of which may be constant or variable[2]. If $Q(?x, ?y, c_1, c_2)$ belongs to agent A's belief base, it means agent A believes there exist some objects $?x$ and $?y$, which together with the already identified objects c_1 and c_2, have the relation described by Q. For the example in section 3, suppose agent B has observed an enemy unit e with unknown number of enemies is in area 4 and moving northeast toward agent A, and thus identified an *incomplete* threat $threat(e, area_4, northeast, ?num)$, B may inform such incomplete information to A immediately rather than wait until the number of e is acquired. There are many reasons for timely exchanging incomplete information: B may never be able to get the number of e for lack of observability, A may already get the number of e from another teammate, A may be able to deal with the threat even when it is incomplete, etc.

Incomplete information can be generated by reasoning appropriate precondition trees. For example, Fig.1 can be used to generate $threat(e, area_4, northeast, ?num)$ by fusing information $IsEnemy(e)$, $At(e, area_4)$, $Dir(e, northeast)$. Several pieces of incomplete information can be combined together if they are complementary. Continue the above example, suppose agent A also gets another incomplete threat $threat(e, ?loc, ?dir, 100)$ from agent C (i.e., the enemy unit e has 100 enemies, their location and moving direction are unknown to C), then $threat(e, area_4, northeast, 100)$ can be derived.

4.2 The Concept of Information Needs

An information-need may state that the agent needs to know the truth value of a proposition. For instance, suppose a person sends a query $Weather(Cloudy, Today)$ to a weather station. The weather station will realize that the person want to know, at least literally, whether today is cloudy [3]. More often than not, an agent may want to be informed of any information that matches his constraints, rather than simply querying whether a specific proposition is true or false. In particular, an agent may want to know the values of some arguments of a predicate, where the values could trusify the predicate. For example, a person may send a query $Weather(?x, Today)$ to a weather station, this will trigger the weather station, if it's benevolent, to inform the person about the (change of) weather conditions whenever necessary.

Thus, corresponding to "information", an expression for information-needs may also be in one of two forms: described either as a proposition, or as a reference expression (which

2. In implementation, the inference engine should treat incomplete information and complete information (e.g., facts) separately.
3. Refer to (Searle, 1975) for indirect speech acts.

actually specifies a class of information). In what follows N is used to refer to a (information) need-expression, $pos(N)$ ($ref(N)$) is true if N is a proposition (reference expression).

Now we come to the representation of information-needs. Obviously, an information need should specify the need-expression and the information consumer (needer). Typically a need becomes meaningless after a certain point (e.g., some event happens). For instance, an agent may no longer need to know the location of enemy units e, if e has already been defeated. Thus, information-needs often have an associated time-limit. In addition, a need is only applicable under certain context, which serves as relativizing conditions of the need, and/or records the reason of adopting the need. To combine them together, we introduce a modal operator $\mathsf{InfoNeed}(A, N, t, C_n)$ to denote information-needs. In case that N is a proposition, it means that agent A needs to know the truth value of N by t under the context C_n; in case that N is a reference expression, it means agent A needs to know those objects satisfying the reference expression N.

Note that the notion of information-needs cannot be defined simply as intentions of beliefs, say $\mathsf{Int.Th}(A, \mathsf{Bel}(A, I, t'), t, t_b, C_n)$, for at least two reasons. First, there are situations where an agent does need I but itself is not aware of such need. The notion in terms of intention has difficulty in expressing such situations clearly: on the one hand, $\mathsf{Int.Th}(A, \mathsf{Bel}(A, I, t'), t, t_b, C_n)$ has to be used to specify the information-need from external, but on the other hand, the agent cannot adopt the intention (otherwise it will know the need). Second, $\mathsf{Int.Th}$ is a mental attitude, and it will indirectly force the agent to consider certain means-end reasoning to bring about $\mathsf{Bel}(A, I, t')$. Such assumed, even $\mathsf{Bel}(B, \mathsf{Int.Th}(A, \mathsf{Bel}(A, I, t'), t, t_b, C_n), t)$ holds, agent B might not consider of helping A if B supposes that A could do certain actions (say, ask) to get I.

We then examine the properties of $\mathsf{InfoNeed}$. First $\mathsf{InfoNeed}$ is closed temporally into the past. That is, if an agent needs N by t, it also needs N any time before t (backward up tp the current time):

(T) $\mathsf{InfoNeed}(A, N, t, C_n) \Rightarrow \forall t' < t \cdot \mathsf{InfoNeed}(A, N, t', C_n)$.

It makes no sense to talk about negation of a reference need-expression. In case that the need-expression is a proposition, the information need is insensitive to negation:

(N) $\mathsf{InfoNeed}(A, N, t, C_n) \equiv \mathsf{InfoNeed}(A, \neg N, t, C_n)$, if N is a proposition.

Let $dom(?y)$ be the value domain of variable $?y$. Given vector $\vec{c}$, define

$$\vec{c} \setminus d \triangleq \begin{cases} (c_1, \cdots, c_i, c_{i+1}, \cdot, c_k) & \text{if } \vec{c} = (c_1, \cdots, c_i, d, c_{i+1}, \cdot, c_k) \\ \vec{c} & \text{if } d \text{ not occur in } \vec{c} \end{cases},$$

$$\vec{c} \oplus d \triangleq \begin{cases} (c_1, \cdots, c_k, d) & \text{if } \vec{c} = (c_1, \cdots, c_k) \\ (d) & \text{if } \vec{c} \text{ if empty} \end{cases}.$$

For information needs regarding reference need-expressions, abstract ones imply more concrete ones:

(S) $\mathsf{InfoNeed}(A, P(\vec{?x}, \vec{c}), t, C_i) \Rightarrow$
$\forall ?y \in \vec{?x} \forall k \in dom(?y) \cdot \mathsf{InfoNeed}(A, P(\vec{?x} \setminus ?y, \vec{c} \oplus k), t, C_i)$.

For example, if agent A knows agent B needs to know information about threat of form $Threat(?e, ?loc, ?dir, ?num)$, A may rationally assume B is also interested in more concrete information like $Threat(e1, ?loc, ?dir, 100)$, even though it's still incomplete.

$\mathsf{InfoNeed}$ distributes over conjunction and disjunction.

