A Formal Model of Responsibilities in Agent-Based Teamwork

Thomas R. Ioerger and J. Colby Johnson
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112
{ioerger,jcj0650}@cs.tamu.edu

Abstract ***Responsibilities are an important component of teamwork in multi-agent systems. In this paper, we present a formal model of responsibilities based on beliefs and goals. Our approach focuses on two central features of responsibilities: persistence and dependency. We define a special form of persistence that obligates agents to keep trying an action until it is done, but allows the caveat that they might delegate the task. Dependency means that the responsibility was assigned by a delegating agent who wants it done. We model this dependency through a requirement to maintain mutual belief on the status of the goal, as well as the need by the superior, and the achievability by the subordinate. Thus agents with responsibilities are guaranteed to inform their superiors when they are about to fail, allowing others to take over, which provides robustness. Finally, we treat joint responsibilities among multiple agents as a mutual agreement to keep each other informed of their individual abilities to achieve the goal. These definitions provide a semantics for responsibilities in multi-agent systems, and can be used to design correct specifications of agent behavior, and justify decisions such as when to act, wait, or communicate within a team.***

Keywords: autonomous agents, multi-agent systems, teamwork, responsibilities

1 Introduction

Responsibilities are an essential component of teamwork in human teams. [1]. In many domains, the structure of a team is often broken-down into a set of pre-defined roles, with each role having a specific set of responsibilities associated with it (for example, the setters, spikers, server, and captain on a volleyball team). This structure can increase the efficiency of teamwork by reducing the need for constant communication–much of the coordination can be done implicitly through role assignment. The concept of responsibilities can also be of great benefit for designing teams of agents in a multi-agent system [2; 3]. Descriptions of responsibilities in a team can enable agents to make decisions about what each team member is expected to do, and to anticipate the actions of others.

We would like to be able to encode the structure of a team, including roles and responsibilities, in a formal knowledge representation language for constructing multi-agent systems [4]. In such systems, agents will need to be able to reason dynamically about the team and make decisions about what actions to take, which strongly depends on the semantic interpretation underlying the notion of responsibilities. In this paper, we give a preliminary formal account of responsibilities in terms of the mental state of agents (i.e. beliefs and goals) that has practical implications for implementing teamwork in multi-agent systems.

We begin by noting that there are two possible perspectives for thinking about responsibilities: retroactive and proactive. In the *retroactive* point of view, responsibility is a relationship between an agent and some event that happened in the past, and seeks to assign blame or fault (i.e. “culpability”). Generally, this involves establishing criteria such as causality and non-inevitability [5]. However, for implementing teamwork in multi-agent systems, we are primarily concerned with the *proactive* point of view: what does knowledge of a responsibility mean that an agent should do now (and in the future), i.e. how does it affect decision-making?

In our model, we attempt to capture two essential aspects of the mental state associated with responsibilities that happen to have significant relevance to agent behavior: persistence and dependency. Agents who have a responsibility should maintain a certain degree of *persistence* in accomplishing this goal. However, it is not quite the same as a commitment or intention [6]. For example, an intention to do something requires one to do the action oneself, whereas carrying out a responsibility can involve having someone else do it, e.g. through delegation [7]. In fact, an important difference that we want to capture in our model is the way that responsibilities are “transferrable,” whereas intentions are not. Still, having a responsibility *in certain cases* requires one to keep trying to achieve something until it is done. Norman and Reed [8] address this through the use of deontic logic (essentially expressing preferences among possible world states) to capture the unique sense of obligation imposed via delegation

However, this non-standard form of persistence alone is not sufficient to characterize responsibilities because it does not express the nature of the relationship between the agent who has the responsibility and the agent who gave (delegated) it to them. The delegating agent wants the goal achieved, and knowledge of this *dependency* has important implications for the mental state of the agent with the responsibility. Most importantly, we will show that this knowledge obligates the agent to inform its superior if it comes to realize that it will never be able to carry out its responsibility (i.e. will fail). This constraint is crucial for generating non-trivial teamwork behavior in synthetic multi-agent systems because it drives just the right kind of communication to allow teams to re-organize and dynamically respond to unexpected changes in conditions. For example, it gives a formal foundation for the triggering of agents playing "backup" roles, which can produce more robust overall performance by the team. The concept of dependency (and delegation, and reporting of failures) is also of central importance for properly handling various forms of *joint*, or shared, responsibilities.

