A First-Order Theory of Communication and Multi-Agent Plans

Ernest Davis[*]
Courant Institute
New York University
davise@cs.nyu.edu

Leora Morgenstern
IBM Watson Labs
leora@us.ibm.com

May 16, 2005

Abstract

This paper presents a theory expressed in first-order logic for describing and supporting inference about action, knowledge, planning, and communication, in an egalitarian multi-agent setting. The underlying ontology of the theory uses a situation-based temporal model and a possible-worlds model of knowledge. It supports plans and communications of a very general kind, both informative communications and requests. Communications may refer to states of the world or states of knowledge in the past, present, or future. We demonstrate that the theory is powerful enough to represent several interesting multi-agent planning problems and to justify their solutions. We have proven that the theory of knowledge, communication, and planning is consistent with a broad range of physical theories, despite the existence of a number of potential paradoxes.

Keywords: multi-agent planning, knowledge, communication.

1 Introduction

An autonomous agent who shares his environment with other autonomous, cooperative agents and who wishes to make plans that involve them should be able to reason about his own actions and their actions; his knowledge and their knowledge; his plans and their plans; and communications of information and requests between him and them. This paper presents a theory that integrates these various issues, and a representation of this theory in a first-order language.

Consider the following reasoning problems:

Problem 1: Ann and Barry are sitting together. Ann knows that Barry has her cell phone. Infer that Ann can get her cell phone back by asking Barry to give it to her.

Problem 2: Carol wishes to email David, but does not know his email address. However, Carol knows that Emily knows David's email address, and Emily is nearby. Infer that Carol can email David by executing the following plan: Ask Emily for David's email address. When she answers, email David.

[*] The research reported in this paper was supported in part by NSF grant IIS-0097537.

Problem 3: A warehouse is manned by a collection of robots, one on each floor. There is an elevator (actually, perhaps, more like a dumbwaiter) used for moving packages from one floor to another. The robots can communicate by radio. Each robot can carry out the following actions: call for the elevator; load a package on the elevator; unload a package from the elevator; communicate a fact to another robot; or broadcast a request to all the other robots.

In the starting situation, a particular robot, called the "hero", wants to get a particular package labelled b1. He knows that it is on some other floor, but he does not know where. Infer that the following plan will result in the hero having b1:

```
I will broadcast the following request:
     If you have package b1, then {
        call the elevator;
        when the elevator arrives, load b1;
        announce that b1 is on the elevator; }
When I know that b1 is on the elevator, I will call the elevator;
When the elevator arrives, I will unload b1 off the elevator.
```

Our objective in this paper is to develop a representation language in which problems like the above can be expressed and a theory in which inferences like the above can be justified.

The domain described in our theory is complex, and our theory combines a number of pre-existing theories and adds some original elements:

- The theory of time and action is an adaptation of McDermott's temporal logic [17].
- The theory of knowledge is a standard possible-worlds semantics.
- The theory of planning and of knowledge preconditions for plans follows Moore [18, 19] as extended by Davis [8].
- The theory of informative actions and their relation to knowledge follows our previous work in [9, 10].
- The theory of requests and commitments is original here.

The development of our theory is guided by the following considerations:

Expressivity. We wish our representation language to be as expressive as possible, in two senses: First, the language in which we (the external reasoners) represent knowledge *about* the agents and their plans should be as broad as possible. Second, the language in which the agents themselves *communicate* information and requests should be as broad as possible.

Our language supports both the representation and communication of information and requests involving physical states, knowledge states, and informative communications in the past, present, and future. It supports the representation of information about plans, requests, and commitments and it supports plans that incorporates requests. It does not, however, support the communication by agents of information about plans and requests; this is a gap that we hope to fill in later work.

Modularity: We have proven that our theory of knowledge, communication, and planning is compatible with almost any causal, physical theory over discrete time. This is made precise in the statement of Theorems 12 and 14 (section 9).

Minimality. In general, we wish to make as few unnecessary assumptions as possible; that is, to keep the theory as *weak* as possible. In particular, we have tried to avoid making closed-world assumptions, such as positing that the only events that occur are those required by the plan. In a few cases, we have violated this principle, using strong assumptions where weaker assumptions would suffice, for the sake of simplifying the analysis.

Monotonicity. We have used a monotonic logic, specifically first-order logic. The main motivation for this choice was just simplicity of analysis; assembling a complex theory is difficult enough without adding the well-known difficulties of plausible inference. The above principle of minimality also contributed to this decision. Non-monotonic theories work by filling gaps in a theory; it is difficult to get them to fill only the gaps that need to be filled without making many additional unnecessarily strong assumptions. Probabilistic theories in general require, first, the assignment of rather arbitrary numbers as base probabilities; second, the adoption of strong and often implausible independence assumptions to enable the determination of complex probabilities.

Of course, the use of a monotonic logic has its own drawbacks. First, there is a loss of realism. In almost all real cases, predictions of the behavior of other agents are plausible rather than deductive inferences. As we shall see, forcing these inferences into the Procrustean bed of monotonic inference necessarily involves doing some violence to reality and to common sense. Second, we cannot achieve our aims without making *some* closed-world assumptions, and such assumptions as we do make are, of course, made absolutely, whereas a non-monotonic theory makes these assumptions only provisionally.

Egalitarianism. The problem of multi-agent planning is much simpler and much less interesting, if we assume there is one boss, and that the other agents immediately carry out his requests. Indeed, such a theory is hardly a multi-agent theory at all; the only agent with true choice is the boss, and the rest are just automata. In the theory developed here, any agent can make a request of any other agent, and the latter will make a sincere effort to carry it out.

This principle is, of course, on a very different level from the first three; it is a particular type of multi-agent theory, rather than a general methodological choice. It has many ramifications for our theory development, some favorable, others more problematic. On the one hand, it increases the importance of the minimality assumption. Certainly, in an egalitarian theory, we cannot assume that the other agents are immediately available to carry out a request; they may be busy with their own affairs or with someone else's request, and the state of the world may change while they are doing this. The planner himself may have to be busy at times with someone else's requests. Both the specific plan and the axiomatic theory of plans must be formulated in a way that accommodates this.

On the other hand, the requirement that the social theory be egalitarian jibes rather awkwardly with the need to prove that these plans will work. It is hard to see, in an egalitarian theory, why the hero of problem 3 should be *certain* of getting the package; suppose someone else requests the same package simultaneously? Indeed, to achieve monotonicity, we will be required, both to impose a rather restrictive protocol on how agents decide to service requests, and to posit that the hero owns package b1 and therefore gets to say what is done with it. Egalitarian societies do not have fewer rules than despotic ones, just fairer rules. Ultimately, though, the fundamental mode of interaction in egalitarian societies is not through making requests and acceding to requests; it is through negotiation and bargaining [14]. The analysis in this paper, however, does not attempt to deal with the complex issues involved in negotation. Rather it focusses on a rather special case, though an important one, in which a cooperative society has been established so that an individual request does not require a specific *quid pro quo.*

A number of further strikes against our theory should also be noted. First, the theory of requests is highly idealized. The actual rules of social and personal interaction that determine, in human intercourse, whether person A can reasonably request B to do P and whether he can be confident that B will in fact do as requested are immensely complex, and depend on the social relation of A and B; their personal relation if any; the ease with which B can do P; the previous interactions of A and B (making one request is different from making a hundred requests); and so on. We have not modeled any of this. Rather, we have invented a highly idealized protocol that is devised to enable the target inferences to be justified while keeping the theory consistent.

Second, we only allow agents to request other agents to *carry out specified plans,* not to *achieve specified goals.* Our theory of plans is very general, so this is not quite as burdensome as it may seem; nonetheless, it certainly does lead to a rather micromanaging style of agent interaction. We hope to remove this restriction in future work, but major technical issues need to be resolved to do so.

Third, in order to avoid inconsistencies, our language excludes a number of types of communications that intuitively would seem to fall into the same category as the above examples. For example, in problem 1, after Ann has asked Barry to return her cell phone, she cannot then tell Charley that she *will* be getting her cell phone back. All that she is allowed to say is that she has asked Barry to give it back to her, and that if Barry does so, then she will have it. Not that our theory *forbids* her to say that she will have the cell phone; rather, there is no syntactically well-formed way, in our representation language, that she can even express that she will have her phone. We, using the language, can express it, but she cannot. This will all be explained in detail in section 8.

Despite these major limitations, we believe that this work is important for the following reasons:

- The language of requests is, in important respects, the most general that has been developed among theories that have a well-defined semantics and that have been proven consistent.

- It is reasonable to expect that it will be found possible to extend our language of requesting plans to a more expressive language of requesting goals.

- The protocol we define, though idealized, is a first approximation at a theory of agent interaction, and for some problems will suffice. As more realistic theories of general human interactions and of constrained interactions in specific contexts (e.g. client-server models for Web interactions) are developed, the same language of requests can still probably be used with little change. Our theory of requests here can probably serve as the basis or template for the new theory, though this would by no means be a plug-and-play operation.

Section 2 of this paper describes pre-formally how we develop an egalitarian multi-agent protocol, in such a way that plans can be guaranteed to succeed. Sections 3 through 6 develop the the representation and axiomatization of, respectively, time, events and actions; knowledge; speech acts; and multi-agent plans, incorporating our protocol. Section 7 shows how problems 1, 2, and 3 above can be expressed in our representation language. Section 8 deals with formulating comprehension axioms for fluents and plans. This requires some care, so that, on the one hand, you allow for very general communication of information and requests, and, on the other hand, you avoid potential inconsistencies analogous to Russell's paradox. Section 9 presents a theorem that our theory of knowledge, communication, and planning is consistent with a wide range of physical theories. Section 10 reviews the related literature. Section 11 presents our conclusions. Appendix A discusses the differences between the theory of time, knowledge, and informative acts presented here and that presented in [9, 10], and sketches how the proof of consistency for the theory given there can be extended to a proof of consistency for the theory presented here. Appendix B gives a detailed proof of the correctness of the plan in problem 3. Due to length limitations, Appendices A and B are published on the Web (http://cs.nyu.edu/faculty/davise/commplan/commplan-appa.pdf and commplan-appb.pdf) but not included here.

2 Protocol

In this section, we address the following problem: We want to set up our theory so that, under suitable circumstances, if agent AR requests agent AC to carry out plan P, then AR can be sure that AC will make an earnest attempt to carry out P if possible. We want to be able to do this with the fewest presuppositions about what else is going on. In particular, we do not want to rule out the possibility that other agents may make their own requests of AC or that AC may have his own plans to work on. We do not want to demand that AR should know what AC is doing, or what requests have been made of him, or what he is working on. (We do require that AR should know that AC has no outstanding requests from AR himself that he is still working on.) We want the theory to be egalitarian over the agents; the problem of conflicts is trivial if all requests are issued by an autocratic boss to underlings.

Therefore, we need a *protocol* for making sure that the plans, either personal or requested, being attempted by an agent do not conflict. We put the entire burden of coordinating these plans on the acting agent AC; there are no constraints placed on what one agent may request of another, except that AR may not issue two requests to AC simultaneously. (He may, however, issue a single request of the form "Please accomplish both $P1$ and $P2$.") However, we give the acting agent fairly large latitude in deciding to *abandon* a plan if he is unable to proceed.

Our protocol is based around two ideas: Agent AC *reserves* a block of time for AR, and AR *governs* certain actions of AC. First, we posit that an agent AC reserves blocks of time for each of his fellow agents. A reserved block of time has a minimum duration "min_reserved_block", and there is a maximum delay "max_delay" between successive blocks reserved for the same agent. During a block that AC has reserved for AR, AC is committed to working on whatever plan has been requested by AR, if any. Therefore, AR can be sure that, within a time no more than delay_time, AC will give his full attention to AR's plan for a period of time not less than min_reserved_block. If AC cannot finish the plan by that time, then he will continue returning to it at future time blocks that he reserves for AR until he finishes it.

However, AR may also need to exert some more limited control to block seriously counter-productive actions during time periods not reserved for him. What he is allowed to do is to prohibit actions that he "governs". In problem 3, for example, we will suppose that the hero "owns" the package that he is trying to get, and that he therefore "governs" the actions of loading and unloading that package. (If he does not own it, then why should we be sure that he ends up with it in a case where another agent requests it at the same time?) Governing an action is an exclusive relation; only one agent may govern a particular action.

At best, however, there is necessarily a brief delay between the time when the hero decides to make the request for the package, and the time when that request is accepted by the other agents — namely, the time that it takes to carry out the action of broadcasting the request — and during this delay, the hero has, of course, no control whatever over what happens. There may also be a long delay between the time that the request is broadcast and the time when the agent who has the package can attend to it (because he has reserved the time for the hero), and another long delay between the time when the second agent informs the hero that the package has been loaded onto the elevator and the time when the hero can take his attention away from jobs that he is doing for other people and actually call for the elevator. During both of these periods, the only control that the hero has over the actions of the other agents, or even his own actions, is to prohibit them from carrying out actions that he governs. In particular, he can instruct the other agents not to unload the package, since he owns it. Therefore plans in this microworld end up with something of the flavor of Schoppers' "universal plans" [30], with built-in contingency plans for whatever is the state of the world when the hero or his friends can get around to working on the plan.

Plans are specified in terms of the "next_step" and "succeeds" predicates, as in [8]. The relation "next_step($E, P, S1, S2$)" means that, in situation $S2$, E is an acceptable next step of an execution of plan P that started in situation $S1$. The relation "succeeds($P, S1, SZ$)" means that an execution of plan P that started in situation $S1$ succeeds in situation SZ. You can define the "success" of P as any property as any property you want of the history between $S1$ and SZ; e.g. "Block A is on block B in SZ," "Agent A runs around the block three times between $S1$ and SZ," "SZ is ten years after $S1$, and agent A does not go bankrupt between $S1$ and SZ," and so on. This kind of specification supports an extremely broad class of plans, subsuming many standard plan representations [8].