(C) $\mathsf{InfoNeed}(A, N_1 \wedge N_2, t, C_i) \Leftrightarrow \mathsf{InfoNeed}(A, N_1, t, C_i) \wedge \mathsf{InfoNeed}(A, N_2, t, C_i)$,

(D) $\mathsf{InfoNeed}(A, N_1 \vee N_2, t, C_i) \Rightarrow \mathsf{InfoNeed}(A, N_1, t, C_i) \vee \mathsf{InfoNeed}(A, N_2, t, C_i)$.

$\mathsf{InfoNeed}$ is reversedly closed under implication. That is, weaker information-needs entail stronger ones:

(I) $\mathsf{InfoNeed}(A, N_1, t, C_i) \wedge (N_2 \Rightarrow N_1) \Rightarrow \mathsf{InfoNeed}(A, N_2, t, C_i)$.

For example, believing agent B needs N_1 and "$N_2 \Rightarrow N_1$" is commonly known, it's rational for agent A to assume B also needs N_2, for otherwise B could have derived N_1 by itself.

Agents can partially figure out its own information needs by reflecting on incomplete information:

(R) $\mathsf{Bel}(A, P(?\vec{x}, \vec{c}), t) \Rightarrow \mathsf{Bel}(A, \exists t', C' \cdot \mathsf{InfoNeed}(A, P(?\vec{x}, \vec{c}), t', C'), t)$.

Oftentimes, an agent gets incomplete information from its teammates, who believe the beneficiary agent will need the information even though it is incomplete. The beneficiary agent will have to fill out or solicit for the missing part if itself does need the information which is currently incomplete. Only until this stage would the agent be able to identify the expiry time (t') and relevant context (C') so that the information need can evolve into a full-fledged one.

However, agents cannot reflect on its information-needs, because an agent may not be able to figure out its own information needs for many reasons. Thus,

$\mathsf{InfoNeed}(A, N, t', C_i) \not\Rightarrow \mathsf{Bel}(A, \mathsf{InfoNeed}(A, N, t', C_i), t)$,

$\neg\mathsf{InfoNeed}(A, N, t', C_i) \not\Rightarrow \mathsf{Bel}(A, \neg\mathsf{InfoNeed}(A, N, t', C_i), t)$.

In later sections we will introduce axioms that relate $\mathsf{InfoNeed}$ with intentional attitudes.

We now define a generated set. For any set of formula C, let $Needs(C)$ be a set of need-expressions generated from C:

1. $p \in Needs(C)$, if $p \in C$ is a proposition;
2. $(any\ ?\vec{x}\ p(?\vec{x})) \in Needs(C)$, if $p \in C$ is of form $p(?\vec{x})$ [4].

Need-expressions can be generated from action (or plan) preconditions. For action α, we write $Needs(\alpha)$ to refer to $Needs(pre(\alpha))$.

4.3 Levels of Information Needs

The notion of precondition-tree helps in dealing with levels of information needs. Because of properties (I) and (C) of $\mathsf{InfoNeed}$, there may exist information-needs at different levels but for the same purpose. For instance, suppose agent A recognized that an enemy unit e is approaching agent B, who needs to react to the threat (say, perform *RemoveThreat*) at least as late as time t'. Then A may think $\mathsf{InfoNeed}(B, \mathit{Threat}(e, ?loc, ?dir, ?num), t', C)$ holds, where C records A's explanation of holding this need. By property (I), A will hold $\mathsf{InfoNeed}(B, \mathit{IsEnemy}(e) \wedge \mathit{At}(e, ?loc) \wedge \mathit{Dir}(e, ?dir) \wedge \mathit{Number}(e, ?num), t', C)$, and by property (C), A will also hold

$\mathsf{InfoNeed}(B, \mathit{IsEnemy}(e), t', C)$, $\mathsf{InfoNeed}(B, \mathit{At}(e, ?loc), t', C)$,

4. Depending on domains, need-expressions of the form $(iota\ ?\vec{x}\ p(?\vec{x}))$ or $(all\ ?\vec{x}\ p(?\vec{x}))$ can also be generated. For instance, if α is a joint action where some doer should be exclusively identified, *iota* expression is preferred. *all* expression is suitable if all objects substitutable for variables in $?\vec{x}$ will be needed in the performance of α.

$\mathsf{InfoNeed}(B, Dir(e, ?dir), t', C)$, $\mathsf{InfoNeed}(B, Number(e, ?num), t', C)$.
Such proliferation process may continue in a top-down way along the precondition tree, which results in several levels of information-needs. Redundant assist would occur if A commits to satisfying all these information-needs.

To avoid unnecessary assist, agents could leverage precondition-trees by considering information-needs from the most abstract level first. Only when an agent cannot satisfy the information-needs at level i (i.e., there are critical information unknown) [5], will it consider those needs at level $i + 1$.

This use of precondition trees is highly related to information fusion. Level 2 processing of the JDL Data Fusion Process Model (Brooks & Iyengar, 1998) uses the raw data combined and refined at level 1 processing to develop a description of relationships among entities and to interpret the current situation. For instance, level 2 fusion can be used to develop an interpretation of the composition and disposition of the local threat forces and their current activities (Powell & Broome, 2002).

For the example shown in Fig.1, suppose agent A_2 is the doer of *RemoveThreat*, agent A_1's belief base includes facts like: $IsEnemy(e1)$, $At(e1, area_2, NOW)$, $Dir(e1, north)$, and $Number(e1, 80)$ (may be observed by A_1 itself or informed by others). To meet A_2's needs, A_1 may query its reasoning engine to check whether $Threat(?e, ?loc, ?dir, ?num)$ holds or not. According to the Horn clause with *Threat* as its head, a threat can be successfully identified from A_1's belief base. Here, we say A_1, through its reasoning engine, generated a threat information by fusing together relevant information. Alternatively, knowing what kinds of information are necessary in deriving a threat, and knowing that the inference knowledge (i.e., the relevant Horn clauses) is also shared by A_2, A_1 may choose to send the lower-level relevant information (i.e., *At*,*Dir*,*Number*), while leaving the task of fusion computing to A_2 itself. The difference of the two alternatives is who will do fusion computing, the information provider or the information consumer.

Information fusion can be carried out either by depth-first reasoning or by breath-first reasoning along precondition trees. Breath-first reasoning is preferred when the information consumer only has limited cognitive capacity, because it guarantees that the higher-level information will always be delivered with higher priorities than the lower-level information, which enables the information consumers consider higher-level information first.