We note that our concept of responsibility within agent teams is restricted, in that it does not cover the wide variety of other senses of the term "responsibility." For example, one sense is "accountability," which involves being prepared to answer for or provide a justification of actions when requested by an authority. Also, there are complex notions of being responsible for another person or their development (as in a child or employee), or being a "responsible citizen," which has to do with obeying social norms (like cleaning up, or reporting theft). Some models of responsibility take this sociological view by relating responsibilities to other complex inter-personal relationships, such as trust, reputation, or power [9]. While these views are certainly important for understanding responsibility in the more general sense (a complete axiomatization of "responsibility" would no doubt be interesting, if possible), most of these alternative notions are not needed for interpreting responsibility assignments in artificial mutli-agent systems. Because the design of the agents is under the control of the implementer, we can assume the agents are all honest and cooperative[1]. We are interested in developing practical methods for building multi-agent systems that produce intelligent behaviors and solve complex problems through teamwork. The important question from a computational point of view is: under normative conditions (i.e. with trustworthy and like-minded teammates), what does a set of responsibilities given in a team definition prescribe that each agent should do at each time step?

In this paper, we propose a formal model of responsibilities in modal logic, using the same framework as Cohen and Levesque [6]. We start by introducing a partial order to represent the hierarchy of responsibility assignments (delegation structure). Then we offer definitions to characterize the mental state associated with having a responsibility. Rather than treating responsibilities as a unique, irreducible mental state (as intentions are treated in the BDI model, [10]), we model them as specific combinations of mutual beliefs and persistent goals. In the later sections, we briefly describe how the the model can be extended to incorporate joint responsibilities, and finally we discuss some practical consequences for the implementation of teamwork in multi-agent systems.

2 Partial Orders

Our formal model of responsibilities is set within the framework of Cohen and Levesque [6]. In particular, we assume that beliefs and goals are expressed using standard modal operators, **Bel** and **Goal**, with a possible worlds semantics. Further, we assume temporal information can be expressed in an Event Calculus whose operators are based on linear-time world models, which consist of infinite (indexed) sequences of events. We want to extend the formalism with axioms that capture the meaning of responsibilities, while making as few changes to the framework as possible.

We start by defining what responsibilities can be about. We view responsibilities as attitudes toward *actions* (that someone wants to make happen in the future). Primitive actions are simply events in world histories that are "caused" by some agent (have a non-empty Agt functions). Actions can also be complex, including multiple primitive actions combined together using sequential, conditional, non-deterministic, or iterative constructs (see the action expression language in Cohen and Levesque [6]). An action is said to have occurred in a world at a given point in time (index) if the subset of prior events includes the action(s) described, with other events possibly interspersed (including actions of other agents). Supposing that the desired action has not yet taken place in the real world, there are variety of possible courses of events that could happen in the future, with each sequence represented by a different possible world, and the action might eventually occur in only a subset of these worlds. Hence responsibilities can be thought of as a set of constraints on (desired) possible worlds, i.e. those whose sequences of events contain (or are consistent with) the specified set of actions after the present time. We will also call these objectives of responsibilities "tasks" and "goals" interchangably. Note that, for simplicity, in this paper, we do *not* incorporate the achievement of arbitrary logical conditions as goals. This would require reasoning about the potential for constructing unknown plans, which complicates the assessment of the mental state (this is the essence behind Grosz and Kraus's [11] distinction between $Intend.To$, which applies to actions, and $Intend.That$, which applies to goals, or conditions to be achieved). For us, in the remainder of this paper, goals always refer to sequences of concrete actions (or constraints thereon).