The next_step relation is also used, in a negative way, to represent the actions that AR prohibits to AC during times that AC does not reserve for AR. A plan P must be defined to explicitly take into account the actions that AC takes in a situation S reserved for other agents. In general, P will permit AC to take *any* action other than a small number of actions that AR governs and that he wishes to prohibit as counter-productuve for the progress of P. All permitted actions are considered "possible next steps" of P in S. Thus, an action E executed by AC in a situation S is generally a "possible next step" of all the active plans. The plan requested by the agent for whom AC reserves S specifies E as a "possible next step" in the positive sense: it is one of a few choices of actions that will advance the plan. The plans requested by all the other agents also charaterize E as a "possible next step"; indeed, each such plan defines its own "possible next steps" as any action except for the few it wishes to prohibit. The theory does not make any formal distinction, however, between the two cases; in either case, E is simply characterized as a "possible next step".

Thus, in problem 1, the plan P that Ann asks Barry to carry out has the following form: In any situation S, if you have reserved time(S) for me, and you have the phone, then the next step of P in S is to give me the phone. If you have not reserved time(S) for me then the next step of P in S may be any action except to give the phone to someone else.

Putting all this together, we arrive at the following rule for an agent carrying out a plan: (Tables 17 and 18 contain the formal statement of this rule.)

Let AC be a cooperative agent who has accepted plan P from source AR in situation $S1$. Suppose that AC has continued his work on P up to situation $S2$. Then the following possibilities obtain in $S2$:

1. In $S2$, AC is engaged in some action. Then he continues that action.
2. In $S2$, AC knows that plan P has succeeded. Then he need do nothing more with P.
3. In $S2$, AC is idle, and AC does not know that P has succeeded (usually because P has not yet succeeded.)

    3.1. AC has reserved the time of situation $S2$ for AR.

    An action E is said to be an executable continuation of P if AC knows that E is a next step of P, is possible, and is not governed by $A3 \neq AR$.

    3.1.1. If there exists an executable continuation of P, then AC executes some executable continuation of P.

    3.1.2. If there does not exist an executable continuation of P, then AC may abandon P. (This is abandonment condition 1.)

    3.2 AC has not reserved the time of situation $S2$ for AR. Then P may prohibit AC from carrying out E only if AR governs E.

    3.2.1. AC knows that the only actions that are prohibited by P are governed by AR. Then AC may carry out any action that he knows is a next step of P (i.e. not prohibited by P.)

    3.2.2. AC does not know that the only actions that are prohibited by P are governed by AR. (I.e. as far as AC knows, it is possible that there is an action E that is not governed by AR and that is not a possible next step of P.) Then AC may abandon P. (This is abandonment condition 2.)

Our protocol is quite inflexible and inefficient; for example, every agent AC reserves a regular time-slot for every other agent AR even if AR has not made any request of AC. We have done this in order to simplify as far as possible the logical statement of the protocol. It would be easy to define informally a much more flexible and more efficient protocol and we believe that most such protocols could be integrated into the axiomatization of the rest of our theory in an analogous way without great difficulty. However, we have not done so, because our focus in this paper is not on protocol development and analysis but on formalizing the theory of planning and communication. We have therefore tried to make our protocol as simple as possible while still giving a consistent theory that can support our target inferences. What would be very desirable, but we have not yet attained, would be to separate the axiomatization of the protocol from the axiomatization of planning, so that the identical axioms of planning could be used with any (reasonable) protocol.

The particular abandonment conditions have been adopted in order to simplify the proof that the plan is executable. Abandonment condition 1 is natural enough; if the agent AC does not know any way to continue the plan, he abandons it. Abandonment condition 2 seems over restrictive; why is it necessary that AC knows that AR governs *all* the actions prohibited by plan P? Why does it not suffice that AC knows of *some* action he can do that he can do that is not prohibited by P? The reason is AC needs to keep in mind the plans of all the agents he is interacting with. In particular, if at a given time AC is executing plan $P1$ of agent $AR1$, and E is a next step of $P1$, we need to make sure that none of the plans that are currently on the back-burner prohibit E. Moreover, we have to ensure that, when AR requests AC to do P, AR can be sure that his prohibitions do not give rise to this situation, regardless of *whatever* any other plans other agents have requested or will request from AC (since AR cannot know what these are). There may be less restrictive ways of accomplishing this, but this seems like the simplest.

3 Time

3.1 Formal notation

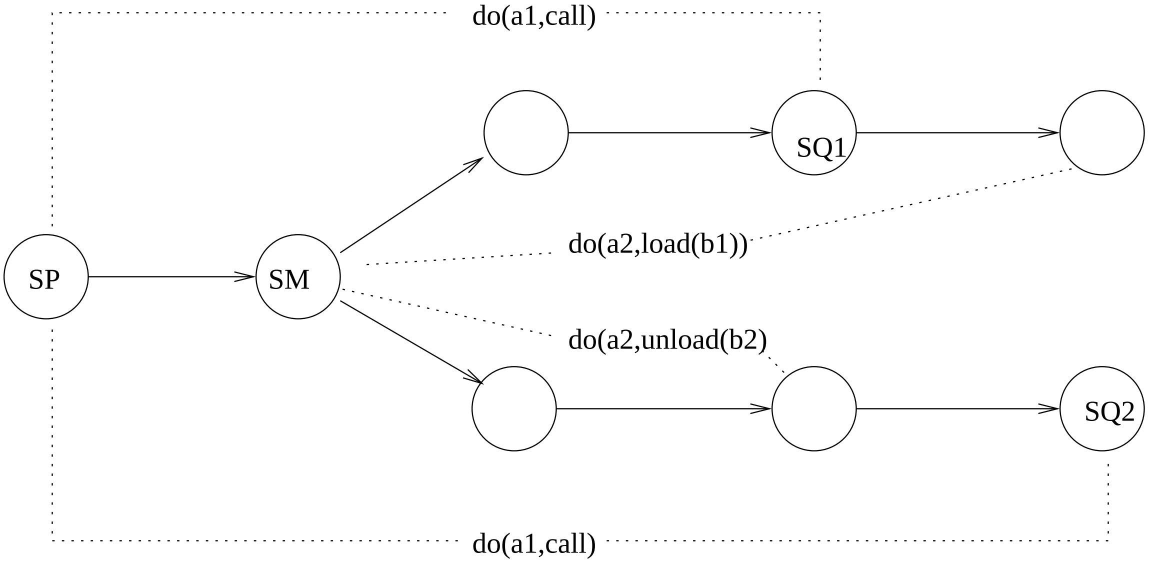
First, a few general rules about our formal notation. We use a sorted first-order logic. The sort of a variable is indicated by its first letter. In almost all cases, any two different sorts are disjoint. The one major exception is that "actions" is a sub-sort of "events"; a couple of further minor exceptions are described in section 8. The scope of a quantifier is to the end of the formula or of an outside close bracket. The definitional equivalence sign "$\equiv$" has lower precedence than implication "$\Rightarrow$" or two-way implication "$\Leftrightarrow$", which have lower precedence than conjunction "$\wedge$", disjunction "$\vee$" or negation "$\neg$"; otherwise, precedence is indicated with square brackets. Free variables are assumed to be universally quantified with the whole sentence as scope. Variables are in upper case italics; other symbols are lower case. All functions are total, over the sort of the arguments; when we have to deal with what intuitively ought to be a partial function, we use a relation and add an axiom stating that the relation is single-valued in its last argument.

3.2 The time structure

The temporal theory is basically that of McDermott [17] except that, for simplicity, it uses a discrete rather than a continuous time line. We describe it here in terms of its relation to the better-known situation calculus [16, 26]. The original version of the situation calculus is designed for reasoning about a single agent who can choose between actions. Each action E is taken to be atomic, starting in a situation S and finishing in a successor situation result(E, S). No situations occur between S and result(E, S). The time structure reflects the fact that the agent has a choice between actions by having it branch at S; each possible action E gives rise to a different resulting situation S.

Our world is rather different. We have multiple agents. By the principle of making minimal assumptions, we wish to posit that they act asynchronously; by the principle of egalitarianism, we wish to posit that every agent can choose between actions. We do assume, for simplicity, that there is a level of primitive, atomic (non-decomposable) actions, and, more restrictively, that each agent is individually serial — that is, each agent can execute only one primitive action at a time.

However, since agents are acting asynchronously, that means that if agent a2 finishes what he was doing and must choose his next action while a1 is in the middle of action e1, then the time structure must split in the middle of a1's execution of e1. (Figure 1). Therefore, we cannot use a "result" function; if a1 starts to execute action e1 in situation s0 the execution may end in any number of different situations, depending on the actions begun by other agents while a1 is busy. Instead, we use a predicate "occurs$(E, S1, S2)$"; action E occurs between situations $S1$ and $S2$. The constraint that an agent executes only one action at a time is now expressed in axioms A.1 and A.2. This form of representation was introduced in [17] and is expanded in [6].



The event "do(a1,call)" occurs, both over the interval [SP,SQ1] and over the interval [SP,SQ2].
The time line splits at situation SM depending on whether agent a2 decides to unload package b1 or to load package b2.

Figure 1: A branch in the middle of an event

Reiter [26] uses a different solution to the same problem. A single action at the intuitive level, such as walking from home to the office, is broken formally into three parts: The action of starting the walk, the fluent of walking as a state, and the action of ending the walk. For most domains, the two approaches are probably isomorphic, but we think the one here is clearer both ontologically and notationally.[1] Reiter's approach has the advantage — a critical one, in terms of his research program — of being a direct extension to the standard situation calculus representation, and therefore more easily integrated with the large body of theory that has been developed for that representation.

Following Reiter [26] we posit that no two agents attain a choice point simultaneously; we call this the *axiom of anti-synchrony.* There are two advantages of this. The lesser advantage is that it simplifies the physical theory; we do not have to characterize under what conditions two agents can start two actions simultaneously. The more important advantage is that it simplifies the theory of plan feasibility. If every choice point involves only one agent, then we can easily locate the choices of a given agent in the time structure. If two agents $A1$ and $A2$ reach a choice point simultaneously, then it becomes much more difficult to distinguish the choices that are under the control of $A1$ from the choices that are under the control of $A2$.

Other aspects of our theory are reasonably straightforward; they derive partly from our principle of minimality and partly from convenience. For the purposes of our theory of planning, it is useful to posit that an agent is always busy with some action. In our theory of planning, a plan specifies what actions might be performed in a given circumstance; it is simply awkward to allow the alternative "no action". The theory of action provides an action "wait" to fill up the gap when an agent does not want to be more energetically active.

One final difficulty: We wish to assume that there is an initial starting time 0t; otherwise, the consistency proof does not work. (Time 0t does *not* have to be the time when the plan is begun.) But by definition all agents must start an action in the starting situation, violating the axiom of anti-synchrony. We get around this problem by making the starting situation a specific exception to the axiom, and requiring that all agents begin with the action "wait" in the starting situation, so that no problems of interaction arise then.

3.3 Metric time

Our protocol and our description of the elevator world are phrased in terms of metric time. Our theory of metric time involves two sorts: *durations*, such as a minute, and *clock-times,* such as "12:35 P.M. May 12, 2001." Durations are taken to be the integers. (Axioms T.4 – T.9, T.15 of table 3). A duration can be added to a clock time to give another clock time. For any clock-time $T0$, the function $\lambda(D)T0 + D$ is an isomorphism from the space of duration to the space of clock-times (Axioms T.11 – T.14). The function "time(S)" maps a situation S to a non-negative clock-time. If $\mathcal{L}$ is a time line in the branching structure of situations – that is, $\mathcal{L}$ is a maximal ordered set of situations – then the function time($\cdot$) is an isomorphism from $\mathcal{L}$ to the space of non-negative clock-times (Axioms T.16-T.18).

[1] In domains with continuous branching, it is possible to use the "occurs" representation [7] whereas it is hard to see how the situation calculus could be extended to deal with these.

Sorts in the temporal theory: Situation (S) (= possible world), clock time (T), duration (D). For mathematical convenience, we allow negative durations and clock-times.

Non-logical symbols:

$S1 < S2$, $S1 > S2$, $S1 \leq S2$, $S1 \geq S2$ — Predicates. The order relations on situations.
$T1 < T2$ etc. Predicates. The order relations on times.
$D1 < D2$ etc. Predicates. The order relations on durations.
$D1 + D2$. Function. Addition of durations.
$T1 + D2$. Function. Time plus duration → time.
0, 1. Constants. Duration.
0t. Constant. Time
time(S). Function situation → time.
ordered(SA, SB) — Predicate. Situations SA and SB are ordered.

Table 1: Durations, Times, Situations

Definitions:

TD.1 $X1 > X2 \equiv X2 < X1$.
$X1 \leq X2 \equiv X1 < X2 \vee X1 = X2$.
$X1 \geq X2 \equiv X1 > X2 \vee X1 = X2$.
(Definition of the other order relations.)

TD.2 ordered(SA, SB) $\equiv SA < SB \vee SA = SB \vee SA > SB$.

Table 2: Basic Temporal Elements: Definitions

Axioms:

T.1 $S1 < S2 \Rightarrow \neg(S2 < S1)$.
(Asymmetry)

T.2 $[S1 < S2 \wedge S2 < S3] \Rightarrow S1 < S3$.
(Transitivity)

T.3 $SA < S \wedge SB < S \Rightarrow \text{ordered}(SA, SB)$.
(Forward branching)

Axioms of times and durations

T.4 $D1 + D2 = D2 + D1$

T.5 $D1 + (D2 + D3) = (D1 + D2) + D3$

T.6 $D + 0 = D$

T.7 $\forall_{D1} \exists_{D2}\, 0 = D1 + D2$.

T.8 $D1 < D2 \Rightarrow D1 + D < D2 + D$.

T.9 $[D1 < D2 \ \wedge \ D2 < D3] \Rightarrow D1 < D3$.

T.10 $D1 < D2 \dot{\vee} D2 < D1 \dot{\vee} D1 = D2$.

T.11 $T + 0 = T$.

T.12 $\forall_{T1,T2} \exists^1_D\, T1 = T2 + D$.

T.13 $(T + D1) + D2 = T + (D1 + D2)$.