4.4 Types of Information Needs in Agent Teamwork

A team is a set of agents having a shared objective and a shared mental state (Cohen, Levesque, & Smith, 1997). In the SharedPlans theory, shared objectives are given in terms of intentions-that (a team's wanting to do a certain team action), and shared mental states are reflected by partial shared plans (PSP) and the ultimate full shared plans (FSP). As well as establishing requisite mutual beliefs and ensuring the satisfaction of shared objectives, communication in effective agent teams also plays a central role in satisfying others' information needs. In agent teamwork, we distinguish four types of information-needs usually emerging in the pursuit of team or individual goals.

Action-performing information-need This type of information-needs enables an agent to perform certain (complex) actions, which contributes to the agent's individual com-

5. When incomplete information is allowed, an agent can *partially* satisfy an information-need.

mitments to the whole team. Typically, an action-performing information-need is derived from the preconditions of the action. For instance, in the example given in section 3, *Threat* is a kind of action-performing information-need with respect to action *RemoveThreat*.

Decision-making information-need As well as domain actions, those information-needs emerging in the mental action *decision-making* is of particular interest, which helps an agent to reduce uncertainty in the process of making decisions, and consequently enables the agent to rationally select a course-of-action (COA) from several potential choices. In the terminology of the SharedPlans theory, this kind of information-needs makes an agent better equipped to adopt an appropriate intention-to by reconciling those potential intentions-to. Typically, a decision-point has several branches to be explored, and each branch is associated with some *preference* constraints. For instance, preference conditions can be specified for each branch of a choice statement in MALLET(Yen, Yin, Ioerger, Miller, Xu, & Volz, 2001). Normally a default branch is pre-determined, and dynamic branch selection is triggered whenever information relevant to the preference conditions becomes available.

Many factors may affect the process of decision-making; the more factors are taken into consideration (i.e., the more relevant information is known), the less likely the decision-maker will make mistakes. For instance, in fire-rescue domains, fire-fighters normally use water to extinguish fires. Suppose a building containing materials that reacts with water is catching fire. It is highly needed to inform this to the fire fighters for them to decide on an appropriate COA; otherwise the consequence may be ruinous.

Goal-protection information-need This type of information-needs allows an agent to protect a goal (intention-that) from becoming unachievable. Information regarding potential threats to a goal belongs to this category; knowing such information will help an agent to adjust its behavior to either remove or avoid the threat to its goal. Information regarding conflicts between potential-intentions and the goal (i.e., adopted intention) also belongs to this category; knowing such information will help an agent to rationally postpone or drop those potential intentions. For instance, suppose the goal of logistics is to transport ammunition to the frontier, and the enemy units approaching the logistics introduces a threat to the logistics's goal. Then, the information regarding the enemy units (e.g., moving direction, etc.) is needed by the logistics in order to keep its goal persistent.

Goal-escape information-need A goal ultimately becomes achieved, unachievable or irrelevant. This type of information is needed by an agent to drop the impossible or irrelevant commitments (goals). A goal is achievable and relevant only when the context holds. Thus, typically goal-escape information-needs can be derived from the context of the goal under concern. If any part of the context no longer holds and this is observed by an agent, being helpful, the agent will inform this fact to the other teammates involved in the goal for them to abandon the goal.

5. Anticipate Information Needs

The concept of shared plans (recipes) in the SharedPlans theory offers us the basis for studying agents' anticipation capability regarding teammates' information needs. In this section, we propose some axiom schemas for agents to anticipate different types of information needs identified in the previous section. The ways of anticipating others' information needs proposed here may lay the foundation for developing algorithms for agents to dynamically reason about information-needs of their teammates.

5.1 Anticipate Action-performing Information Needs

Oftentimes an agent cannot proceed due to obstacles to individual or team actions. Here we focus on informational obstacles, which refer to the pre-requisite information for performing an action, and we assume they are specified as part of the preconditions of the action.

Intuitively we say an agent A can anticipate that another agent B will need to know the pre-requisite information for performing an action, if A recognizes that B has a potential intention to do that action. Formally,

Axiom 2 (Action-performing Information-Need)
$\forall A, B \in TA, \alpha, C_\alpha, t, t' \geq t, \forall N \in Needs(\alpha) \cdot$
$\mathsf{Bel}(A, \mathsf{Pot.Int.To}(B, \alpha, t, t', C_\alpha), t) \Rightarrow \mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t', C_n), t)$, *where*
$C_n = C_\alpha \wedge \mathsf{Pot.Int.To}(B, \alpha, t, t', C_\alpha)$.

Axiom 2 states that agent A believes that agent B will need the information described by N by t', if A believes that B is potentially intending to perform action α at time t' under context C_α. The context C_n of the information-need consists of C_α and B's potential intention to performing α.

To justify this axiom, some issues deserve further explanation. The first question is how an agent get to know others' intentions. We assume the agents in TA as a team are either evolving or acting on some shared plans collaboratively generated for some team task. To have a shared plan means all the team members have agreed with each other on some specific recipe. Even though individuals might have different partial view of the recipe, minimally an agent knows such information as the decomposition (at least at the immediate next level) of the actions he/she is (potentially) committed to (most typically jointly committed to together with some other teammates), the potential doers of each subaction, the performance sequence of those subactions, etc. Consequently, an agent can infer the (potential) intentions-to from his/her partial view of the recipe tree that all the teammates are focusing on.

A critical point made in the SharedPlans theory is that planning is interleaved with acting, which means agents can act on partial recipes, and a group of agents may not have a complete plan until after they have done some of the actions in the partial recipe (Grosz & Kraus, 1996). To have a partial shared plan means there are still some actions that remain to be resolved (i.e., role assignment) or decomposed further. For those unresolved actions, an anticipating agent cannot surely know who will be the actual performers; the best he can do is to assume all those agents with the specific capability would be the potential performers. This is the reason of using Pot.Int.To instead of Int.To for generality.

Second, it seems too strong that the axiom requires the anticipating agent know the preconditions of the action to be performed by other teammate. Usually, different agents may have different recipes for an action. Even though they do share some recipes, an agent may not know exactly which recipe will be used by another agent to achieve its commitment regarding the action. We relax this by letting the anticipating agent only consider those recipes itself is aware of (refer to the definition of $pre(\alpha)$ and $Need(\alpha)$). This means, the anticipation of others' needs is based on its own viewpoint by assuming itself as the performer. One drawback is that the anticipated information-needs may not reflect the real information-needs. This can be improved by allowing agents to exchange expertise on recipes. On the other hand, as a helping behavior, anticipating others' information-needs is not required to be always precise. Conversation sessions may be triggered when the beneficiary agent realizes its needs were incorrectly predicted, which helps the anticipating agent refine its model regarding others information-needs.