[1] Note that in this paper, we are explicitly concerned with "closed" multi-agent systems. In an open agent environment, where agents must interact with other agents designed differently, or even with humans themselves, notions of sincerity, reliability, etc. might need to be taken into consideration, along with other standard issues like agreement on communication protocols, ontologies, etc.

$$A \xrightarrow{>_\theta} B \xrightarrow{>_\theta} C$$

Figure 1: Illustration of responsibilities as a graph. This shows a delegation hierarchy among 3 agents, A, B, and C for a task θ. The interpretation of the arrows is *delegated to* (the reverse relation would be *responsible to*). In this case, A delegated the task to B, who delegated it to C. We represent this situation formally as $A >_\theta B >_\theta C$.

Fundamentally, responsibilities involve relationships *between agents*, relative to a certain action, in particular, the agent who gave (or delegated) the responsibility, and the agent who has (or received) the responsibility. To capture these relationships among agents, we introduce a partial order to represent this delegation hierarchy, which can be different for each responsibility. Suppose θ represents an action or sequence of actions that is the target of a responsibility, and that agent A initially had this responsibility, but delegated it to agent B. Then we would say that $B <_\theta A$, which reads as "B is responsible to A for performing θ." If B then sub-delegates responsibility for θ to C, we would extend the partial order to: $C <_\theta B <_\theta A$. We use a partial order for its mathematical properties: transitivity ($C <_\theta A$), anti-symmetry ($B \not<_\theta C$), and anti-reflexiveness[2] ($C \not<_\theta C$). The partial orders representing responsibilities can be viewed as directed graphs among agents (see Figure 1). (Branches in the graph would correspond to joint responsibilities, which are discussed in Section 4.1 below.) Importantly, note that, due to the properties of partial orders, the graphs must be acyclic (i.e. DAG's), so one cannot be in one's own chain-of-command. We will also freely use $>_\theta$, which has the obvious related meaning:

$$\forall x, y \; (x >_\theta y) \equiv (y <_\theta x)$$

A partial order can just be thought of as a binary relation among objects in a set that obeys the properties above. For each responsibility θ there is a unique set of pairs in $P \times P$, where P is the set of agents (people) in Cohen and Levesque's framework. To distinguish between singletons in the partial order that represent agents truly uninvolved in the responsibility from those who have simply not delegated it to or from another, we use $inv(\theta)$, which denotes the subset of agents ($inv(\theta) \subset P$) involved in each responsibility; at the very least, $\forall XY \; (X <_\theta Y) \rightarrow X \in inv(\theta) \wedge Y \in inv(\theta)$, but there may be some more. In order to capture the change in responsibilities over time (e.g. as a result of delegation), the partial order may be indexed by integers, corresponding to different points in time. For example, to represent the change in responsibilities in the example above, we could modify the expressions to say: $C \not<^t_\theta B <^t_\theta A$ and $C <^{t+1}_\theta B <^{t+1}_\theta A$. In this sense, the partial orders for each responsibility are actually subsets of 3-tuples from $P \times P \times Z$, where Z is the integers, which represent points in time.[3] These partial orders can be added (along with the inv function) to the Kripke structures in the Cohen and Levesque framework, augmenting the interpretations of sentences in the modal logic with models for reasoning about the hierarchical nature of responsibilities. Hence a model structure for us is a tuple:

$$\langle \Theta, P, E, Agt, T, B, G, \Phi, \{<_{\theta_i}\}, inv \rangle$$

where Θ is the universe of discourse, P is a set of agents, E is a set of events, Agt maps specific events in a world to a set of agents that cause them, T is a set of worlds (indexed sequences of events), B and G are accessibility relations capturing agents' beliefs and goals, respectively, Φ gives the interpretation of propositions in various worlds at various times, and $\{<_{\theta_i}\}$ is the set of partial orders, one for each responsibility θ_i (as an action expression) defined in the team model.