T.14 $\forall_{T,D}\ T < T + D \Leftrightarrow 0 < D$

T.15 Durations are integers. Axiom schema: Let $\phi(D)$ be a formula with an open duration variable D. Then the closure of
$[\phi(0) \ \wedge \ \forall_D[(D \geq 0) \ \wedge \ \phi(D)] \Rightarrow \phi(D+1)] \Rightarrow [\forall_D\, D \geq 0 \Rightarrow \phi(D)]$
is an axiom.

Axioms of situations and times.

T.16 $S1 < S2 \Rightarrow \text{time}(S1) < \text{time}(S2)$.

T.17 $\forall_S\ \text{time}(S) \geq 0\text{t}$.

T.18 $\forall_{S1,T2} T2 \geq 0\text{t} \Rightarrow \exists_{S2}\ \text{ordered}(S1, S2) \wedge T2 = \text{time}(S2)$.

Table 3: Basic Temporal Elements: Axioms

Sort: Fluents (Q).

Non-logical symbols:
holds(S, Q) — Predicate. Fluent Q holds in situation S.
throughout($S1, S2, Q$) — Predicate. Fluent Q holds throughout interval $[S1, S2]$.
or_f($Q1, Q2$) — Function fluent,fluent → fluent. Either $Q1$ or $Q2$ holds.
and_f($Q1, Q2$) — Function fluent,fluent → fluent. Both $Q1$ and $Q2$ holds.
neg_f(Q) — Function fluent → fluent. Q does not hold.
true_f — Constant. Fluent that is always true.
within(S, D, Q) — Predicate: Fluent Q will hold within time D of S.

Definitions:

FD.1 $\text{throughout}(S1, S2, Q) \equiv \forall_S \; S1 \leq S \leq S2 \Rightarrow \text{holds}(S, Q)$.

FD.2 $\text{holds}(S,\text{or\_f}(Q1, Q2)) \equiv \text{holds}(S, Q1) \vee \text{holds}(S, Q2)$

FD.3 $\text{holds}(S,\text{and\_f}(Q1, Q2)) \equiv \text{holds}(S, Q1) \wedge \text{holds}(S, Q2)$

FD.4 $\text{holds}(S,\text{neg\_f}(Q)) \equiv \neg\text{holds}(S, Q)$

FD.5 $\forall_S \; \text{holds}(S,\text{true\_f})$.

FD.6 $\text{within}(S, D, Q) \equiv$
$\forall_{S3} \; [S3 > S \wedge \text{time}(S3) = \text{time}(S){+}D] \Rightarrow \exists_{S2} \; S < S2 \leq S3 \wedge \text{holds}(S2, Q)$.

Axiom:
Comprehension axiom on fluents: See section 8.

Table 4: Fluents

3.4 Fluents

A *Boolean fluent*, such as “has(hero,b1)” is either true or false in any given situation. As usual we write “holds(S, Q)” to mean that fluent Q holds in situation S. (Table 4)

There is a comprehension axiom which states, roughly, that any property of situations can be considered a fluent. Unfortunately, it can’t state exactly that, because that ends up running into Russell’s paradox. The details of the problem and a correct statement of the comprehension axiom will be discussed in section 8.

3.5 Unbounded Intervals

If the time structure is discrete, there is not much to be gained by including bounded intervals as ontological entities; any property of the bounded interval $[S1, S2]$ can be stated as a property of the two situations $S1, S2$. (In the English exposition, it will sometimes be smoother to talk about the interval $[S1, S2]$.) However, this does not apply to intervals that are unbounded above, which we will call “u-intervals”; these cannot be specified in terms of any finite number of situations. We will need unbounded intervals because plans in a general planning language can go into an infinite loop, and this behavior can only be adequately described in terms of unbounded time intervals. The axioms governing u-intervals are given

Sort: U-intervals.

Non-logical symbols:
elt(S, I) – Situation S lies in u-interval I.

Axioms:

I.1 $\text{elt}(S1, I) \wedge \text{elt}(S2, I) \Rightarrow \text{ordered}(S1, S2)$.
(Totally ordered)

I.2 $\text{elt}(S1, I) \wedge \text{elt}(S2, I) \wedge S1 < S < S2 \Rightarrow \text{elt}(S, I)$.
(No gaps)

I.3 $\forall_{I1,I2}\ [\forall_S\ \text{elt}(S, I1) \Leftrightarrow \text{elt}(S, I2)] \Rightarrow I1 = I2$.
(Extensionality)

I.4 $\forall_{I,S1}\ \exists_{S2}\ \text{elt}(S2, I) \wedge \neg(S2 < S1)$.
(Unbounded above)

I.5 (Schema) Let $\phi(S)$ be a formula with an open situational variable S. Then

$$[\phi(S1)\ \wedge\ [\forall_S\ \phi(S) \Rightarrow \exists_{S2}\ S < S2\ \wedge\ \phi(S2)]] \Rightarrow$$

$$\exists_I\ S1 = \text{start}(I) \wedge [\forall_S\ \text{elt}(S, I) \Rightarrow \exists_{S2}\ S < S2\ \wedge\ \phi(S2)\ \wedge\ \text{elt}(S2, I)]$$

(Comprehension axiom for u-intervals unbounded above. If ϕ holds in $S1$ and every occurrence of ϕ is followed by a later occurrence of ϕ, then there is an u-interval (unbounded above) starting in $S1$ in which ϕ occurs infinitely often.)

Table 5: U-Intervals: Axioms

in table 5. This includes a comprehension axiom I.5 that states, intuitively, that all the u-intervals that ought to exist actually do exist.

3.6 Actions, Events, Actionals

An action is an entity like "do(hero,unload(b1))". An occurrence of an action takes place from a starting situation to a later ending situation. An action has a single actor. As discussed above, actions are atomic and mutually exclusive in the sense that an agent carries out exactly one action at a time.

The category of events (event types) is a supercategory of actions. (This is the one major case in our theory where two sorts are not disjoint.) An occurrence of an event takes place from a starting situation to an ending situation. Events need not be associated with a single actor, need not be atomic, and need not be mutually exclusive. In this paper, we use events chiefly to encode abstract or partial characterizations of speech acts, as described in section 5.

The second argument "unload(b1)" of an action term like "do(hero,unload(b1))" is, so to speak, an action unanchored from the particular agent. For want of a better name, we call this kind of term an "actional".

Sort: Event

Non-logical symbols:

occurs$(E, S1, S2)$ — Predicate. Event E occurs from $S1$ to $S2$.

leads_towards$(E, S1, S2)$ — Event E in situation $S1$ leads towards $S2$. (See discussion in the text.)

feasible(E, S) — Predicate. Event E can occur beginning in situation S.

Definitions:

EVD.1 leads_towards$(E, S1, SZ) \equiv \exists_{S2}$ occurs$(E, S1, S2) \wedge$ ordered$(S2, SZ) \wedge S1 < SZ$.

EVD.2 feasible$(E, S) \equiv \exists_{S2}$ occurs$(E, S, S2)$.

Axiom:

EV.1 occurs$(E, S1, S2) \Rightarrow S1 < S2$.
(Directionality. Note there are no instantaneous events.)

Table 6: Events

One predicate that we will find particularly useful is "leads_towards$(E, S1, S2)$", defined in definition EVD.1, table 6. What this means is that there is a time line containing situations $S1$ and $S2$ and containing an occurrence of E that starts in $S1$. $S2$ may be either during the occurrence of E or after the completion of E. For example, in figure 1, the action "do(a2,load(b1))" leads from SM towards SQ1; the action "do(a2,unload(b2))" leads from SM towards SQ2.

Tables 6, 7, and 8 give the axioms for events, actions, and actionals.

Primitive actions are a sub-sort of events. A primitive action is associated with a unique actor. When one action finishes, another begins. An agent A does one action at a time; that is, for any situation S, either A is in the middle of some action in S, or S marks the end of one action and the beginning of the next. Of course, the action may be "wait".

Sorts: Actions (E), actionals (Z), agents (A).

Non-logical symbols:
do(A, Z) — Function: agent, actional → action.
action(E, A) – E is a primitive action of agent A.
choice(A, S) — In S, A has just finished one action and must choose his next action.
engaged(E, A, S) — Predicate. Agent A is engaged in action E in S.
wait — Constant. Actional of waiting.

Definitions:

AD.1 $\text{engaged}(E, A, S) \equiv \text{action}(E, A) \wedge \exists_{S1,S2}\ S1 < S < S2 \wedge \text{occurs}(E, S1, S2)$.

AD.2 $\text{choice}(A, S) \equiv \text{time}(S) \neq 0\text{t} \wedge \exists_{Z,S1}\ \text{occurs}(\text{do}(A, Z), S, S1)$.

AD.3 $\text{action}(E, A) \equiv \exists_{Z}\ E = \text{do}(A, Z)$.

Table 7: Primitive actions: Definitions

Axioms:

A.1 $\forall_{A,S1,S2}\ \text{choice}(A, S1) \wedge S1 < S2 \Rightarrow \exists^{1}_{E}\ \text{action}(E, A) \wedge \text{leads\_toward}(E, S1, S2)$.
(In every branch of the time structure out from a choice point $S1$, agent A executes exactly one action E. That action can be waiting, of course.)

A.2 $\forall_{A,S}\ \text{choice}(A, S) \Leftrightarrow \exists_{Z,S1}\ \text{occurs}(\text{do}(A, Z), S1, S)$.
(Choice points for agent A occur when A has completed an action do(A, Z).)

A.3 $\text{do}(A1, Z1)=\text{do}(A2, Z2) \Rightarrow A1 = A2 \wedge Z1 = Z2$.
(Unique names axiom for actions.)

A.4 $[\text{choice}(A1, S) \wedge \text{choice}(A2, S)] \Rightarrow A1 = A2$.
(Axiom of anti-synchrony: No two agents are at a choice point at the same time.)

A.5 $\forall_{A,S}\ \text{choice}(A, S) \Rightarrow \text{feasible}(\text{do}(A,\text{wait}), S)$.
(At a choice point, an agent can always choose to wait.)

A.6 $\text{time}(S0)=0\text{t} \wedge \text{occurs}(\text{do}(A, Z), S0, S1) \Rightarrow Z=\text{wait}$.
(Initially, all the agents begin by waiting.)

A.7 Unique names axioms over actionals.

Table 8: Primitive actions: Axioms

4 Knowledge

As first proposed by Moore [18, 19] and widely used since, knowledge is represented by identifying temporal situations with epistemic possible worlds and positing a relation of knowledge accessibility between situations. The relation k_acc(A, S, SA) means that situation SA is accessible from S relative to agent A's knowledge in S; that is, as far as A knows in S, the actual situation could be SA. The statement that A knows ϕ in S is represented by asserting that ϕ holds in every situation that is knowledge accessible from S for A. As is well known, this theory enables the expression of complex interactions of knowledge and time; one can represent both knowledge about change over time and change of knowledge over time.

Again following Moore [19], the state of agent A knowing *what something is* is expressed by using a quantifier of larger scope than the universal quantification over accessible possible worlds. For example, the statement, "In situation s1, John knows who the President is" is expressed by asserting that there exists a unique individual who is the President in all possible worlds accessible for John from s1.

$$\exists_X \forall_{S1A} \text{ k\_acc(john,s1,}S1A) \Rightarrow \text{holds(}S1A\text{,president(}X\text{))}.$$

For convenience, we posit an S5 logic of knowledge; that is, the knowledge accessibility relation, restricted to a single agent, is in fact an equivalence relation on situations. This is expressed in axioms K.1, K.2, and K.3 in table 9. Three important further axioms govern the relation of time and knowledge. (These are discussed at greater length in [9].)

K.4. Axiom of memory: If A knows ϕ in S, then in any later situation, he remembers that he knew ϕ in S.

K.5. A knows all the actions that he has begun, both those that he has completed and those that are ongoing. A also knows what actions are feasible for him now. In terms of possible worlds, this is expressed by stating that if SB is accessible from SA and action E occurs on a time line containing SA and starts no later than SA then E also occurs on a time line containing SB, and the relation between SA and the start and end times of E is the same as the relation between SB and the start end end times of E.

K.6 Knowledge accessibility relations do not cross in the time structure.

Our theory also includes common knowledge. A set of agents U has *common knowledge* of ϕ if they all know ϕ, they all know that they all know ϕ and so on. We represent this by defining a further accessibility relation, "ck_acc(U, S, SA)" (SA is accessible from S relative to the common knowledge of U). This is defined as the transitive closure of links of the form k_acc($A, \cdot, \cdot$) where A is in U. (Of course, transitive closure cannot be exactly defined in a first-order theory; axioms CK.1 and CK.2 define an approximation that is adequate for our purposes.)

Finally, extending Moore's theory, we say that agent A *knows how* to accomplish event E in situation S if there exists an atomic action $E1$ such that A knows that executing $E1$ will accomplish E; that is the execution of $E1$ in any situation knowledge accessible from S entails the occurrence of E. (Table 11.) For example, taking "Dialing a sequence of numbers" to be an atomic action, we can say that Sam knows how to call Margaret if there is a sequence S such that Sam knows that dialing S constitutes calling Margaret.

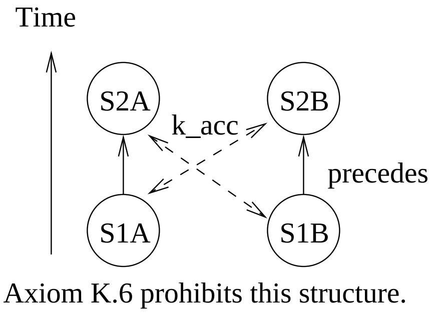


Figure 2: Axiom K.6

Non-logical symbols:
k_acc(A, SA, SB) — SB is accessible from SA relative to A's knowledge in SA.

Axioms:

K.1 $\forall_{A,SA}$ k_acc(A, SA, SA).