Third, the axiom indicates that an agent may generate one information-need for any need-expression in $Need(\alpha)$. Whenever communication bandwidth permitted, it could be leveraged to enhance team-wide situation awareness. However, most multi-agent systems only have restricted communication bandwidth. Moreover, according to the definition of $Need$, the set of need-expressions generated for an action could be large. Thus, certain meta-level strategies or assumptions common to all the teammates need to be employed to preclude unnecessary assist. As far as action-performing is concerned, an agent may not proceed when lacking of some pre-requisite information for performing some action; it may simply wait until more information becomes available (e.g., being informed by teammates). If such a "wait" semantics is taken as a common assumption among team members, it will be unnecessary for another agent to tell the action performer regarding the negative facts related to the preconditions. For example, suppose agent A requires p to be true prior to performing α. Agent B need not inform $\neg p$ to A when it believes p is false. Alternatively, an agent may proceed anyway even lacking of some pre-requisite information. If such a "risk" semantics is commonly assumed, agent B may want to let A know $\neg p$ hoping that A could choose more appropriate recipes if A knows this. Either semantics can be employed as a common knowledge when team activities start.

On the other hand, even though the set of anticipated information-needs is large, most of them will be dropped because the agent will ultimately make decisions on whether to react to (i.e., proactive communication) the anticipated information-needs based on such factors as the possibility that the prospective beneficiary agent already knows the information; the possible side-effects (e.g., overheard by opponents) of sending the information, etc.

Fourth, the context C_n is composed of agent B's potential intention under A's concern and the context of the potential intention. This is easy to justify because the anticipated information-need will make no sense if A no longer believes B has the potential intention, or from A's viewpoint the potential intention is no longer relevant. One question is to what extent agent A could get to know the intentional context C_α. As well as the pursue of higher-level joint goals, there are many private reasons for B to hold the potential intention regarding α. In section 2.2 we identified four possible uses of contexts: constraints, trace of explanation, attention management, and social specification. Among these four components, probably A has the best knowledge of the trace of explanation part due to the shared plans and shared recipes the whole team are working on. A will have no problem in knowing the social specification part of C_α if the social specifications of the whole team are taken as common knowledge. A may know some of the constraints on the action committed to by B. For example, suppose A and B are (potentially) committed to doing an action β by time t', which is composed of B's doing of α_1 followed by A's doing of α_2. To ensure success, both A and B knows that B has to finish doing α_1 by some time $t'' < t'$ so that A could be left with enough time to finish α_2. But in normal cases A is unaware of the constraints on B's action and B's strategies of attention-management regarding the action. Hence, when lack of information, the best A can do is to approximate B's intentional contexts based on its model of B; the context C_α here only reflects A's approximation, which may be different from the actual one that B is holding.

Lemma 1 $\forall A, B \in TA, \phi, \alpha, C_\phi, \Theta_\alpha, t, t' \geq t, t'' \geq t', \forall N \in Needs(\alpha)\cdot$
$\mathsf{Bel}(A, \mathsf{Int.Th}(B, \phi, t, t'', C_\phi), t) \wedge \mathsf{Bel}(A, \neg\mathsf{Bel}(B, \phi, t), t) \wedge$
$\mathsf{Bel}(A, \mathsf{Lead}(B, \alpha, \phi, t', t, \Theta_\alpha), t) \Rightarrow \exists C_n \cdot \mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t', C_n), t).$

Proof. Follows directly from axiom 1 and 2.

5.2 Anticipate Decision-making Information Needs

An agent A may be able to recognize the information-needs of another agent B if A knows B is facing a critical decision point for choosing the next course of action to fulfill its commitment. This takes place when A knows the possible choices B is reckoning and A knows, viewing itself as the decision-maker, that some information is essential for making a better decision on these choices. Being helpful, A will anticipate that these information will also be needed by B to make the decision. Let $reckon(A, \Omega, \phi)$ refer to the set of information used by A in evaluating the utilities of those actions in Ω with respect to the goal state ϕ.

Axiom 3 (Decision-making Information-Need)
$\forall A, B \in TA, \phi, C_\phi, t, t' \geq t, t'' > t', N, \Omega\cdot$
$\mathsf{Bel}(A, \bigwedge_{\alpha_i \in \Omega}[\mathsf{Pot.Int.To}(B, \alpha_i, t, t', C_{\alpha_i}) \wedge \mathsf{Int.Th}(B, \phi, t, t'', C_\phi) \in C_{\alpha_i}], t) \wedge$
$\mathsf{Bel}(A, N \in reckon(A, \Omega, \phi), t) \Rightarrow \mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t', C_n), t),$ *where*
$C_n = C_\phi \wedge \bigwedge_{\alpha_i \in \Omega} \mathsf{Pot.Int.To}(B, \alpha_i, t, t', C_{\alpha_i}) \wedge \mathsf{Int.Th}(B, \phi, t, t'', C_\phi).$

Axiom 3 states that in case that agent A believes that agent B is considering several potential actions in its pursue of some adopted commitment ϕ, it will assume B will need information N if from A's viewpoint, N does affect the evaluation of those potential actions as far as ϕ is concerned. The context of the information-need consists of B's chosen intention and its context, as well as those potential intentions.

To justify axiom 3, we need to answer the question of when an agent A can get to know another agent B's potential intentions, which are not observable yet. It's easy if both A and B are involved in the same shared recipes where the decision-points are explicitly specified. Knowing B's role in the team activities under concern and the behavior patterns common to that role, A could also recognize or predict B's possible courses of action based on B's past and current behaviors.

Axiom 3 is useful in helping others evaluate multiple (typically exclusive) potential intentions to see which one works better in fulfilling their goals.

5.3 Anticipate Goal-protection Information Needs

The first type of goal-protection information-needs is those information needed to reconcile a potential intention with an already adopted intention. Specifically, if A knows B has a chosen goal ϕ and is potentially to do action α, A will assume B needs to know N (for dropping the potential intention) in case that when N holds, B's commitment to doing α will make ϕ impossible. Upon knowing N, B will abandon the potential intention to do α.