In a partial order for a responsibility, agents can play three distinct roles. First, an agent can be a "minimal element," such that there is no other agent to which it has delegated its responsibility. We say these agents have *direct responsibility*:

$$directResp(A, \theta) \equiv inv(A, \theta) \;\wedge\; (\neg \exists B \; A >_\theta B)$$

Second, an agent can be a "maximal element," such that there is no other agent to which it is responsible. We refer to these agents as having *ultimate responsibility*:

$$ultimateResp(A, \theta) \equiv inv(A, \theta) \;\wedge\; (\neg \exists B \; A <_\theta B)$$

Singleton agents could have both direct and ultimate responsibility simultaneously. Finally, it is helpful to define a notion of *indirect responsibility*, which refers to any non-minimal agents:

$$indirectResp(A, \theta) \equiv \exists B \; B <_\theta A$$

We define two special predicates to refer to a direct responsibility relationship between a pair of agents. If no other agents are "in between" in the graph, then we refer to one direction as *delegated to* and the other as *responsible to*:

$$delegated(A, B, \theta) \equiv (A >_\theta B) \wedge \nexists C \; (A >_\theta C >_\theta B)$$
$$\forall A, B, \theta \; respTo(A, B, \theta) \leftrightarrow delegated(B, A, \theta)$$

[2] Though one could argue that having a responsibility to oneself is meaningful, it is not needed for constructing most multi-agent systems.

[3] In order to represent cases where different agents hold divergent beliefs about responsibilities, we would have to be able to associate unique partial orders with specific worlds in their belief sets. Hence the partial orders for responsibilities could be extended to relations over $P \times P \times Z \times T$, where T represents the set of possible worlds. This would allow sentences with nested operators, such as $Bel(A, Resp(A, B, \theta))$.

3 Persistence

One of the primary motivations for defining explicit notions of direct and indirect responsibilities in the section above is that it helps in writing axioms to capture the meaning of responsibilities, which differs significantly between the two cases. In particular, having a responsibility does not necessarily obligate an agent to do anything; however, having a *direct* responsibility does require a special type of *persistence*. In this section, we consider only the most basic situation (binary) in which one agent (the "superior") has delegated a responsibility to another ("the subordinate"). (We discuss joint responsibilities in a later section.) Here we characterize what the responsibility means to the subordinate, in terms of what he must be persistent about.

Our view of the mental state of an agent with a direct responsibility is that they will be persistent in making sure that the action gets carried out. This is slightly different from the concept of persistence introduced in Cohen and Levesque [6]. In their single-agent view, an agent is said to be persistent if it has a **PGoal** to accomplish something θ, which is defined as an *individual* goal to do θ that is maintained over time until one of three conditions holds: 1) the goal is achieved, 2) the agent comes to believe the goal is unachievable (will never be true), or 3) the context changes, making the goal irrelevant. Our view of persistence for direct responsibilities augments this with one additional option: the agent could decide to delegate the responsibility. Provided that they can find another agent who is "capable" of performing the task, they can delegate the responsibility, which re-configures the partial order in the next time step by introducing a new minimal element in the graph beneath them. Then they would no longer have direct responsibility, and hence can give up being persistent on achieving the goal themselves.

We call this new type of persistence *r-persistence*, for responsibilities. We abbreviate r-persistent goals as RPG's. We use the following axiom to define this type of persistence:

DEFINITION: *R-persistence*

$$RPG(A,\theta) \equiv PGoal(A,\ Done(A,\theta) \vee [\exists B\ (canDelegate(A,B,\theta) \wedge Done(doDelegate(A,B,\theta)))])$$

Agents with direct responsibilities have RPG's: $directResp(A,\theta) \rightarrow RPG(A,\theta)$. This axiom expresses the intuitive idea that agents with direct responsibility must persist toward achieving a goal until they "do it, or delegate it" (or it becomes unachievable, which we handle below). This definition is almost equivalent to saying that the agent has the *intent* to do θ, in the sense of [6], but that they are also allowed to sub-delegate it as an option, and thus do not have to be quite so committed.[4] So we do *not* want to say: $directResp(A,\theta) \rightarrow Intend(A,\theta)$.

Of course, the critical terms in the axiom above that still need to be defined have to do with delegation. We must be able to reason about delegation as an action the agent can take in the environment, represented by the *doDelegate* event type. Regardless of how it is communicated to (or negotiated with) the new subordinate, the consequence of a delegation action is to modify the partial order, which is its primary effect. For example, if A delegates responsibility for θ to B at time t, then: $Happens(doDelegate(A,B,\theta),t) \rightarrow (A \not>^t_\theta B) \wedge (B \not>^t_\theta A) \wedge (A >^{t+1}_\theta B)$. This formula merely states some things that will be true after a delegation event; as additional consequences, A will become indirectly responsible, and B will become directly responsible, which can be derived from the properties of the new partial order $<^{t+1}_\theta$.