K.2 k_acc$(A, SA, SB) \Rightarrow$ k_acc(A, SB, SA)

K.3 k_acc$(A, SA, SB) \wedge$ k_acc$(A, SB, SC) \Rightarrow$ k_acc(A, SA, SC).
(K.1 through K.3 suffice to ensure that the knowledge of each agent obeys an S5 logic: what he knows is true, if he knows ϕ he knows that he knows it; if he doesn't know ϕ, he knows that he doesn't know it.)

K.4 [k_acc$(A, S2A, S2B) \wedge S1A < S2A] \Rightarrow$
$\exists_{S1B}\ S1B < S2B \wedge$ k_acc$(A, S1A, S1B)$.
(Axiom of memory: If agent A knows ϕ at any time, then at any later time he knows that ϕ was true.)

K.5 [occurs(do(A, Z),$S1A, S2A) \wedge S1A \leq SA \wedge$
ordered$(SA, S2A) \wedge$ k_acc$(A, SA, SB)] \Rightarrow$
$\exists_{S1B,S2B}$ occurs(do(A, Z),$S1B, S2B) \wedge$
$S1B \leq SB \wedge$
$[S2A < SA \Rightarrow S2B < SB] \wedge$
$[S2A = SA \Rightarrow S2B = SB] \wedge$
$[SA < S2A \Rightarrow SB < S2B] \wedge$
$[S1A = SA \Rightarrow S1B = SB]$
(An agent knows which actions he has completed, which actions he has begun, and which actions are now feasible.)

K.6 $\neg\exists_{A,S1A,S1B,S2A,S2B}$
$S1A < S2A \wedge S1B < S2B \wedge$ k_acc$(A, S1A, S2B) \wedge$ k_acc$(A, S2A, S1B)$.
(Knowledge accessibility links do not cross in the time structure (Figure 2).)

Table 9: Axioms of Knowledge

Sort: Set of agents (U).

Non-logical symbols:
$A \in U$ — Agent A is in set U.

$\{ \; A1 \ldots Ak \; \}$ — Outfix function. The set of agents containing $A1 \ldots Ak$.

ck_acc(U, SA, SB) — SB is accessible from SA relative to the common knowledge of all agents in U.

Axioms:

CK.1 ck_acc$(U, SA, SB) \Leftrightarrow$
$[\exists_A \; A \in U \wedge$ k_acc$(A, SA, SB)] \vee$
ck_acc$(U, SB, SA) \vee$
$\exists_{SC}$ ck_acc$(U, SA, SC) \wedge$ ck_acc$(U, SC, SB)]$.
Definition of ck_acc as a equivalence relation that includes the k_acc links for all the agents in U.

CK.2 (Induction from k_acc links to ck_acc links.) Let $\Phi(S)$ be a formula with a free situational variable S. Then the closure of the formula

$$\forall_U \; [\forall_{A,SA,SB} \; A \in U \wedge \Phi(SA) \wedge \text{k\_acc}(A, SA, SB) \Rightarrow \Phi(SB)] \Rightarrow$$
$$[\forall_{SA,SB} \; \Phi(SA) \wedge \text{ck\_acc}(U, SA, SB) \Rightarrow \Phi(SB)].$$

is an axiom.

CK.3 $A \in \{A1 \ldots Ak\} \Leftrightarrow [A = A1 \vee \ldots A = Ak]$.

Table 10: Common Knowledge

Non-logical symbol:
know_how(A, E, S) — Agent A knows how to accomplish event E in situation S.

Definition:

KHD.1 know_how$(A, E, S1) \equiv$
$\exists_{E1}$ action$(E1, A) \wedge$
$\forall_{S1A}$ k_acc$(A, S1, S1A) \Rightarrow$
[feasible$(E, S1A) \Rightarrow$ feasible$(E1, S1A)] $\wedge$
$\forall_{S2A}$ occurs$(E1, S1A, S2A) \Rightarrow$ occurs$(E, S1A, S2A)$.

Table 11: Definition of knowing how

In our theory, saying that A knows how to accomplish E in S does *not* require that E is physically feasible in S, only that A knows what primitive action would be involved in accomplishing E. (In fact, if A knows that E is not physically feasible in S, then by our definition he vacuously knows how to accomplish E.)

5 Speech acts

Our theory of speech acts includes representation of a speech act at three different levels of abstraction:

- At the locutionary level, "do(A,utter(U, C))" is the atomic action of speaker A uttering content C to hearers U. ("Utter" here should be broadly construed to include any physical medium of communication.)
- At the illocutionary level, there are four types of events:
  - "inform(A, U, Q)" is the event of agent A informing agents U that fluent Q holds at the time of the beginning of the utterance.
  - "request(AS, AH, P)" is the event of speaker AS requesting AH to carry out plan P.
  - "broadcast_req(AS, U, R)" is the event of speaker AS broadcasting a request that the agents in set U carry out the multi-agent plan R.
  - "commit(A, P)" is the action of agent A committing to his own plan P. It turns out to simplify the analysis if we view this as A talking to himself and asking himself to carry out P.
- At the physical level, "communicate(AS, U)" is the event of speaker AS communicating something to hearers U. A communicate event occurs whenever an utterance action occurs; the primitive "communicate" just abstracts away the content. We introduce this primitive in order to be able to isolate the interaction of the physical theory from the theory of knowledge, communication and planning. Specifically, we posit that a physical theory can include the primitive "communicate" but not any of the other speech act primitives. (See section 9, definition 1.) That is, the physical preconditions for communication and the physical effects of communication are independent of the content being communicated; they depend only on the state of the agents.

One odd feature of our theory is that we shall say nothing more about what kind of thing is the content of an utterance. We introduce no further primitives that use that sort in any way. If you like, you can think of the content as the actual string or signal that is being uttered. (In our formal semantics, it will turn out that the content of an informative act is a fluent of a particular kind and that the content of a broadcast is a multi-agent plan.) What is important here is that an utterance is an atomic action, since our whole theory of planning is based on the foundation that the execution of a plan consists of a series of atomic actions. By axioms A.1, A.2, an agent executes only one atomic action at a time. This is not true, in general, of our other levels of descriptions of speech acts; as we shall see, an informative utterance entails many inform events and a broadcast of a multi-agent plan entails many request events.

Sorts: Content of an utterance (C).

Non-logical symbols:
utter(U, C) — Function: set of agents, content → actional.
communicate(A, U) — Function: agent, set of agents → event.

Axioms:

U.1 occurs(communicate(A, U),$S1, S2$) $\Leftrightarrow \exists_C$ occurs(do(A,utter(U, C)),$S1, S2$).
(A communication event occurs if and only if some utterance occurs.)

U.2 occurs(communicate(A, U),$S1, S2$) $\Rightarrow A \in U$.
(By convention, an agent is always part of his own audience.)

U.3 choice(A, S) $\Rightarrow$ feasible(communicate($A, \{A\}, S$)).
(An agent can always communicate with himself.)

Table 12: Physical axioms of communication

A *multi-agent plan* is an assignment of a plan to every agent. What a plan is will be discussed in section 6.

Table 12 shows the physical axioms governing utterances and communication. Table 13 shows the axioms relating the different speech acts. Table 14 shows the axioms governing the relation of informing and requesting to knowledge. The relation between requests and plan execution must be deferred until after our discussion of plans.

Axioms S.1, C.1, C.2, and C.3 are presented and discussed at length in [9] in a slightly different form.[2] Axiom C.2, it should be noted, precludes the possibility of AS simultaneously communicating N separate messages to N separate agents. This restriction could easily be lifted, either by extending the theory of action to allow concurrent utterances, or by introducing a new physical forms of utterance that entails separate inform events. Also, C.2 is incompatible with many common scenarios, such as unsigned messages, open letters, and messages sent over communication media that are unreliable or have unbounded delays.

Axiom C.3 is so peculiar that some explanation of why it is reasonable is needed here. We begin with the following observation: in general, it is only necessary to distinguish an occurrence of action A1 from an occurrence of action A2 if they have different causal consequences. For instance, in the blocks world, if all you are interested in is the position of blocks, then all that matters in discriminating actions is the ending position of the block being moved; the trajectory through which it moves is immaterial.

Now, in the case of informative acts, the causal consequence of concern is the effect on knowledge states. The main effect of AS informing U of Q is that, when the communication is complete, the agents in U have common knowledge that Q held at the beginning of the communication. Therefore, if $Q1$ and $Q2$ are two informative contents such that the effects on the common knowledge of U following a communication of $Q1$ from AS to U are the same as those effects following a communication of $Q2$, then we can treat the communication of

[2] That paper deals only with communication to a single hearer, not with communication to a set of hearers.

Sorts: Plan (P), Multi-agent plan (R).

Non-logical symbols:
inform(A, U, Q) — Function: agent, set of agents, fluent → event.
request($A1, A2, P$) — Function: agent, agent, plan → event.
broadcast_req(A, U, R) — Function: agent, set of agents, multi-agent plan → event.
commit(A, P) — Function: agent, plan → event.
assignment(R, A) — Function: multi-agent plan, agent → plan.

Axioms:

S.1 occurs(inform(A, U, P),$S1, S2$) $\Rightarrow$ occurs(communicate(A, U),$S1, S2$).

S.2 occurs(request($A1, A2, P$),$S1, S2$) $\Rightarrow$
$\exists_U$ $A2 \in U$ $\wedge$ occurs(communicate($A1, U$),$S1, S2$).

S.3 occurs(broadcast_req(A, U, R),$S1, S2$) $\Rightarrow$ occurs(communicate(A, U),$S1, S2$).
(Axioms S.3 – S.4: inform, request, and broadcast events are types of communication.)

S.4 feasible(commit(A, P),S) $\Leftrightarrow$ feasible(communicate($A, \{A\}, S$).
(It is feasible to commit to a plan if one can communicate with oneself.)

S.5 occurs(commit(A, P),$S1, S2$) $\Rightarrow$ occurs(request(A, A, P),$S1, S2$).
(Committing to P is requesting oneself to do P.)

S.6 occurs(broadcast_req(A, U, R),$S1, S2$) $\wedge$ $A2 \in U$ $\Rightarrow$
occurs(request($A, A2$,assignment($R, A2$)),$S1, S2$).
(Broadcasting a request entails requesting each of the hearers to carry out his assignment.)

S.7 occurs(request(AR, AC, P),$S1, S2$) $\Rightarrow$
[$\exists_R$ occurs(broadcast_req(AR, U, R) $\wedge$ $AC \in U$ $\wedge$ assignment(R, A)=P] $\vee$
[$AC = AR$ $\wedge$ occurs(commit(AR, P),$S1, S2$)].
(The only forms of requests are broadcasts and commitments. A closed-world assumption.)

Table 13: Axioms of Speech Acts

C.1. feasible(communicate(AS, U),$S1$)] $\Rightarrow$
$[\forall_{Q}$ feasible(inform(AS, U, Q),$S1$) $\Leftrightarrow$
$[\forall_{S1A}$ k_acc($AS, S1, S1A$) $\Rightarrow$ holds($S1A, Q$)]].
(If a speaker AS can communicate with hearers U, then AS can inform U of some specific Q if and only if A knows that Q holds at the time he begins speaking.)

C.2. $\forall_{S1,S2,S2A}$ [occurs(inform(AS, U, Q),$S1, S2$) $\wedge$ ck_acc($U, S2, S2A$)] $\Rightarrow$
$\exists_{S1A}$ occurs(inform(AS, U, Q),$S1A, S2A$).
(If AS informs U of Q from $S1$ to $S2$, then in $S2$, the agents in U have common knowledge that AS has informed them of Q.)

C.3. occurs(inform($AS, U, Q1$),$S1, S2$) $\Rightarrow$
[occurs(inform($AS, U, Q2$),$S1, S2$) $\Leftrightarrow$
$[\forall_{S1A}$ ck_acc($U, S1, S1A$) $\Rightarrow$ [holds($S1A, Q1$) $\Leftrightarrow$ holds($S1A, Q2$)]]].
(If AS informs U of $Q1$ over $[S1, S2]$ and the common knowledge of U in $S1$ implies that holds($S1, Q1$) $\Leftrightarrow$ holds($S1, Q2$), then AS has also informed U of $Q2$ over $[S1, S2]$. Conversely, the two events inform($AS, U, Q1$) and inform($AS, U, Q2$) co-occur only if $Q1$ and $Q2$ are related in this way.)

C.4 $\forall_{AS,U,Q,S}$ know_how(AS,inform(AS, U, Q),S).
(An agent always knows how to express any content that he wishes to communicate.)

C.5 $\forall_{A,R,S}$ feasible(communicate(AS, U),S) $\Rightarrow$ feasible(broadcast_req(AS, U, R),S).
(If AS can communicate with hearers U then AS can broadcast to U a request for any multi-agent plan.)

C.6 $\forall_{AS,U,R,S}$ know_how(AS,broadcast_req(AS, U, R),S).
(An agent always knows how to express any multi-agent plan that he wishes to broadcast.)

Table 14: Axioms of Communication and Knowledge

$Q1$ and the communication of $Q2$ as *the same action*; they, so to speak, attain the same end state via different trajectories. And a sufficient condition to ensure this is that U has common knowledge at the *start* of the communication that $Q1$ and $Q2$ are equivalent.

For example, if Jack and Jane share the knowledge that George Bush is the President and that 1600 Pennsylvania Avenue is the address of the White House, then the action of Jack informing Jane that Bush is at the White House is identical to the act of Jack informing Jane that the President is at 1600 Pennsylvania Avenue. If they do not share this knowledge, then these two acts are different.

As we will discuss in section 8, the purpose of this rather convoluted formulation is to ensure that the overall theory is consistent; it turns out that simpler and more natural formulations lead to inconsistencies.

6 Plans

We now have the foundations on which to develop the main point of this paper, the theory of multi-agent plans. The informal theory is rather complex, and we develop it in five stages. First, we discuss how a plan P is specified. Second, we define four key predicates on plans:

1. Agent AC executes plan P from $S1$ to $S2$.

2. Agent AC executes plan P deliberately from $S1$ to $S2$.

3. Agent AC executes plan P deliberately from $S1$ to $S2$, following the protocol described in section 2.

4. Agent SA knows in situation S that he can execute P successfully, assuming that all agents follow the protocol.

6.1 Specifying a plan

In our theory, any plan P has a unique actor AC=actor(P), and an execution of P consists of the performance of a sequence of actions of AC.