Axiom 4 (Goal-Protection Information Needs 1)
$\forall A, B \in TA, \phi, C_\phi, t, t' \geq t, t'' > t', t_1 < t', N, \alpha, C_\alpha \cdot$
$\mathsf{Bel}(A, \mathsf{Pot.Int.To}(B, \alpha, t, t', C_\alpha) \wedge \mathsf{Int.Th}(B, \phi, t, t'', C_\phi), t) \wedge$
$\mathsf{Bel}(A, N \wedge \mathsf{Int.To}(B, \alpha, t_1, t', C_\alpha) \Rightarrow \neg\phi), t) \Rightarrow \mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t_1, C_n), t)$, *where*
$C_n = C_\phi \wedge C_\alpha \wedge \mathsf{Pot.Int.To}(B, \alpha_j, t, t', C_\alpha) \wedge \mathsf{Int.Th}(B, \phi, t, t'', C_\phi)$.

For example, suppose in fire-rescue domains, $N = \mathit{has\_chemical}(T1, M1)$, which means the building $T1$ contains chemical material $M1$, which produces noxious vapor when reacting with water. As an engineer of the building, A knows the fact N; but the fire fighters are unaware of this fact, and they always use water by default. Herein, the fire fighters have a goal to put out the fire on $T1$ with minimum loss, and are potentially intending to extinguish the fire using water. In such a case A is obligated to let the fighters know N so that they can drop the potential intention of extinguishing the fire using water. N could also be treated as decision-making information-need generated from axiom 3 when A knows there are more than one potential choice under the fighters' concern.

The first type of information needed for goal-protection purpose is related to internal threat: lack of the information may allow an agent to do actions that would prohibit itself from fulfill its own chosen goal. The second type of information needed for goal-protection is related to external threat: lack of the information may allow agents in an opposite team to do actions that would prohibit the agent from fulfilling its chosen goal.

Axiom 5 (Goal-Protection Information Needs 2)
$\forall A \in TA, B \in TA, \phi, C_\phi, N, t, t'' > t, \forall G \in TB, \alpha, t_1 < t'' \cdot$
$\mathsf{Bel}(A, \mathsf{Int.Th}(B, \phi, t, t'', C_\phi), t) \wedge$
$\mathsf{Bel}(A, \exists C' \cdot \mathsf{Pot.Int.To}(G, \alpha, t, t_1, C'), t) \wedge \mathsf{Bel}(A, \exists \Theta_\alpha \cdot \mathsf{Do}(G, \alpha, t_1, \Theta_\alpha) \Rightarrow \neg\phi, t) \wedge$
$\mathsf{Bel}(A, [\exists \beta, \Theta_\beta, t_b < t_1 \cdot \mathsf{Bel}(B, N, t_b) \wedge \mathsf{Do}(B, \beta, t_b, \Theta_\beta) \Rightarrow \nexists R, \Theta_\alpha \cdot \mathsf{CBA}(G, \alpha, R, t_1, \Theta_\alpha)], t)$
$\Rightarrow \mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t_1, C_n), t)$, *where*
$C_n = C_\phi \wedge \mathsf{Int.Th}(B, \phi, t, t'', C_\phi) \wedge [\exists C' \cdot \mathsf{Pot.Int.To}(G, \alpha, t, t_1, C')]$.

Axiom 5 says that if A knows (1) B has a chosen goal ϕ; (2) an agent G in an opposite team potentially intends to do action α, the doing of which will make ϕ impossible; (3) in case that B knows N and performs some action β timely, it will disable G in performing α, then, A will assume N would be needful for B to deal with the external threat. The context of the information-need consists of agent B's chosen goal and the embedded context, and the potential intention from agent G.

It is worth noting that the anticipating agent A need not know which action agent B will choose to respond to the coming threat. Thus, it leaves open the possibility of searching for recipes (plans) to avoid the threat. On the other hand, this offers B the flexibility of choosing one from several possible reactions. Axiom 5 will further elicit the anticipation of action-performing information-needs, once it becomes clear to agent A that agent B will adopt a particular action ($\mathsf{Int.To}$) to deal with the threat.

5.4 Anticipate Goal-escape Information Needs

It could be the case that if an agent did not know that the context (or escape) condition had changed status, the agent might take actions that would foil the mission of the whole team. Axiom 6 states that if agent A believes that agent B has a goal towards ϕ, it will assume B will need information described by N, which is generated from the context of B's intention. The context of the information-need consists of C_ϕ and B's intention.

Axiom 6 (Goal-escape Information-Need)
$\forall A, B \in TA, \phi, C_\phi, t, t' \geq t \forall N \in Needs(C_\phi)\cdot$
$\mathsf{Bel}(A, \mathsf{Int.Th}(B, \phi, t, t', C_\phi), t) \Rightarrow \mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t', C_n), t)$, *where*
$C_n = C_\phi \wedge \mathsf{Int.Th}(B, \phi, t, t', C_\phi)$.

The joint-intention theory requires that all the agents involved in a joint persistent goal (JPG) take it as an obligation to inform other agents regarding the achievement or impossibility of the goal. Such requirement on communication among teammates is necessary to model coherent teamwork, but in the real case it's too strong to achieve effective teamwork; enforced communication is not necessary, and even impossible in time-stress domains. Rather than forcing agents to communicate, axiom 6 allows an agent to anticipate the information-needs of others regarding the context conditions of their goals, which may or may not result in communicative actions, depending on whether it's possible for the agent to do those helpful behaviors.

Like action-performing information-needs, for goal-escape information-needs, certain strategies can also be used to preclude unnecessary assist. For instance, suppose $C_\phi = \{\psi_1, \neg\psi_2\}$. If by default all the components of a context are commonly assumed to be true, then only when ψ_1 becomes false or ψ_2 becomes true, will an agent consider informing the information to others.

5.5 Self-reflect on Information Needs

When an agent intends to do some action but lack the pre-requisite information, it could simply wait until some of its teammates can anticipate its information-needs and provide help timely. It will be more flexible if an agent can get to know its own information-needs by self-reflection. For instance, being aware of its own information-needs, an agent may choose to proactively request assistance from teammates; subscribe its information-needs from a known information provider, etc.