The predicate *canDelegate* is intended to represent the pre-conditions of the delegation action, which must be satisfied for it to work. The formal requirements for delegation are non-trivial. A task cannot be delegated to just anybody; there are several important preconditions. First, the superior agent might need to have authority over the subordinate (at least in some domains, such as those with a hierarchical team structure). Second, the new agent must also be "capable." Reasoning about capabilities of agents can be complex; see [12] for an example of a formal model of agent capabilities, which involves interrelating logics of time, action (typically assumed to be discrete and deterministic), and knowledge. Capabilities are often taken to refer to whether or not the post-conditions of an action will hold when a given agent tries to execute it in a situation in which the pre-conditions are true. However, more complex models of capability might also take into account the workload on the agent (i.e. being "too busy"), ability to meet task deadlines, degree to which quality criteria can be satisfied (levels of performance), etc. It is not within the scope of this paper to offer a comprehensive theory of capabilities of agents in teams. However, we feel it is important to raise these issues because simulation of sophisticated teamwork in multi-agent systems requires agents to be able to reason about each other's capabilities, especially when making the decision to delegate tasks. Therefore, we offer this *preliminary* definition of *canDelegate*:

DEFINITION: *canDelegate* [preliminary]

$$canDelegate(A,B,\theta,t) \equiv hasAuthority(A,B) \wedge isCapable(B,\theta,t)\ \wedge (A \not<^t_\theta B)\ \wedge\ (B \not<^t_\theta A)$$

Despite the intuitive sense that this preliminary definition makes, it is, in fact, not quite accurate. When one agent delegates a responsibility to another agent, the subordinate agent does not actually have to be capable of doing the action itself, since it could sub-delegate to yet another agent. This often happens with middle-level managers in a company - they accept responsibil-

[4] For simplicity, we have left out the requirement of the agent believing that it is about to do θ, which is significant to the meaning of intentions, but cumbersome here.

ities from their superiors, knowing only that one of the employees who works for them can do it. All that the agent who is delegating the responsibility needs to be sure of is that there is *some* way of accomplishing the task, whether the subordinate does it directly or delegates further to someone who can, and so on.

We can describe this formally using the idea of *extensions* of partial orders. An extension $<'$ of a partial order $<$ is another partial order which must be consistent with $<$ on all relationships specified in $<$, but may add some more objects or ordering constraints, i.e. the tuples in the binary relation for $<'$ must be a superset of those for $<$: $\forall x, y\ (x < y) \rightarrow (x <' y)$. Applying this to delegation, we only require that when one agent delegates a responsibility to another agent, the first agent is sure that there is some (possibly trivial) way to extend the delegation structure by adding new agents to the partial order such that there is at least one member that ends up with direct responsibility who is capable of doing the task.

However, we do not want to say $extension(<, <')$ directly, which is non-first-order. In order for agents to be able to reason about partial orders, they need to be able to refer to them as terms. Hence we introduce two new predicates: $porder(H, S)$ which declares an object H to be a partial order over the set S, and $below(X, Y, H)$, which intends the same relationship between objects X and Y (assumed to be in S) as $<$ would in the concept denoted by H. An axiom can easily be written that relates these two predicates based on the three usual properties of partial orders. Now we can define a first-order axiom for $extension()$: $\forall HJS\ porder(H, S) \wedge porder(J, S) \wedge [\forall XY \in S\ below(X, Y, H) \rightarrow below(X, Y, J)] \rightarrow extension(J, H)$. Using this notation, we give an improved definition for $canDelegate$ as follows:

DEFINITION: *canDelegate* [improved]

$$canDelegate(A, B, \theta, t) \equiv \exists H\ porder(H, \tau)$$
$$[\forall XY \in \tau\ (X <^t_\theta Y) \leftrightarrow below(X, Y, H)]$$
$$\wedge\ \exists J\ extension(J, H)$$
$$\wedge\ [\forall XY\ below(X, Y, J) \rightarrow hasAuthority(Y, X)] \wedge$$
$$\wedge\ [\exists C\ directResp(C, J) \wedge (below(C, B, J) \vee (C = B))$$
$$\wedge\ isCapable(C, \theta, t)]$$

where τ is the set of agents in the team. H is a term referring to a partial order isomorphic with the current delegation hierarchy $<^t_\theta$, and J is some possible extension of it (that must be legal according to constraints on authority). $directResp(C, J)$ has the obvious definition that parallels the meta-linguistic definition given in Section 2.

This more circumspect definition of when a responsibility can be delegated is important to the definition of persistence because it prevents agents from giving away a responsibility when they know there is no way it can be carried out (directly or indirectly) through the agent to whom they delegated it.

4 Mutual Belief

Defining persistence captures one aspect of responsibilities, in terms of helping the agent decide what to do, but it does not sufficiently get at the idea that responsibility involves some form of commitment to, and interaction with, a superior who delegated the responsibility. There is a sense in which the superior depends on the subordinate, perhaps because the responsibility is part of his plan (or a responsibility delegated to him). Intuitively, we feel that the subordinate must keep this in mind. Awareness of this dependency explains why agents realize they must inform their superiors when they realize they are going to fail. Hence capturing this aspect of the mental state is critical to showing how agents are motivated to generate this type of communication, which is essential for robust teamwork.

We approach this task in a way analogous to how Cohen and Levesque extend their framework to joint intentions in [13]. They view joint intentions as a special relationship among multiple agents in which each agent is individually committed to performing an action, they mutually believe this about each other, and furthermore they are all committed to maintaining consistent mutual beliefs about the status of the goal. We view responsibilities similarly, in that the superior and subordinate agents know about each other's desire for the goal and want to maintain mutual belief about its status, though it is more *asymmetric* because only one of them has to act (possibly). These mutual beliefs are what bind the agents together and provide a sense of dependency and cooperation.

We start by defining an *accountable r-persistent goal* (A-RPG), which augments an r-persistent goal (i.e. to do an action or delegate it) with an obligation to keep whoever gave the agent the responsibility informed of its status. We define an A-RPG for agent A to do θ for agent B (the superior) as:

DEFINITION: *Accountable r-persistent goal*

$$\textit{A-RPG}(A, B, \theta) \equiv RPG(A, \theta)$$
$$\wedge\ BEFORE(PGoal(A, MB(\{A, B\},$$
$$[\neg RPG(A, \theta) \wedge status(\theta)])\ ,\ \neg RPG(A, \theta))$$

Mutual belief (MB) is treated as an infinite conjunction of beliefs among a set of agents, such that each knows about the other's beliefs, including the recursive and self-referential cases (i.e. common knowledge [14]). This definition says that, if θ is not true now but still could be made true, then agent A will be r-persistent in carrying it out. However, if θ has been achieved or has become unachievable, A must adopt a persistent goal to establish mutual belief with the agent B, who gave him the responsibility, that he is relinquishing his RPG, and also the reason why ($status(\theta)$, shorthand for communicating its truth-value). The most important consequence of this constraint is that it effectively commits the subordinate agent to informing the superior when he is about to fail. This will cause the link $A <_\theta B$ to be deleted from the partial order (see Figure 2), giv-

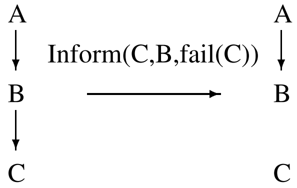

Figure 2: Reporting failure results in breaking a link in the delegation hierarchy, automatically changing that status of B from indirectly responsible back to directly responsible.

ing direct responsibility back to B and allowing him to try to find another way to achieve θ. This approach is analogous to the definition of *weak achievement goals* (WAG's) in Cohen and Levesque [13], where agents that share a joint intention have individual commitments to achieve a goal, but are also committed to keeping each other informed of the status of the goal.