A plan P is specified in terms of two predicates. First, "next_step($E, P, S1, S2$)" asserts that E is a possible next step in $S2$ of an instance of plan P begun in $S1$. Second, "succeeds($P, S1, S2$)" asserts that plan P, begun in $S1$, terminates successfully in $S2$. (This does not, of course, mean that plans in practice have to be written in terms of these two predicates, any more than programs have to be written in machine language. One can define a higher-level planning language whose semantics are defined in terms of "next_step" and "succeeds", just as one can define a higher-level machine language, which is translated into machine language.) A few examples:

Example 1 Consider a blocks world situation s1 where A, B, C, and D are on the table, and where agent Joe has the goal, "on(A,B) and on(C,D)." The plan p1="put A on B; put C on D" can be defined by the following axioms:

$\forall_{E,S2}$ next_step(E,p1,s1,$S2$) $\Leftrightarrow$
[[$S2$ =s1 $\Rightarrow$ E=do(joe,put_on(a,b))] $\wedge$
[occurs(do(joe,put_on(a,b)),s1,$S2$) $\Rightarrow$ E=do(joe,put_on(c,d))]].

$\forall_{S2}$ succeeds(p1,s1,$S2$) $\Leftrightarrow$ holds($S2$,on(a,b)) $\wedge$ holds($S2$,on(c,d)).

Thus, in $S1$, p1 requires Joe to put A on B, and when that is complete, p1 requires him to put C on D. In all other situations, "next_step" is unconstrained; Joe can do what he wants.

Example 2: In the same starting situation and with the same goal, let p2 be the plan, "In either order, put A on B and put C on D." Plan p2 can then be defined by the following axioms:

$\forall_{E,S2}$ next_step(E,p1,s1,$S2$) $\Leftrightarrow$
[[$S2$ =s1 $\Rightarrow$ [E=do(joe,put_on(a,b)) $\vee$ E=do(joe,put_on(c,d))]] $\wedge$
[occurs(do(joe,put_on(a,b)),s1,$S2$) $\Rightarrow$ E=do(joe,put_on(c,d))] $\wedge$
[occurs(do(joe,put_on(c,d)),s1,$S2$) $\Rightarrow$ E=do(joe,put_on(a,b))]].

$\forall_{S2}$ succeeds(p1,s1,$S2$) $\Leftrightarrow$ holds($S2$,on(a,b)) $\wedge$ holds($S2$,on(c,d)).

Example 3: In the same starting situation, suppose the goal is "Get A on B within an hour," and it is known that any single action takes at most one minute. Let p3 be the following plan:

> If there are less than two minutes left in the hour and A is not yet on B, then put A onto B. (We have to allow two minutes: The first so that the agent can complete whatever other action he is working on, the second to complete the action of putting A on B.) Otherwise, do whatever you want, except that, if A is not on B, then don't put anything onto A or anything other than A onto B.

We can formalize this plan as follows:

$\forall_{E,S2}$ next_step(E,p3,s1,$S2$) $\Leftrightarrow$
[time($S2$) $-$ time(s1) $>$ 58*minute $\Rightarrow$ E=do(joe,put_on(a,b))] $\wedge$
[time($S2$) $-$ time(s1) $\leq$ 58*minute $\Rightarrow$
[$\forall_X$ E $\neq$do(joe,put_on(X,a) $\wedge$ [$\forall_Y$ E =do(joe,put_on(Y,b)) $\Rightarrow$ Y=a.]]]

$\forall_{S2}$ succeeds(p3,s1,$S2$) $\Leftrightarrow$ time($S2$) $\leq$ time(s1) + hour $\wedge$ holds($S2$,on(a,b)).

We can now define the successful execution of a plan: Agent AC *begins the execution* of plan P over the interval $[S1, S2]$ if at every choice point SM for AC between $S1$ and $S2$, AC carries out an action that is a possible next step for P. (In this definition, saying that AC "begins the execution of P" does not necessarily imply that AC will be able to complete P.) Agent AC *successfully executes* P over $[S1, S2]$ if he begins P over $[S1, S2]$ and P succeeds over $[S1, S2]$.

6.2 Deliberate execution of a plan

Agent AC *knowingly* or *deliberately executes* plan P over interval $[S1, S2]$ if (1) at every choice point SM between $S1$ and $S2$, AC carries out an action that he knows to be a next step of P in SM; and (2) AC knows that P succeeds over $[S1, S2]$. (In the spirit of our theory of knowledge, "knows" here means just "can in principle deduce from what he knows". There is no implication that AC is thinking about P or is interested in P.)

The distinction between simple execution and deliberate execution embodies our theory of knowledge preconditions, discussed at greater length in [8]. Suppose that Mary's phone number is 546-9845 but that Sam does not know this; i.e. there are worlds that are knowledge accessible for Sam in which Mary has a different phone number. Sam can execute the plan

"Dial Mary's phone number" because dialing 546-9845 is a physically possible action, but he cannot deliberately execute the plan, because he does not know that this action constitutes an execution of the plan. Formally, there is no action E that is a next step of the plan in all possible worlds that are knowledge accessible for Sam.

6.3 Following the Protocol

To impose our protocol, we need to consider the source AR of plan P as well as its actor. Recall that, in our protocol, AC reserves certain time slots for AR. During a time reserved for AR, AC will work on P as long as P does not require him to carry out an action governed by someone other than AR. During a time not reserved for AR, the only constraint that P may impose is that AC refrain from actions governed by AR. It is the responsibility of the requestor AR to formulate P in a way that satisfies the protocol; otherwise, AC may abandon P.

For example, consider the starting situation and the goal of Example 1. Suppose that AR governs manipulations of the blocks, but AC is the only agent with physical access to the blocks. Then AR can request that AC carry out the following plan:

```
When you are at a choice point at a time reserved for me:
    if block A is not on B then put A on B
    else if C is not on D then put C on D.

At any other time, do anything you want with the following exceptions:
Do not put anything on A or C; do not put anything other than A on B; do not
put anything other than C on D; do not take A off B.
```

If AC is working on P and he arrives at a choice point when the time is reserved for AR, but every feasible action that he knows to be a next step of P is governed by some actor other than AR, then AC abandons P (abandonment condition 1). If the time is not reserved for AR and AC is not certain that every feasible action prohibited by P is governed by AR, then AC abandons P (abandonment condition 2). Otherwise, AC carries out some action that he knows to be a next step of P.

The protocol insures that AC can always accommodate requests from any number of different agents. It does not, however, deal with conflicts between two different requests from a single agent. To take care of this, we posit a rule that AC need only accept one request at a time from AR; that is, if AC is still working on one request of AR in situation S, he may ignore any new request of AR that he received in S.

The following predicates are used to describe execution of the plan within the protocol:

The predicate exec_cont($E, P, AC, AR, S1, S2$) asserts that E is a next step in situation $S2$ of the execution, within our protocol, of plan P with actor AC and source AR beginning in $S1$, where time($S2$) is reserved by AC for AR.

The predicates abandon1($P, AC, AR, S1, S2$) and abandon2($P, AC, AR, S1, S2$) are the two abandonment conditions. The predicate terminates($P, AC, AR, S1, S2$) characterizes the situation $S2$ in which AC may quit working on P; either AC knows in $S2$ that the plan has succeeded or one of the abandonment conditions holds.

The predicate begin_plan($P, AC, AR, S1, S2$) asserts that AC begins the execution of P over the closed interval $[S1, S2]$ following the protocol. The predicate completes($P, AC, AR, S1, S2$) asserts that AC completes the execution of P over interval $[S1, S2]$ following the protocol.

The predicate working_on($P, AC, AR, S1, S2$) asserts that AC is working on P at the request of AR beginning in $S1$ and continuing through $S2$. The predicate accepts_req($P, AC, AR, S1$) asserts that in $S1$, AC accepts the request that he work on P. Axiom Q.5 asserts that AC is working on P if AC accepted a request to work on P and AC has begun work on P but not finished it. Axiom Q.6 asserts that, if AC is not working on some other request of AR's in $S1$ and AR request him to work on P, then he will accept the request. Thus, axioms Q.5 and Q.6 cause "accepts_req" and "working_on" to be mutually recursive working backward through time.

6.4 Cooperative agents and multi-agent plans

Since the time structure includes all physically possible actions of agents, the description of an agent as cooperative is construed as a property of a time interval rather than of the agent. Over the time structure as a whole, there are intervals in which an agent does what he is requested and those in which he does not; the theory singles out the former for special interest. We say that agent A is *cooperative* up to S if the following holds: for any SM before S, if A accepts a request in SM then he attempts to carry it out over some interval $[SM, SN]$, where SN and S are on the same time line. Moreover if $SN \leq S$, then A's attempt terminates in SN either because he achieves the request or because he encounters one of the protocol's abandonment conditions.

We say that situation S is *socially possible*, notated soc_poss(S), if all agents are cooperative up to S. An unbounded interval I is socially possible, notated soc_poss_int(I), if every situation in I is socially possible.

We now arrive at our objective. *Agent AC can successfully execute plan P in situation $S0$,* notated "executable($P, AC, S0$)", if the following holds: Suppose that AC commits to P over the interval $[S0, S1]$ where $S1$ is socially possible. Let I be any socially possible u-interval following $S1$. Then P completes in I. In other words, if AC works on his plan and all other agents cooperate, then the plan will be successfully completed.

Sorts: Plans (P).

Non-logical symbols:

actor(P) — Function: plan → agent.

next_step($E, P, S1, S2$) — Predicate. Action E in situation $S2$ is a possible next step in the carrying out of P starting in $S1$.

succeeds($P, S1, S2$) — Plan P starting in situation $S1$ terminates successfully in situation $S2$.

know_next_step($E, P, A, S1, S2$) — A, the actor of P, knows in $S2$ that next_step($E, P, S1, S2$).

know_succeeds($P, A, S1, S2$) — A, the actor of P, knows in $S2$ that P starting in $S1$ succeeds in $S2$.

Definitions:

PD.1 know_next_step($E, P, A, S1, S2$) $\equiv$
A=actor(P) $\wedge$ feasible($E, S2$) $\wedge$
$\forall_{S1A,S2A}$ [k_acc($A, S2, S2A$) $\wedge$ k_acc($A, S1, S1A$) $\wedge$ $S1A \leq S2A$] $\Rightarrow$
next_step($E, P, S1A, S2A$)

PD.2 know_succeeds($P, A, S1, S2$) $\equiv$
$\forall_{S1A,S2A}$ [k_acc($A, S2, S2A$) $\wedge$ k_acc($A, S1, S1A$)] $\Rightarrow$ succeeds($P, S1A, S2A$)

Axioms:

P.1 next_step($E, P, S1, S2$) $\Rightarrow$ action(E,actor(P)).

P.2 Comprehension axiom on plans. See section 8.

Table 15: Planning

Non-logical symbols:

Note: In all the primitives below, AC is the agent carrying out the plan, and AR is the agent who requested the execution of the plan.

governs(AR, E) — Agent AR "governs" action E.

reserved(T, AC, AR) — Time T is in a block reserved by AC to carry out plans whose source is AR.

reserved_block(T, AC, AR, D) — The time interval from T to $T + D$ is reserved by AC for plans of AR.

min_reserve_block, delay_time — Constant durations. Agent AC must reserve time blocks of length at least min_reserve_block separated by no more than delay_time. (See axiom Q.2)

working_on$(P, AC, AR, S1, S2)$ — In $S2$, AC is still working on a previous request P of AR's, accepted in $S1$.

accepts_req$(P, AC, AR, S1)$ — At time $S1$, AC accepts request P from AR.

exec_cont$(E, P, AC, AR, S1, S2)$, abandon1$(P, AC, AR, S1, S2)$,
abandon2$(P, AC, AR, S1, S2)$ – As defined on page 26.

begin_plan$(P, AC, AR, S1, S2)$ — Predicate. AC begins plan P in interval $[S1, S2]$ at the request of AR.

completes$(P, AC, AR, S1, SZ)$ — Plan P is completed in the interval $[S1, SZ]$.

attempt_toward$(P, AC, AR, S1, S2)$ — An attempt of AC to carry out P begins in $S1$ and proceeds on the timeline leading toward $S2$.

soc_poss(S) — All agents behave according to social norms up to S.

soc_poss_int(I) — All agents behave according to social norms throughout u-interval I.

executable(P, AC, S) — Plan P is executable by agent AC in situation S.

know_achievable(Q, P, AC, S) — Agent AC knows in S that he can achieve Q through the execution of P.

Table 16: Multi-agent plans: Primitives

Definitions:

QD.1 $\text{reserved\_block}(T, AR, AC, D) \equiv$
$\forall_{T1}\ T \leq T1 \leq T + D \Rightarrow \text{reserved}(T1, AC, AR)$.

QD.2 $\text{exec\_cont}(E, P, AC, AR, S1, S2) \equiv$
$\text{choice}(AC, S2) \land \text{reserved}(\text{time}(S2), AC, AR) \land \text{know\_next\_step}(E, P, AC, S1, S2) \land$
$\neg\exists_{A3 \neq AR}\ \text{governs}(A3, E)$.

QD.3 $\text{abandon1}(P, AC, AR, S1, S2) \equiv$
$\text{choice}(AC, S2) \land \text{reserved}(\text{time}(S2), AC, AR) \land \neg\exists_{E}\ \text{exec\_cont}(E, P, AC, AR, S1, S2)$.

QD.4 $\text{abandon2}(P, AC, AR, S1, S2) \equiv$
$\text{choice}(AC, S2) \land \neg\text{reserved}(\text{time}(S2), AC, AR) \land$
$\exists_{E}\ \text{action}(E, AC) \land \neg\text{governs}(AR, E) \land \neg\text{know\_next\_step}(E, P, AC, S1, S2)$.