When the generation set of need-expressions is explicitly given, it's possible for an agent to anticipate its own information-needs. For instance, agent names A and B in axiom 2 and axiom 6 may refer to the same agent, which characterize how an agent reflects on its own information-needs. However, there are many cases wherein it is impossible for an agent to know its own information needs by reflection. For instance, usually decision-making information-needs and goal-protection information-needs are difficult for an agent to anticipate by its own due to lack of expertise or observability. Sometimes, even though they can, they may not know to whom to ask. In such cases, teammates' anticipation and proactive assist play a critical role.

6. Commit to Other's Information Needs

When an agent recognizes the information-needs of its teammates by being informed or by anticipating, it will consider providing help if possible. A critical issue here is to relate an agent's belief about the information-needs of teammates to intentions to help. One may be tempted to establish this linkage using an axiom similar to axiom 1: If (1) agent A believes agent B has an information-need, (2) A believes B does not have the information, and (3) the performance of some action β can lead to B's awareness of the information, then A will consider to do β. However, this seemingly intuitive approach has two drawbacks: (1)it requires the action β be explicitly specified, (2)the persistence of the helpful commitment regarding the information-needs is not explicitly specified.

One more general approach is to make abstract rather than specific commitments for satisfying others' information needs, and postpone the specific commitments (and their reconciliation) to later stages. In this way, the commitment to providing help can be clearly separated from the decisions on how to provide help, which is more flexible in implementing agent teams with multiple proactive behaviors.

Let B_A be the belief base of agent A, and $\mathsf{B}_A \models p$ means p is a logical consequence of B_A. For any agent A and need-expression N, function $info(A, N)$ returns the information with respect to N evaluated by A:

$$info(A,N) \triangleq \begin{cases} N & \text{if } \mathsf{B}_A \models N, \text{ and } N \text{ is a proposition,} \\ \neg N & \text{if } \mathsf{B}_A \models \neg N, \text{ and } N \text{ is a proposition,} \\ Refer(N,Q) & \text{if } N = (iota\ ?\vec{x}\ p(?\vec{x})), \\ & Q \in \Sigma = \{\theta \cdot ?\vec{x} : \mathsf{B}_A \models \theta \cdot p, \theta \text{ is most general} \\ & \ \text{substitution (mgs)}\}, \text{ and } \Sigma \text{ is singleton,} \\ Refer(N,Q) & \text{if } N = (any\ ?\vec{x}\ p(?\vec{x})), \\ & Q \in \Sigma = \{\theta \cdot ?\vec{x} : \mathsf{B}_A \models \theta \cdot p, \theta \text{ is mgs}\} \neq \emptyset, \\ Refer(N,\Sigma) & \text{if } N = (all\ ?\vec{x}\ p(?\vec{x})), \\ & \Sigma = \{\theta \cdot ?\vec{x} : \mathsf{B}_A \models \theta \cdot p, \theta \text{ is mgs}\}, \end{cases}$$

$info(A, N)$ is undefined in case that N is a proposition, but neither $\mathsf{B}_A \models N$ nor $\mathsf{B}_A \models \neg N$; or in case that $N = (any\ ?\vec{x}\ p(?\vec{x}))$ but $\Sigma = \emptyset$; or in case that $N = (iota\ ?\vec{x}\ p(?\vec{x}))$ but Σ is not a singleton. In case that $N = (any\ ?\vec{x}\ p(?\vec{x}))$ and $|\Sigma| > 1$, a randomly selected element of Σ is returned. Proposition $defined(info(A, N))$ is true when $info(A, N)$ is defined, and false otherwise.

The following axiom says that, when an agent comes to know another agent's information needs, it will adopt an attitude of potential intention-that towards "the other's belief about the needed information". It is worth noting that even if A is unaware of the information needed by B, it can still adopt an intention to help which might lead it to engage other agents in helping (e.g., forward the information-need to another agent).

Axiom 7 (ProAssist) $\forall A, B \in TA, N, C_n, t, t' > t \cdot$
$\mathsf{Bel}(A, \mathsf{InfoNeed}(B, N, t', C_n), t) \Rightarrow$
$\quad [defined(info(A, N)) \Rightarrow \mathsf{Pot.Int.Th}(A, \mathsf{Bel}(B, info(A, N), t'), t, t', C_n)] \vee$
$\quad [\neg defined(info(A, N)) \Rightarrow \mathsf{Pot.Int.Th}(A, defined(info(B, N)), t, t', C_n)].$

We use Pot.Int.Th rather than Pot.Int.To in the axiom because Pot.Int.To requires the agent adopt a specific action to help the needer, while Pot.Int.Th offers the agent with the flexibility in choosing how to help. Note that A and B could refer to the same agent, that means agent A will try to help itself by adopting appropriate intentions. Axiom 7 relates information-needs with potential intentions-that. It, together with axiom 1, specifies how an agent choose appropriate actions to satisfy its own or other's information-needs.

Being aware of others' information needs does not entail helping behavior. Many factors may prevent an agent from really making the commitment. For instance, an agent is just too busy. This is the reason why we choose Pot.Int.Th rather than Int.Th which offers agents with the flexibility of deciding whether to help. Once the Pot.Int.Th is reduced to Int.Th, the agent is committed to retrying until either the information needer is satisfied or the information-need is no longer relevant.

Furthermore, according to axiom 1, the beliefs of information-needs are ultimately reduced to *Pot.Int.To*. This enables the framework to specify the situations in which an agent could reflect on its assist behaviors, yet leaving open the agent's commitment on such behaviors. When an agent faces multiple assist opportunities, it will not restrict agents to commit to specific assist opportunity.

7. Discussion

Proactive information delivery behavior was recognized by researchers in the studies regarding indirect speech acts (Searle, 1975) in the field of human discourse understanding (Allen, 1983; Allen & Perrault, 1980; Litman & Allen, 1990). Indirect speech acts are those that appear to mean one thing yet are treated as though they mean something else. In (Allen, 1983), based on a plan-recognition model of the language comprehension process, Allen explained why a hearer could generate helpful responses that convey more information to the speaker than was explicitly requested.