Next, we offer a more complete definition of responsibilities based on mutual belief of goals and A-RPG's. It is not only important for the subordinate to have the accountable r-persistent goal, but for the superior to know this, and for the subordinate to know that the superior knows this, and so on.

DEFINITION: *Responsibility*

$$\forall A, B, \theta \ \underline{directResp(A, \theta)} \wedge delegated(B, A, \theta) \rightarrow$$
$$\textit{A-RPG}(A, B, \theta) \wedge$$
$$PGoal(A, MB(\{A, B\}, \textit{A-RPG}(A, B, \theta)))$$

$$\forall A, \theta \ \underline{indirectResp(A, \theta)} \rightarrow$$
$$Goal(A, Done(\theta)) \ \wedge \ [\forall B \ delegated(A, B, \theta) \rightarrow$$
$$BEFORE(PGoal(A, MB(\{A, B\},$$
$$\neg Goal(A, Done(\theta))), \neg Goal(A, Done(\theta)))]$$

The first axiom says that agents with direct responsibility will not only have an accountable r-persistent goal to achieve it, but will also make sure that its superior knows this and can thus expect all the consequences. The second axiom says that agents with indirect responsibilities are charged with keeping their subordinates apprised of any changes in the need for the goal. Note that the fact that the superior has the goal that the task eventually gets done ($Goal(A, Done(\theta))$), this does not mean it is committed to doing the task itself, which would be $Goal(A, Done(A, \theta))$. However, mutual belief in the existence of this generalized goal is essential for capturing the dependency in the responsibility relationship.

The importance of maintaining this system of mutual beliefs is, first, that the superior (or delegating) agent can now count on the subordinate informing it upon success (so it can drop its goal, and perhaps propagate this knowledge to its superior) or failure (in which case it becomes directly responsible again and must swing back into action). Similarly, based on the second axiom, agents trying to carry out responsibilities can rest assured that they will be promptly informed by their superiors whenever the need for them has dissipated.

4.1 Joint Responsibilities

Joint responsibilities can also be defined in a formal way using a modal logic of beliefs and goals. Like joint intentions [13], joint responsibilities are shared among two or more agents. Jennings [15; 2] discusses the utility of joint responsibilities in agent-based teams. Though not shown in this paper, axioms can be written that place additional constraints on agents that share a joint (direct) responsibility for a goal, requiring them to maintain persistent goals to keep each other informed about their individual abilities to accomplish the goal (and its achievement status), and only give up as a group and report failure to whomever delegated them the responsibility when they are sure that none of them can succeed.

5 Implementation in CAST

While the formal framework we have presented is abstract, it has some important practical implications for the design of multi-agent systems that incorporate assignment of responsibilities, such as our CAST architecture ("Collaborative Agent Architecture for Simulating Teamwork," [4]). In particular, the definitions in this paper offer a semantics to descriptions of responsibilities in our teamwork knowledge representation language, MALLET. In MALLET, it is possible to define a set of roles and responsibilities that associate agents with certain steps in a team plan. While delegation itself does not play a significant role in CAST, each agent can generally be thought of as being accountable to the team as a whole. Definitions such as A-RPG are crucial in helping agents decide what to do in various situations based on their responsibilities; communication is generated through the need to maintain mutual belief. In particular, such definitions facilitate the implementation of *procedural specifications*, such as:

1. when agents should try to accomplish the goal themselves,
2. when they should simply wait for others to do it,
3. that they must inform other agents when they are going to fail, and
4. how agents with indirect responsibility should react if subordinate or jointly responsible agents fail.

How agents in CAST interpret and act on responsibilities in cases of joint responsibility depends on attributes of the responsibility itself. In our previous work [4], we have identified three distinct types of responsibilities: *shared cooperative* responsibilities (AND type), *shared compatible* responsibilities (OR type), and *shared competitive* responsibilities (XOR type). With shared cooperative responsibilities (AND), all agents must act together. With shared compatible responsibilities (OR), any agent (one or more) can handle it. And with shared competitive responsibilities (XOR), at most one agent must carry it out, for example due to the potential for interference. These responsibility types bear some resemblance to AND- and OR-combinations of team operators in STEAM [16].