QD.5 $\text{terminates}(P, AC, AR, S1, S2) \equiv$
$\text{know\_succeeds}(P, AC, S1, S2) \lor \text{abandon1}(P, AC, AR, S1, S2) \lor$
$\text{abandon2}(P, AC, AR, S1, S2)$

QD.6 $\text{begin\_plan}(P, AC, AR, S1, S2) \equiv$
$S1 \leq S2 \land$
$\forall_{SM}\ S1 \leq SM < S2 \Rightarrow$
$\quad [\neg\text{terminates}(P, AC, AR, S1, SM) \land$
$\quad [\text{choice}(AC, SM) \Rightarrow$
$\quad \exists_{E}\ \text{know\_next\_step}(E, P, AC, S1, SM) \land \text{leads\_towards}(E, SM, S2)]]$.

QD.7 $\text{completes}(P, AC, AR, S1, SZ) \equiv$
$\text{begin\_plan}(P, AC, AR, S1, SZ) \land \text{know\_succeeds}(P, AC, S1, SZ)$

QD.8 $\text{attempt\_toward}(P, AC, AR, S1, S2) \equiv$
$\text{begin\_plan}(P, AC, AR, S1, S2) \lor$
$\exists_{S3}\ S1 \leq S3 < S2 \land \text{begin\_plan}(P, AC, AR, S1, S3) \land \text{terminates}(P, AC, AR, S1, S3)$.
($S2$ lies in a time-line in which AC attempts to execute P starting in $S1$.)

QD.9 $\text{soc\_poss}(S1) \equiv$
$\forall_{P,AC,AR,SP}\ SP < S1 \land \text{accepts\_req}(P, AC, AR, SP) \Rightarrow$
$\text{attempt\_toward}(P, AC, AR, SP, S1)$.

QD.10 $\text{soc\_poss\_int}(I) \equiv \forall_{S}\ \text{elt}(S, I) \Rightarrow \text{soc\_poss}(S)$.

QD.11 $\text{executable}(P, AC, S0) \equiv$
$\forall_{S1,I}\ [\text{soc\_poss\_int}(I) \land \text{elt}(S1, I) \land \text{occur}(\text{commit}(AC, P)), S0, S1)] \Rightarrow$
$\exists_{S2}\ \text{elt}(S2, I) \land \text{completes}(P, AC, AC, S1, S2)$.

QD.12 $\text{know\_achievable}(Q, P, AC, S1) \equiv$
$\forall_{S1A}\ \text{k\_acc}(AC, S1, S1A) \Rightarrow$
$[\text{executable}(P, AC, S1A) \land \forall_{SZA}\ \text{completes}(P, AC, AC, S1A, SZA) \Rightarrow \text{holds}(SZA, Q)]$.

Table 17: Multi-agent plans: Definitions

Axioms:

Q.1 reserved$(T, AC, AR1) \wedge$ reserved$(T, AC, AR2) \Rightarrow AR1 = AR2$.
(Any time is reserved for at most one agent.)

Q.2 $\forall_{AC,AR,T}\ \exists_{T1}\ T1 \leq T+$delay_time $\wedge$ reserved_block$(T1, AR, AC,$min_reserve_block$)$.
(At any time T one can be sure that, within time delay_time, there will be a block of time that AC reserves for AR of length at least min_reserve_block.)

Q.3 governs$(AR1, Z) \wedge$ governs$(AR2, Z) \Rightarrow AR1 = AR2$.
(At most one person governs any given action Z.)

Q.4 $\neg$governs$(AR,$do$(AC,$wait$))$.
(No one governs the action of waiting.)

Q.5 working_on$(P, AC, AR, S0, S1) \Leftrightarrow$
accepts_req$(P, AC, AR, S0) \wedge$ begin_plan$(P, AC, AR, S0, S1) \wedge$
$\neg$terminates$(P, AC, AR, S0, S1)$.
(In $S1$, AC has not finished a plan P that he accepted from AR earlier.)

Q.6 accepts_req$(P, AC, AR, S) \Leftrightarrow$
$\exists_{S2}$ occurs(request(AR, AC, P),$S2, S) \wedge$
$\neg\exists_{PB,S0}\ PB \neq P \wedge$ working_on$(PB, AC, AR, S0, S)$.
(Agent AC accepts plan P from AR in S if there are no outstanding requests from AR.)

Q.7 accepts_req$(P, AC, AR, S) \wedge$ k_acc$(AC, S, SA) \Rightarrow$ accepts_req(P, AC, AR, SA).
(Agent AC knows when he has accepted a request.)

Table 18: Multiagent Plans: Axioms

Non-logical symbols:
instance($E1, E2, S$): Predicate. Event $E1$ is an instance of event $E2$ in situation S.
opportunity($S2, AC, AR, Q$): Predicate. See text.
first_opportunity($S1, AC, AR, S0, Q$): Predicate. See text.
max_action_time: Duration constant.

Definitions:

MD.1 $\mathrm{instance}(E1, E2, S) \equiv \exists_A \mathrm{action}(E1, A) \wedge [\forall_{S2} \mathrm{occurs}(E1, S, S2) \Rightarrow \mathrm{occurs}(E2, S, S2)]$.

MD.2 $\mathrm{opportunity}(S1, AC, AR, Q) \equiv$
$\mathrm{choice}(AC, S1) \wedge \mathrm{reserved}(\mathrm{time}(S1), AC, AR) \wedge \mathrm{holds}(S1, Q)$.

MD.3 $\mathrm{first\_opportunity}(S1, AC, AR, S0, Q) \equiv$
$S0 \leq S1 \wedge \mathrm{opportunity}(S1, AC, AR, Q) \wedge$
$\neg\exists_{SM}\ S0 \leq SM < S1 \wedge \mathrm{opportunity}(SM, AC, AR, Q)$.

Axiom:

M.1 $\forall_{A,Z,S1,S2}\ \mathrm{occurs}(\mathrm{do}(A, Z), S1, S2) \Rightarrow \mathrm{time}(S2) \leq \mathrm{time}(S1) + \mathrm{max\_action\_time}$

Table 19: Useful abbreviations in plan statement

7 Examples

We now return to the examples in section 1, and show how the plans are represented in our language.

In representing plans, the following defined predicates will be useful. An utterance action A is an *instance* of speech act E in situation S if the execution of A starting in S entails the concurrent occurence of E. Situation $S1$ is an *opportunity* for agent AC to react to a circumstance Q (a fluent) mentioned in a request of agent AR if $S1$ is a choice point for AC, AC has reserved $S1$ for AR and Q holds in $S1$. We also have a predicate stating that $S1$ is the *first opportunity* of this kind following $S0$. The constant max_action_time is the maximum time needed for any primitive action. (Table 19.)

7.1 Problem 1

Ann and Barry are sitting together. Ann knows that Barry has her cell phone. Infer that Ann can get her cell phone back by asking Barry to give it to her.

Representation: Let s0 be the starting situation. The plan p.1.1 that Ann requests Barry to carry out is as follows: As soon as Barry reaches a time that he has reserved for Ann, he should give her the cell phone. In the meantime, he may do whatever he likes except giving it to someone else. His part is complete when he no longer has the cell phone. Symbolically,

$\forall_{E,S2}$ next_step(E, p.1.1, s0, $S2$) $\Leftrightarrow$
action(E,barry) $\wedge$
[[reserved(time($S2$),barry,ann) $\wedge$ has(barry,cellPhone)] $\Rightarrow$
E=do(barry,give(cellPhone,ann))] $\wedge$

$[\neg$reserved(time($S2$),barry,ann) $\Rightarrow$
$\neg\exists_{AX}$ $AX \neq$ann $\wedge$ E =do(barry,give(cellPhone,AX))].

$\forall_{S2}$ succeed(p.1.1,barry,$S2$) $\Leftrightarrow$ $\neg$holds($S2$,has(barry,cellPhone)).

Ann's plan p.1.2 is that, at her first opportunity, she will request Barry to give her the cell phone. The plan succeeds when she has the cell phone.

$\forall_{E,S2}$ next_step(E,p.1.2,s0,$S2$) $\Leftrightarrow$
action(E,ann) $\wedge$
[first_opportunity($S2$,ann,ann,s0,true_f) $\Rightarrow$
instance(E,request(ann, {barry}, p.1.1), $S2$)]

$\forall_{S2}$ succeeds(ann,s0,$S2$) $\Leftrightarrow$ holds($S2$,has(ann,cellPhone)).

To justify the inference that Ann knows that this plan is feasible and achieves the goal — that is, to prove the proposition "know_achievable(has(ann,cellPhone),p.1.2,ann,s0)" — the domain theory and problem specification must include the following constraints, or something similar. It should be emphasized, since it is the whole point of defining this complex and restrictive protocol, that this proof establishes that the plan is correct *regardless of whatever other plans Ann and Barry are involved in, whatever any other agents are doing, and whatever other requests are made.*

- X can give O to Y if X and Y are nearby and X has O.
- Ann knows that Barry is nearby and will remain so for at least the time period 2 $\times$ delay_time + max_action_time. (Ann may be delayed for delay_time before she can make the request; then max_action_time may pass before the request is complete; then delay_time may pass until Barry can act on her request.)
- The only way to cease to have an object is to give it to someone else.
- A person knows whether or not he has the cell phone.
- Ann is physically able to communicate to Barry.
- Barry is not currently working on some previous request of Ann's.
- The trickiest part is actually to ensure that Barry does not give away the cell phone to someone else until Ann has been able to finish speaking her request. There are a number of ways in which this can be kludged; perhaps the simplest is to specify physical axioms that guarantee that no other agents will be nearby during that delay.

7.2 Problem 2

Carol wishes to email David, but does not know his email address. However, Carol knows that Emily knows David's email address, and Emily is nearby. Infer that Carol can email David by executing the following plan: Ask Emily to tell me David's email address. After she answers, email David.

Representation: Assume that the term "eaddress(A, X)" denotes the fluent[3] of X being A's email address and that the function "email(AR, X, C)" denotes the actional of emailing content C to recipient AR at address C. Let s0 be the starting situation.

The plan p.2.1 that Emily is supposed to carry out is that, at her first opportunity, she should tell David's address to Carol. The plan is complete when this is done.

$\forall_{E,S2}$ next_step(E,p.2.1,s0,$S2$) $\Leftrightarrow$
action(E,emily) $\wedge$
[first_opportunity($S2$,emily,carol,$S1$,nearby(emily,carol)) $\Rightarrow$
$\exists_X$ instance(E,inform(emily,{emily,carol},eaddress(X,david)),$S2$)].

$\forall_{S2}$ succeeds(p.2.1,s0,$S2$) $\Leftrightarrow$
$\exists_{SA,SB,X,E}$ $SB \leq S2 \wedge$ occurs(E, SA, SB) $\wedge$
instance(E,inform(emily,{emily,carol},eaddress(X,david)),SA).

The condition (fluent) q.2.1 that Carol knows David's email address is represented:

$\forall_S$ holds(S,q.2.1) $\Leftrightarrow$
$\exists_X \forall_{SA}$ k_acc(carol,S, SA) $\Rightarrow$ holds(SA,eaddress(X,david)).

Carol's plan p.2.2 is (a) to request Emily to carry out p.2.1; (b) to email David content c0 at her first opportunity when q.2.1 is satisfied. (Note that she has to wait until a time that she has reserved for her own plans.) The plan succeeds when she has emailed David.

$\forall_{E,S2}$ next_step(E,p.2.2,s0,$S2$) $\Leftrightarrow$
action(E,carol) $\wedge$
[first_opportunity($S2$,carol,carol,s0,true_f) $\Rightarrow$
instance(E,request(carol,emily,p.2.1),$S2$)] $\wedge$
[first_opportunity($S2$,carol,carol,s0,q.2.1) $\Rightarrow$
$\exists_X$ holds($S2$,eaddress(X,david)) $\wedge$ E=do(carol,email(david,X,c0))].

$\forall_{S2}$ succeeds(p.1.1,s0,$S2$) $\Leftrightarrow$
$\exists_{SA,SB,X}$ s0 $\leq SA < SB \leq S2 \wedge$ occurs(do(carol,email(david,X,c0)),SA, SB).

To prove this plan correct requires the following assumptions, in addition to those listed in the problem specification.

- It is universally known that email addresses don't change (or, at least, Carol knows that Emily knows this.)
- Carol and Emily can communicate if they are nearby.
- Carol knows that Emily will remain nearby for at least the time period max_action_time + delay_time.
- It is always possible to send email.

[3] This has to be a fluent because it is unknown, and thus varies from one epistemically possible world to another, even if it taken to be constant over time.

Problem 3: A warehouse is manned by a collection of robots, one on each floor. There is an elevator used for moving packages from one floor to another. The robots can communicate by radio. Each robot can carry out the following actions: call for the elevator; load a package on the elevator; unload a package from the elevator; communicate a fact to another robot; or broadcast a request to all the other robots.

In the starting situation, a particular robot, called the "hero", wants to get a particular package labelled b1. He knows that it is on some other floor, but he does not know where. Infer that the following plan will result in the hero having b1:

```
I will broadcast the following request:
    If you have package b1, then {
      call the elevator;
      when the elevator arrives, load b1;
      announce that b1 is on the elevator; }
When I know that b1 is on the elevator, I will call the elevator;
When the elevator arrives, I will unload b1 off the elevator.
```

Representation: Tables 20 and 21 give an axiomatization for the elevator world. Tables 22, 23, and 24 give the representation of this plan.

Sorts: Packages (B).

Note: We conflate a robot with the floor that it is on.

Non-logical symbols:
Predicate: owns(A, B)
Fluent functions: elevator_at(A), on_elevator(B), has(A, B).
Actional constant: call.
Actional functions: load(B), unload(B).

Duration constants:
max_elevator_wait — The longest delay between calling an elevator and its arrival.
min_elevator_open — The minimum time an elevator will wait on its floor.

Axioms

Comparative time lengths

E.1 $0 <$ max_action_time $<$ min_elevator_open.

E.2 $0 <$ max_elevator_wait.

Atemporal axiom:

E.3 $\forall_B \, \exists^1_A$ owns(A, B).
(Unique owner for each package.)