Allen's observations of helping behavior regarding other's lack of information within human dialogs certainly shed light on the study of proactive information delivery behavior. Nevertheless, proactive information delivery becomes more involuted in multi-agent teamwork settings. First, in human dialogs, indirect speech acts can be understood by considering the *idiomatic* meaning in addition to the literal meaning, using inference schema (i.e., to rate the potential choices by heuristics or inference rules), or using background/context knowledge to infer other's intentions (Brown, 1980; Allen, 1983), etc. When it comes to model proactive information delivery in large teams (probably mixed with human and software agents), more subtle issues need to be considered, such as the level of abstraction, shared mental states, computational complexity, etc. Second, Allen's work heavily relies on the audience's recognition of the speaker's intentions and plans based on certain rules and heuristics. While the idea of modeling discourse understanding as plan recognition is reasonable from the standpoint of human discourse, for large agent teams, it's not only impractical because each member needs to keep recognizing other teammates' plans from plenty of potential choices (Cohen & Levesque, 1990b) [6], but also difficult (e.g., how to match each other's reasoning) because the divergency among agents in anticipating a certain agent's intents may impact the performance of the whole team, even inhibit the teamwork. Third,

6. It's still computational hard even the *intended recognition assumption* (Cohen, 1981) is adopted.

the distinction of indirect and direct speech acts existing in human discourse is no longer that important for teams facing overwhelming amount of information under time pressure. Most likely, proactive information delivery is no longer triggered by the understanding of the implicit meaning conveyed by the preceding speech act (i.e., the ask/reply mode, also known as *master-slave assumption* (Grosz & Sidner, 1990)) under certain discourse, but triggered by the anticipation of other's needs even *without* any preceding conversation session with that agent regarding its needs (directly or indirectly).

Another significant thread of research in human dialogs is to explain certain properties of discourse using the notion of SharedPlan. In this view, the participants in a discourse mutually believe they are working towards establishing the beliefs and intentions that are necessary for one to say that they have a shared plan (Grosz & Sidner, 1990). Proactive behavior is implicitly captured in their second conversational default rule (CDR2) (Grosz & Sidner, 1990), which states that an agent in a group will adopt an intention to do an action if the performance of the action contributes to the achievement of the group's joint goal. Proactive information delivery can thus be taken as one reification of the schematic rule with appropriate communicative actions as the substitutes. Such an attempt of using SharedPlan can be traced to Lochbaum's work (Lochbaum, 1994, 1995).

The SharedPlan theory seems promising because it does support the shared awareness of team activities owing to the concept of shared plans. Also, proactive information delivery entailed by the SharedPlan theory is abstract, and perhaps intuitively obvious. However, this view is not particularly satisfying. Needed are models that are not too vague or otherwise uninformative nor too constrained or otherwise too specific (Bach, 1990). To the extent possible, we would like a framework where information needs can be taken as first-class objects that are finer-grained than overall observations offered by the SharedPlan theory. In addition, the proactiveness in Lochbaum's work (Lochbaum, 1995) relies on discourse understanding. From our viewpoint, this shares the same limitation as exposed in Allen's work, i.e., it requires the information provider can infer the speaker's needs from the *preceding* utterances between them. The more interesting behavior is, an agent could anticipate other's needs internally and *pushing* the relevant information rationally.

Information pushing is certainly related to proactive information delivery. Information pushing refers to the behavior of delivering information to a user based on the personalized profile specific to that user (an effective design typically defines what information is needed and when it is needed), which has been widely adopted by Web-based information services. Information is only delivered to a user if and only if it fits the personalization criteria set by the user. The criteria could include complicated and dynamic metrics to ensure that users are not "spammed" (Albers, 1998). It even can be automatically updated (i.e., learned) from user's behaviors.

Although proactive information delivery and personalized information pushing are similar in that they both send information to an information consumer in a proactive way based on the anticipation of his/her information needs, they do differ in several aspects. For instance, the former requires a more abstract but broader understanding about the information consumer (e.g., a shared awareness of the team goals, the planned team activities, each other's roles and responsibilities, etc.). Also, proactive communications are bi-directional in a team, whereas personalized information pushing is only from the computer to the user. However, proactive information delivery can be viewed as a general extension of personalized information pushing in the context of teamwork. Thus, feasibility and rationality, as suggested by the practice of information pushing, should be considered in developing theories for proactive information delivery.

Reasoning about information-needs is the first step of establishing a theory for proactive information delivery behaviors among agents in a team. Such a theory has several important benefits. First, it allows an agent to deliver needed information to teammates who could not have requested the information themselves due to their limited sensing capabilities or their incomplete knowledge about the environment. Second, even though broadcasts can be used to deliver information, it would have resulted in an overwhelming amount of information for agents to process. Delivering only the information relevant to the needs of teammates is promising in addressing the issue of information overload. Third, agents committed to others' needs can continuously monitor the environment for detecting changes relevant to the needs of the teammates. Consequently, each agent in a team may become a potential source for pushing information to others. Fourth, it allows an agent to automatically terminate its "monitoring" activity for a teammate's information-need when the need becomes irrelevant (e.g., the context of the need is no longer valid). Finally, it is desirable for the theory to support not only exchanges of information but also flows of information-needs. This will enable agents in a team to establish a "shared mental model" regarding others' needs. Such a shared mental model is important for further enhancing a team in their intelligent information exchange.

8. Conclusion

In this paper we introduced a modal operator to represent agents' information-needs; investigated different levels of information needs using the idea of precondition-tree; identified several types of information-needs prevalent in agent teamwork; provided and justified the axioms for anticipating others' information-needs; and introduced an axiom for enabling agents to commit to helping others with their information-needs. Such a formal framework allows agent systems to explicitly represent and reason about information-needs, and further facilitates an agent team to establish a shared mental model regarding their information-needs.

The SharedPlans theory developed by Grosz and Kraus is chosen as one of the cornerstones of our framework. By exploring the potential axioms leading to communicative actions, the work presented in this paper actually extended the SharedPlans formalism, and it moves a step forward to the goal established in (Grosz & Kraus, 1996): to develop a more complete set of communication axioms in SharedPlans theory for establishing requisite mutual beliefs and ensuring the satisfaction of intentions-that.

There are several important issues remained to be elucidated. For instance, an agent may get overloaded by adopting too many commitments. It is worthwhile to investigate the effects on team performance of different ways by which an agent resolves the conflicts between helpful commitments (e.g., proactive communicative actions) and its own responsibilities. In addition, proactive information delivery behaviors among teammates improve team intelligence but may inevitably introduce redundant information exchanges, because multiple agents in a team might deliver the same piece of information to the information needer. How to reduce redundancy in proactive information delivery among teammates also deserves future research.