Each of these types of joint responsibilities would be handled differently among the agents sharing them, and these interactions often require communication. With AND-responsibilities, the agents must coordinate by synchronizing to perform the act simultaneously, waiting until everybody is ready. With XOR-responsibilities, the agents must communicate to *disambiguate* who is going to take over the responsibility individually. All of these communication requirements can be prescribed by specifying additional constraints on mutual beliefs that must be maintained, such as: 1) mutual belief that everybody is ready, prior to acting on an AND-responsibility, and 2) mutual belief that everybody agrees on a single agent that will perform an XOR-responsibility.

6 Conclusion

The framework we have presented in this paper is important for building multi-agent systems because it offers a semantics for responsibilities that is necessary for determining correct collaborative behavior in teams. It is often convenient to describe the structure of teamwork in a domain by identifying certain tasks or activities for which each agent (or set of agents) is responsible. However, the meaning of responsibilities must be clearly defined for the agents to know when they should act, when they should defer/wait, when they must communicate, etc.

In our framework, we have shown how the mental state associated with responsibilities can be modeled in terms of agents' beliefs and goals. We used the concept of persistent goals to capture the notion that responsibilities involve a desire to make something happen or become true, but, unlike an intention or commitment, has the caveat that allows for delegating that responsibility away and allowing someone else to do it on their behalf. Another essential feature of responsibilities is that they involve dependency, in the sense that (usually) a responsibility is given by a person/agent who needs it done. We expressed the nature of this inter-agent relationship through requirements for maintenance of mutual beliefs among the parties regarding the status of the goal, validity of the need, and possibility of the achievement. From the perspective of an agent who has a responsibility, this framework provides a basis for making intelligent decisions about how and when to act and interact within a team.

7 Acknowledgements

This work was supported in part by MURI grant No. F49620-00-1-0326 through the Air Force Office of Sponsored Research. The authors would like to thank the other members of the MURI research group at Texas A&M for feedback.

References

[1] R.W. Swezey and E. Salas, editors. *Teams: Their Training and Performance*. Norwood, NJ:Ablex, 1992.

[2] N.R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75:195–240, 1995.

[3] P. Stone and M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110:241–273, 1999.

[4] J. Yen, J. Yin, M. Miller D. Xu, R.A. Volz, and T.R. Ioerger. CAST: Collaborative agents for simulating teamwork. In *Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[5] L. Cholvy, F. Cuppens, and C. Saurel. Towards a logical formalization of responsibility. In *Proc. of the Sixth International Conference on Artificial Intelligence and Law*, pages 233–242, 1997.

[6] Philip Cohen and Hector Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[7] C. Castelfranchi and R. Falcone. From task delegation to role delegation. In M. Lenzerini, editor, *Lecture Notes in Artificial Intelligence, 1321*, pages 278–289. Springer-Verlag, 1997.

[8] T.J. Norman and C. Reed. Delegation and responsibility. In *Seventh International Workshop on Agent Theories, Architectures, and Languages*, 2000.

[9] C. Castelfranchi. Modeling social action for ai agents. *Artificial Intelligence*, 103:157–182, 1998.

[10] A. Rao and M. Georgeff. Modeling rational agents within a BDI-architecture. In *Proc., Second International Conf, on Principles of Knowledge Representation and Reasoning*, pages 473–484, 1991.

[11] B. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86:269–357, 1996.

[12] W. van der Hoek, B. van Linder, and J. Ch. Meyer. A logic of capabilities. In *Lecture Notes in Computer Science*, volume 813, pages 366–378. 1994.

[13] Philip R. Cohen and Hector J. Levesque. Teamwork. *Noûs*, 25(4):487–512, 1991.

[14] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning About Knowledge*. MIT Press: Cambridge, MA, 1995.

[15] N.R. Jennings. On being responsible. In *Proc., 3rd European Workshop on Modeling Autonomous Agents in Multi-Agent World*, pages 93–102, 1992.

[16] M. Tambe. Agent architectures for flexible, practical teamwork. In *Fourteenth National Conference on Artificial Intelligence*, 1997.