Causal axioms:

E.4 occurs(do(A,call),$S1, S2$) $\Rightarrow$ within($S2$, max_elevator_wait, elevator_at(A)).

E.5 occurs(do(A,load(B)),$S1, S2$) $\Rightarrow$
$\exists_{SM} \; S1 < SM < S2 \wedge$ throughout($SM, S2$,on_elevator(B)).

E.6 occurs(do(A,unload(B)),$S1, S2$) $\Rightarrow$ holds($S2$,has(A, B))

Elevator stays in place for minimum time and during other actions

E.7 holds(S,elevator_at(A)) $\Rightarrow$
$\exists_{S1,S2} \; S1 \leq S \leq S2 \wedge$ time($S1$) + min_elevator_open $\leq$ time($S2$) $\wedge$
throughout($S1, S2$,elevator_at(A))

E.8 occurs(do(A,load(B)),$S1, S2$) $\wedge \; S1 \leq SM < S2 \Rightarrow$ holds(SM,elevator_at(A))

E.9 occurs(do(A,unload(B)),$S1, S2$) $\wedge \; S1 \leq SM < S2 \Rightarrow$ holds(SM,elevator_at(A))

E.10 occurs(do(A,load(B)),$S1, S2$) $\Rightarrow$ throughout($S1, S2$,or_f(has(A, B),on_elevator(B))).

E.11 occurs(do(A,unload(B)),$S1, S2$) $\Rightarrow$
throughout($S1, S2$,or_f(has(A, B), on_elevator(B))).

Table 20: Elevator world: Part I

Domain constraints

E.12 $\text{holds}(S,\text{on\_elevator}(B)) \mathbin{\dot{\vee}} \exists^1_A \text{ holds}(S,\text{has}(A,B))$

E.13 $\forall_S \exists^1_A \text{ holds}(S,\text{elevator\_at}(A))$

Preconditions

E.14 $\text{feasible}(\text{do}(A,\text{load}(B)),S) \Leftrightarrow$
$\text{choice}(A,S) \land \text{holds}(S,\text{has}(A,B)) \land \text{holds}(S,\text{elevator\_at}(A))$

E.15 $\text{feasible}(\text{do}(A,\text{unload}(B)),S) \Leftrightarrow$
$\text{choice}(A,S) \land \text{holds}(S,\text{on\_elevator}(B)) \land \text{holds}(S,\text{elevator\_at}(A))$

E.15 $\text{feasible}(\text{do}(A,\text{call}),S) \Leftrightarrow$
$\text{choice}(A,S) \land \neg\text{holds}(S,\text{elevator\_at}(A))$

E.16 $\forall_{A,U}\ A \in\text{robots} \Rightarrow \text{feasible}(\text{communicate}(A,\text{robots}),S)$.

Frame axioms: (Note: there is also a frame axiom for "elevator_at", but it has no importance. The frame axioms for "on_elevator" are consequences of E.17, E.18, and E.12.)

E.17 $\text{holds}(S1,\text{has}(A,B)) \land \neg\text{holds}(S2,\text{has}(A,B)) \land S1 < S2 \Rightarrow$
$\exists_{S3,S4}\ S3 < S2 \land S1 < S4 \land \text{ordered}(S2,S4) \land \text{occurs}(\text{do}(A,\text{load}(B)),S3,S4)$.
(A can only cease to have B if he executes a "load" action).

E.18 $\neg\text{holds}(S1,\text{has}(A,B)) \land \text{holds}(S2,\text{has}(A,B)) \land S1 < S2 \Rightarrow$
$\exists_{S3,S4}\ S3 < S2 \land S1 < S4 \land \text{ordered}(S2,S4) \land \text{occurs}(\text{do}(A,\text{unload}(B)),S3,S4)$
(A can only come to have B if he executes a "unload" action).

Knowledge axioms:

E.19 $\text{k\_acc}(A,SA,SB) \Rightarrow [\text{holds}(SA,\text{elevator\_at}(A)) \Leftrightarrow \text{holds}(SB,\text{elevator\_at}(A))]$.
(You know whether the elevator is on your floor.)

E.20 $\text{k\_acc}(A,SA,SB) \Rightarrow [\text{holds}(SA,\text{has}(A,B)) \Leftrightarrow \text{holds}(SB,\text{has}(A,B))]$.
(You know whether you have a package.)

E.21 $[\text{k\_acc}(A,SA,SB) \land \text{holds}(SA,\text{elevator\_at}(A))] \Rightarrow$
$[\text{holds}(SA,\text{on\_elevator}(B)) \Leftrightarrow \text{holds}(SB,\text{on\_elevator}(B))]$.
(If the elevator is on your floor, then you know what packages are in it.)

Governance of actions

E.22 $\text{governs}(AG,\text{do}(AC,Z)) \Leftrightarrow [\exists_B \text{ owns}(AG,B) \land [Z{=}\text{load}(B) \lor Z{=}\text{unload}(B)]]$.
(Loading and unloading a package are governed by the owner of the package. Other actions are ungoverned.)

E.23 $\forall_A\ A \in\text{robots}$.
(The set "robots" includes all the agents.).

Table 21: Elevator world: Part II

Constants:
el1 — the hero's plan.
r2 — the hero's broadcast
b1 — the package being sent.
max_el2b_time — maximum time needed from the start of plan el2 until every agent can be sure that the hero has the package.
robots — the set of all robots.

Functions:
el2(A) — The task assigned to A.

el1_q1($S1, S0$) etc. — This is a flag used in stating the plan el1. $S0$ is the situation in which the hero starts to execute plan el1. In the situation S where el1_q1($S, S0$) is true, the hero should execute the first step of plan el1. Similarly, el1_q2($S, S0$) and el1_q3($S, S0$) are flags indicating that the hero should execute the second and third step of el1 in $S1$; and el2_q1($A, S, S0$), el2_q2($A, S, S0$), and el2_q3($A, S, S0$) are flags indicating that agent A should execute the first, second, and third steps of plan el2(A).

loaded_since(B, A, T) — Fluent. Package B was loaded on the elevator, which was at A, at some time before the present but after T.

Table 22: Problem statement: Predicates

Definitions:

XD.1 holds(S,know_loaded(A, B)) $\equiv$
reserved_block(time(S),A, A,max_action_time + max_elevator_wait) $\wedge$
$\forall_{SA}$ k_acc(A, S, SA) $\Rightarrow$ holds(SA,on_elevator(B))

XD.2 el1_q1($S, S0$) $\equiv$
first_opportunity(S,hero,hero,$S0$,true_f).

XD.3 el1_q2a($S, S0$) $\equiv$
first_opportunity(S,hero,hero,$S0$, know_loaded(hero,b1))

XD.4 el1_q3($S, S0$) $\equiv$
first_opportunity(S, hero, hero, $S0$, and_f(elevator_at(hero), on_elevator(b1))).

XD.5 el1_q2($S, S0$) $\equiv$
el1_q2a($S, S0$) $\wedge \neg\exists_{SA}$ $S0 \leq SA \leq S$ $\wedge$ el1_q3($SA, S0$)

XD.6 holds(S,el2_q1_f(A)) $\equiv$
holds(S,has(A, b1)) $\wedge$ $\neg$holds(S,elevator_at(A)) $\wedge$
reserved_block(time(S),A,hero,
max_action_time + max_elevator_wait + max_action_time + max_action_time).

XD.7 el2_q1($A, S, S0$)) $\equiv$ first_opportunity(S, A, hero, $S0$,el2_q1_f(A))

XD.8 holds(S,el2_q2_f(A)) $\equiv$
holds(S,has(A, b1)) $\wedge$ holds(S,elevator_at(A))) $\wedge$
reserved_block(time(S),A,hero, max_action_time + max_action_time).

XD.9 el2_q2($A, S, S0$) $\equiv$
first_opportunity(S, A, hero, $S0$, el2_q2_f(A)).

XD.10 holds(S,loaded_since(B, A, T)) $\equiv$
$\exists_{SA}$ $T \leq$time(SA) $\wedge$ $SA \leq S$ $\wedge$ holds(SA,on_elevator(B)) $\wedge$
holds(SA, elevator_at(A)) $\wedge$ $\neg$engaged(do(A,unload(B)),A, SA).

XD.11 el2_q3($A, S, S0$) $\equiv$
first_opportunity(S, A,hero,S0,loaded_since(b1,A,time($S0$))).

Table 23: Problem statement: Part I

Axioms

X.1 $\text{succeeds}(\text{el1},S1,SZ) \Leftrightarrow \text{holds}(SZ,\text{has}(\text{hero},\text{b1}))$.

X.2 $\text{next\_step}(E,\text{el1},S1,S2) \Leftrightarrow$
$[\text{action}(E,\text{hero}) \wedge$
$[\text{el1\_q1}(S2,S1) \Rightarrow \text{instance}(E,\text{broadcast\_req}(\text{hero},\text{robots},\text{r2}),S2)] \wedge$
$[\text{el1\_q2}(S2,S1) \Rightarrow E{=}\text{do}(\text{hero},\text{call})] \wedge$
$[\text{el1\_q3}(S2,S1) \Rightarrow E{=}\text{do}(\text{hero},\text{unload}(\text{b1}))] \wedge$
$[\neg\text{el1\_q1}(S2,S1) \Rightarrow \neg\exists_{R,U}\ \text{instance}(E,\text{broadcast\_req}(\text{hero},U,R),S2)]]$.

X.3 $\text{actor}(\text{el1}) = \text{hero}$.

X.4 $\forall_A\ A \neq \text{hero} \Rightarrow \text{assignment}(\text{r2},A) = \text{el2}(A)$.

X.5 $A \neq \text{hero} \Rightarrow$
$\text{succeeds}(\text{el2}(A),A,S1,SZ) \Leftrightarrow \text{time}(SZ) \geq \text{time}(S1) + \text{max\_el2b\_time}$.

X.6 $A \neq \text{hero} \Rightarrow$
$\text{next\_step}(E,\text{el2}(A),S1,S2) \Leftrightarrow$
$[\text{action}(E,A) \wedge E \neq \text{do}(A,\text{unload}(\text{b1})) \wedge$
$[\text{el2\_q1}(A,S2,S1) \Rightarrow E{=}\text{do}(A,\text{call})] \wedge$
$[\text{el2\_q2}(A,S2,S1) \Rightarrow E{=}\text{do}(A,\text{load}(\text{b1}))] \wedge$
$[\text{el2\_q3}(A,S2,S1) \Rightarrow$
$\text{instance}(E,\text{inform}(A,\text{robots},\text{loaded\_since}(\text{b1},\text{az},\text{time}(S1))),S2)]]$

X.7 $\text{min\_reserve\_block} \geq$
$\text{max\_action\_time} + \text{max\_action\_time} + \text{max\_elevator\_wait}$
$+ \text{max\_action\_time} + \text{max\_action\_time}$.

X.8 $\text{max\_el2b\_time} \geq \text{delay\_time} + \text{min\_reserve\_block} + \text{delay\_time} + \text{min\_reserve\_block}$.
(An upper bound on the time necessary for el2(A).)

X.9 owns(hero,b1).

X.10 soc_poss(s0).

X.11 reserved(time(s0),hero,hero).

X.12 choice(hero,s0).

X.13 $\forall_{S0A}\ \text{k\_acc}(\text{hero},\text{s0},S0A) \Rightarrow \neg\exists_{P,AC,SX}\ \text{working\_on}(P,AC,AR,SX,S0A)$.

To prove: know_achievable(has(hero,b1),el1,hero,s0).

Table 24: Problem statement : Part II

Appendix B contains the complete, formal proof that this plan is correct; that is, that the hero knows that it is executable and that it will end in his having the package. This appendix is not physically included with the paper; it is available on the Web in PostScript and PDF at

http://cs.nyu.edu/faculty/davise/elevator/commplan-appb.ps and .pdf.

A few features of this proof may be described here.

Overall, the proof begins with proving some general lemmas about time, actions, plans, and so on. It then proves that if all the agents receive request el2(A), then, within a certain time period, the package will be on the elevator and the hero will be informed that the package is on the elevator. It then proves that, if the hero executes plan p.3.1, then, within a certain time period, the hero will have the package. The proof is pretty straightforward, though long. Some unexpected complexity is introduced by the need to take care of cases where things happen fortuitously; for example, if the agent with the package loads it onto the elevator before he receives the request el2(A); if the elevator happens to come to the hero's floor before the package is loaded on it; and so on. In dealing with all these possibilities, the plan starts to resemble a "universal plan" [30].

One of the lemmas in Appendix B is of some general interest. Lemma B.32 establishes that an agent can always follow the protocol, no matter what requests he has received or what commitments he has made. Of course, the reason for this is that the protocol is specifically designed so that, if an agent cannot continue to execute a plan that was requested of him, he is allowed to abandon it. But this lemma establishes that the theory is properly set up so that the agent has an out in all possible cases of conflict between the requests he has received and the constraints of the external world.

8 Paradoxes and comprehension axioms

Our representation of the examples in section 7 works by positing the existence of a variety of fluents to be communicated and of plans to be broadcast, and associating constant symbols with these. A fluent Q is defined in terms of a formula that characterizes the situations where Q holds; a plan is characterized in terms of two formulas, one of which characterizes the "next_step" relation and the other of which characterizes the "succeeds" relation. How can a planner be sure that the fluent or plan so defined can exist? After all, asserting that fluent Q exists is an assumption with substantial logical consequences: If fluent Q exists and agent A knows that Q, then A can inform other agents of Q, and then they will know Q. Thus, positing that Q exists imposes a substantial demand on the time structure. How can a planner be sure that these demands are consistent with the other axioms? What kinds of formulas can be used to define a fluent or a plan?

To answer this question, we will first, in this section, formulate *comprehension axioms,* which characterize the types of formulas that can legitimately be used to define fluents and plans. Then in section 9 we will assert a theorem that asserts that our theory of knowledge and speech acts is consistent with any purely physical theory satisfying certain constraints.