We believe agents empowered with proactive information delivery capabilities can be used to better simulate, train, or support the information fusion, interpretation, and decision-makings of agent teams that may include both human agents and software agents. The long-term goal of our research is to develop agent theories and technologies related to proactive teamwork behaviors. The work described in this paper establishes a formal basis for achieving this goal.

Acknowledgments

This research has been supported by AFOSR MURI grant No. F49620-00-1-0326.

References

Albers, M. J. (1998). Goal-driven task analysis: improving situation awareness for complex problem-solving. In *Proceedings of the 16th annual international conference on Computer documentation*, pp. 234–242, Quebec, Canada.

Allen, J. (1983). Recognizing intentions from natural language utterances. In Brady, M., & Berwick, R. C. (Eds.), *Computational Models of Discourse*, pp. 107–166. MIT Press, Cambridge, MA.

Allen, J., & Perrault, C. (1980). Analyzing intention in utterances. *Artificial Intelligence*, *15*, 143–178.

Bach, K. (1990). Communicative intentions, plan recognition, and pragmatics: comments on Thomason and on Litman and Allen. In Cohen, P., Morgan, J., & Pollack, M. (Eds.), *Intentions in communication*, pp. 389–400. MIT Press.

Brandenburger, A., & Dekel, E. (1993). Hierarchies of beliefs and common knowledge. *Journal of Economic Theory*, *59*, 189–198.

Brooks, R. R., & Iyengar, S. S. (1998). *Multi-Sensor Fusion: Fundamentals and Applications*. Prentics Hall, New Jersy.

Brown, G. (1980). Characterizing indirect speech acts. *American Journal of Computational Linguistics*, *6*(3-4), 150–166.

Canon-Browers, J., & Salas, E. (1997). A framework for developing team performance measures in training. In Brannick, M., & et. al. (Eds.), *Team performance Assessment and Measurement: Theory, Methods, and Applications*. Lawrence Erlbaum Associates: Hillsdale, NJ.

Cohen, P. R., & Levesque, H. J. (1990a). Performatives in a rationally based speech act theory. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 79–88.

Cohen, P. R., & Levesque, H. J. (1990b). Rational interaction as a basis for communication. In *Intentions in Communication*, pp. 221–255. MIT Press.

Cohen, P. R., Levesque, H. J., & Smith, I. A. (1997). On team formation. In Hintikka, J., & Tuomela, R. (Eds.), *Contemporary Action Theory*.

Cohen, P. (1981). The need for identification as a planned action. In *Proceedings of the seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C.

Davis, E. (1994). Knowledge preconditions for plans. *Journal of Logic and Computation*, *4*(5), 721–766.

Fagin, R., Halpern, J. Y., Moses, Y., & Vardi, M. (1995). *Reasoning About Knowledge*. MIT Press.

FIPA (2002). Agent communication language specification. In *http://www.fipa.org/*.

Grosz, B., & Kraus, S. (1996). Collaborative plans for complex group actions. *Artificial Intelligence*, *86*, 269–358.

Grosz, B., & Kraus, S. (1999). The evolution of sharedplans. In Rao, A., & Wooldridge, M. (Eds.), *Foundations and Theories of Rational Agencies*, pp. 227–262.

Grosz, B., & Sidner, C. (1990). Plans for discourse. In Cohen, P., Morgan, J., & Pollack, M. (Eds.), *Intentions in communication*, pp. 417–444. MIT Press.

Litman, D. J., & Allen, J. F. (1990). Discourse processing and commonsense plans. In Cohen, P., Morgan, J., & Pollack, M. (Eds.), *Intentions in communication*, pp. 365–388. MIT Press.

Lochbaum, K. E. (1994). *Using Collaborative Plans to Model the Intentional Structure of Discourse*. PhD thesis, Harvard University, Tech Report TR-25-94. Cambridge, MA.

Lochbaum, K. E. (1995). The use of knowledge preconditions in language processing. In Mellish, C. (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1260–1266, San Francisco. Morgan Kaufmann.

Morgenstern, L. (1987). Knowledge preconditions for actions and plans. In *Proceedings of IJCAI-87*, pp. 867–874.

OAIS (1999). Reference model for an open archival information system. In *http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf*.

Pollack, M. E. (1990). Plans as complex mental attitudes. In Cohen, P., Morgan, J., & Pollack, M. (Eds.), *Intentions in communication*. MIT Press.

Powell, G. M., & Broome, B. (2002). Fusion-based knowledge for the objective force. In *National symposium on Sensor and Data Fusion*.

Rouse, W., Cannon-Bowers, J., & Salas, E. (1992). The role of mental models in team performance in complex systems. *IEEE Trans. on Sys., man, and Cyber*, *22*(6), 1296–1308.

Searle, J. (1975). Indirect speech acts. In Cole, P., & Morgan, J. (Eds.), *Syntax and semantics. III. Speech acts*, pp. 59–82. NY: Academic Press.

Sycara, K., & Lewis, M. (1991). Forming shared mental models. In *Proceedings of the 13th Annual Meeting of the Cognitive Science Society*, pp. 400–405.

Sycara, K., & Lewis, M. (1994). Modeling teams of specialists. In *Proceedings of the 27th Hawaii International Conference on System Sciences*. IEEE Society Press.

Tidhar, G., Heinze, C., & Selvestrel, M. (1998). Flying together: Modeling air mission teams. *Journal of Applied Intelligence*, *1*(1), 1–1.

Vidal, J., & Durfee, E. (1996). The impact of nested agent models in an information economy. In *Proceedings of the ICMAS'96*, pp. 377–384.

Yen, J., & Fan, X. (2002). The semantics of proactive communication acts among team-based agents. In *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, pp. 447–454.

Yen, J., Fan, X., & Volz, R. A. (2002). On proactive delivery of needed information to teammates. In *Proceedings of the Workshop on Teamwork and Coalition Formation at AAMAS'02*, pp. 53–61.

Yen, J., Yin, J., Ioerger, T., Miller, M., Xu, D., & Volz, R. (2001). CAST: Collaborative agents for simulating teamworks. In *Proceedings of IJCAI'2001*, pp. 1135–1142.

Yin, J., Miller, M. S., Ioerger, T. R., Yen, J., & Volz, R. A. (2000). A knowledge-based approach for designing intelligent team training systems. In *Proceedings of the Fourth International Conference on Autonomous Agents*, pp. 427–434.