In general, it is preferable for the comprehension axioms to be as inclusive as possible, in order that the range of possible speech acts permitted to the agents should be as broad as possible. Care is needed here, however; the domain of speech acts contains potential paradoxes which can lead to inconsistencies if the comprehension axioms are stated too broadly. In particular, the ability of speech acts to refer to other speech acts is potentially dangerous. Agent $A1$ can tell $A2$ that it is raining; he can tell $A3$ that he has told $A2$ that it is raining; he can tell $A2$ that he will tell $A3$ that he has told $A2$ that it is raining. Worse, he can tell $A2$ at noon that he will tell $A3$ at 1:00 that at noon he told $A2$ something; in that case, the content of his communication at noon refers to itself.

The difficulties here manifest themselves in a number of paradoxes. We discuss three of them below: two that are analogous to Russell's paradox; one related to the conflict of free will with knowledge of the future.[4] After presenting the paradoxes, we will present our formulation of the comprehension axioms and show how by using these axioms and using axiom C.3 to individuate occurrences of "inform" acts, these paradoxes can be defanged, so that they do not lead to inconsistency in the theory.

Paradox 1: Let Q be a fluent. Suppose that over interval $[S0, S1]$, agent a1 carries out the action of informing a2 that Q holds. Necessarily, Q must hold in $S0$, since agents are not allowed to lie (axiom C.1). Let us say that this communication is *immediately obsolete* if Q no longer holds in $S1$. For example, if it is raining in s0, the event of a1 telling a2 that it is raining occurs over [s0,s1], and it has stopped raining in s1, then this communication is immediately obsolete. Now let us say that situation S is "misled" if it is the end of an immediately obsolete communication from a1 to a2. As being misled is a property of a situation, it should be definable as a fluent. Symbolically,

$$\text{holds}(S,\text{misled}) \equiv$$
$$\exists_{Q,S0}\ \text{occurs}(\text{inform}(\text{a1}, \{\text{a1},\text{a2}\},Q),S0,S) \wedge \neg\text{holds}(Q,S)$$

Now, suppose that, as above, in s0 it is raining; from s0 to s1, a1 tells a2 that it is raining; and in s1 it is no longer raining and a1 knows that it is no longer raining. Then a1 knows that "misled" holds in s1. Therefore, (axiom C.1) it is feasible for a1 to tell a2 that "misled" holds in s1. Suppose that, from s1 to s2, the event occurs of a1 informing a2 that "misled" holds. The question is now, does "misled" hold in s2? Well, if it does, then what was communicated over [s1,s2] still holds in s2, so "misled" does not hold; but if it doesn't, then what was communicated no longer holds, so "misled" does hold in s2.

Paradox 2: A similar type of problem arises with the interaction between a comprehension axioms for plans and the actions of committing to plans and of broadcasting requests for plans. Plans in our theory are defined by giving conditions on the "next_step" and "succeeds" relations, so a comprehension axiom for plans will have a form something like the following:

Let $\Gamma(E, S0, S1)$ and let $\Delta(S0, S1)$ be formulas. Then the following is an axiom:

$$\exists_P\ [\forall_{E,S0,S1}\ \text{next\_step}(E, P, S0, S1) \Leftrightarrow \Gamma(E, S0, S1)] \wedge$$
$$[\forall_{S0,S1}\ \text{succeeds}(P, S0, S1) \Leftrightarrow \Delta(S0, S1)]$$

That is, we can use any property expressible in the language to define "next_step" and "succeeds" relations, and these will together define a meaningful plan. (Not necessarily, of course, an executable or correct plan, just a well-defined plan.)

Again, however, this would lead to paradox. Let us say that $\Gamma(E, S0, S1)$ holds if E is the act of committing to a plan P and E is not the next step in $S1$ of P started at $S0$:

$$\Gamma(E, S0, S1) \equiv \exists_P\ \text{instance}(E,\text{commit}(\text{a1},P),S0) \wedge \neg\text{next\_step}(E, P, S0, S1).$$

[4] A fourth paradox — the unexpected hanging paradox — is discussed in [9, 10]. A complete discussion of that paradox in terms of the theory here would be even more involved than the discussion there, but the formal resolution of the paradox that we adopt is essentially the same.

Choose Δ however you want. Then the above axiom would say that there is a plan p1 such that next_step(E,p1,$S0, S1$) $\Leftrightarrow \Gamma(E, S0, S1)$. Now, let action e1 be an instance of commit(a1,p1) in $S0$; is e1 an acceptable next step of p1? Again, we have a paradox: e1 is a next step of p1 if and only if it isn't.

The same paradox, with a couple of additional layers of wrapping, applies to the action of broadcasting a multi-agent plan.

Paradox 3: Let p1 be the following plan for actor a1: "I will clap my hands at 12:01 if and only if a2 informs me by 12:00 that I will not clap my hands at 12:01." Suppose that over interval [s0,s1], a1 carries out the action of committing to p1 and that a2 knows that he has committed to p1. Let s2 be the first choice point for a1 after s1; suppose that time(s2)=11:55 and that all inform acts take 1 minute. Let e1 be the action of a2 informing a1 that he (a1) will not clap his hands at 12:01. The question is, is e1 feasible in s2? On the one hand, if e1 is not feasible then a2 will not carry out e1, and therefore a1, following p1, will not clap his hands at 12:01. But since a2 knows that e1 is not feasible, and he knows that a1 is committed to p1, he also knows that a1 will not clap his hands, so by C.1 action e1 is feasible. On the other hand, if e1 is feasible, then it occurs over some interval [s2,s3], and then on the time-line following s3, a1 following p1 will clap his hands. Thus, a2 does not know that a1 will not clap his hands, and therefore e1 is not feasible in s2.

From the above description, the reader may suspect that the problem is due to applying a linear-time argument to a branching time model; in other words, confounding what may happen with what will happen. Indeed, as we shall see, this suspicion is not far from our ultimate resolution, but this contradiction can be made perfectly tight within the theory as we have set it up, as follows:

Define fluent q1, action e1(SX), and plan p1 by the following formulas:

$$\forall_{S1} \text{ holds}(S1,\text{q1}) \Leftrightarrow$$
$$\neg\exists_{S2,S3}\ S1 < S2 < S3 \wedge \text{occurs}(\text{do(a1,clap)},S2,S3) \wedge \text{time}(S2)\text{=1201} \wedge \text{soc\_poss}(S3).$$

$$\text{instance}(\text{e1}(SX),\text{inform(a2,\{a1,a2\},q1)},SX).$$

$$\forall_{E,S1,S2} \text{ next\_step}(E,\text{p1},S1,S2) \Leftrightarrow$$
$$[\text{E=do(a1,clap)} \Leftrightarrow$$
$$[\text{time}(S2)\text{=1201} \wedge \exists_{SX,SY}\ SY < S2 \wedge \text{time}(SY) \leq 1200 \wedge \text{occurs}(\text{e1}(SX),SX,SY)]].$$

$$\forall_{S0,S1} \text{ succeeds}(\text{p1},S0,S1) \Leftrightarrow \text{time}(S1)\text{=1210}.$$

Posit that occurs(commit(a1,p1),s0,s1), time(s1)=1150, s2 > s1, choice(a2,s2), time(s2)=1155, and that soc_poss(s2). Just to make things simple, posit also that k_acc(a2,s2,$S2A$) $\Leftrightarrow$ $S2A$=s2; i.e. that a2 is omniscient in s2. (The argument will go through without this last assumption, but this makes the argument easier to write.) Then the contradiction goes through as above: If e1 is feasible in s2, then do(a1,clap) will occur in every soc_poss interval following s2, so q1 is false in s2, so e1 is not feasible. But if e1 is not feasible, then it does not occur in any interval following s2, so do(a1,clap) never occurs in any soc_poss interval following s2, so q1 is true in s2, so e1 is feasible.

The paradoxes here, which arise from over-powerful comprehension axioms, are closer to Russell's paradox than to paradoxes that arise from over-powerful syntactic theories, such as the Liar's paradox. Finding a formulation of the comprehension axiom which is strong enough to support the intended application (usually mathematics) but still weak enough to be consistent, is one of the major hurdles in axiomatizing set theory. Usually, the objectives of set theory is to make the comprehension axioms as broad as possible while maintaining the consistency of the theory. Within the scope of this paper, however, our interest is the exact reverse; to find a reasonably simple solution that will be adequate for this class of plans.

Our solution to Paradox 1 uses a device that, as far as we know, is original to us [10]. Our solution to Paradoxes 2 and 3 falls back on a stratified language analogous to Russell's theory of types [27] or Tarski's [32] levels of language used to solve the Liar paradox.[5]

The solution to Paradox 1 begins with the observation that the unique names assumption is subtly hidden in the argument. The argument presumes that if fluent $Q1 \neq Q2$, and the event inform$(A1, \{A1, A2\}, Q1)$ occurs from $S1$ to $S2$ then inform$(A1, \{A1, A2\}, Q2)$ does not occur from $S1$ to $S2$. (Our English description of the argument used the phrase "what was communicated between s1 and s2", which presupposes that there was one unique content that was communicated.) However, axiom C.3 specifically denies this. Therefore, the argument collapses.

In particular, it is a consequence of our theory of time and knowledge that the clock time is always common knowledge among all agents. (See [10] appendix A, Theorem 3). Now, let q1 be any fluent, and suppose that inform(a1,{a1,a2},q1) occurs from s1 to s2. Let t1=time(s1) and let q2 be the fluent defined by the formula

$$\forall_S \text{ holds}(S,\text{q2}) \Leftrightarrow \text{holds}(S,\text{q1}) \wedge \text{time}(S)\text{=t1}.$$

Then it is common knowledge between a1 and a2 that holds(s1,q2) $\Leftrightarrow$ holds(s1,q1). Hence, by axiom C.3, inform(a1,{a1,a2},q2) also occurs from s1 to s2. But by construction q2 does not hold in s1; hence the occurrence of inform(a1,{a1,a2},q2) from s1 to s2 is immediately obsolete. Therefore "misled" holds following *any* informative act.

Changing the definition of misled to use the universal quantifier over Q, thus:

$$\text{holds}(S,\text{misled}) \equiv$$
$$\exists_{S0}\ \forall_Q \text{ occurs}(\text{inform}(\text{a1, \{a1,a2\}},Q),S0,S) \wedge \neg\text{holds}(S,Q)$$

does not rescue the contradiction. One need only change the definition of q2 above to be

$$\forall_S \text{ holds}(S,\text{q2}) \Leftrightarrow \text{holds}(S,\text{q1}) \vee \text{time}(S)\neq\text{t1}.$$

Clearly, the new definition of "misled" *never* holds after any informative act.

We have not found an analogous solution to paradoxes 2 and 3, so we have fallen back on the standard device of stratifying the language. In particular, we divide the sort of "plans" into the two subsorts "simple plans" and "complex plans," and we posit the following restrictions.

- Fluents and simple plans cannot refer to the language of plans, requests, or broadcasts.
- A plan may refer to simple plans but not to complex plans. It is convenient, therefore, to divide the "commit" function into two cases: "commit1(A, P)" takes as argument agent A and a simple plan P, and returns a speech act event whose instances may be part of a complex plan. "commit2(A, P)" takes as argument a complex plan, and returns an speech act events whose instances cannot be of a plan.

[5] It is possible instead to use a Zermelo-Fraenkel-like approach to this comprehension axiom. This requires involve quantifying over entire time-structures. We have not explored the consequences of this approach, or whether it leads to a more expressive language. Our thanks to Walter Dean for suggesting this approach.

This eliminates paradoxes 2 and 3. In paradox 2, the quantified variable P can only range over simple plans, not over plans generally, whereas p1 is a complex plan. Hence the definition

$$\Gamma(E,S0,S1) \;\equiv\; \exists_P \text{ instance}(E,\text{commit1}(A,P),S1) \wedge \neg\text{next\_step}(E,P,S0,S1).$$

would allow an utterance E that is an instance of commit1(a1,p1) as a next step of p1, with no circularity.

The statement of paradox 3 requires that the content of the "inform" act include the predicate "soc_poss($S3$)"; but since this predicate is defined in terms of the execution of plans, it is disallowed in the construction of fluents. This point is worth emphasizing: In this theory the distinction between what *can* happen, physically, and what *will* happen, assuming that agents are cooperative, is defined in terms of the theory of plan execution; it is not a fundamental aspect of the theory of time.

We can now give the formal statement of the comprehension axioms: Let $\mathcal{D}$ be a set of domain-specific symbols, disjoint from the symbols defined in tables 1-16. For instance, in the elevator domain, $\mathcal{D}$ would be { "elevator_at", "on_elevator", "call", "load" .... } In the blocks world, $\mathcal{D}$ might be { "on", "clear", "table", "puton" }.

Definition 1 *The language $\mathcal{L}^1(\mathcal{D})$ is the first-order language containing all the sorts and symbols defined in tables 1-10; the symbols "inform", "communicate", and "reserved"; and the symbols in $\mathcal{D}$.*

Definition 2 *The language $\mathcal{L}^2(\mathcal{D})$ is the language containing $\mathcal{L}^1$ plus the sorts "simple_plan" and "multi-agent plan" and the primitives "assignment", "commit1", "request", "broadcast_req", "next_step" and "succeeds".*

Definition 3 *The language $\mathcal{L}^3(\mathcal{D})$ is the first-order language over all the symbols defined in tables 1-16 union $\mathcal{D}$, plus the symbol "commit2", except that*

- *The sort "simple_plan" is defined as a subsort of "plan". (We don't actually need a sort "complex_plan".)*
- *The argument to "commit1" and the value of "assignment" are each restricted to be a simple plan.*

Axiom 4 (Comprehension axiom schema for fluents) *Let $\Phi(S)$ be a formula in $\mathcal{L}^1(\mathcal{D})$ such that S is a free variable in Φ of sort "situation" and Q does not appear free in Φ. Then the closure of the formula "$\exists_Q \forall_S$ holds$(S,Q) \Leftrightarrow \Phi(S)$" is an axiom.*

Example 4: The formula "holds(S,elevator_at(A)) $\wedge$ holds(S,on_elevator(B))" satisfies the conditions on Φ in axiom 4. Therefore, the sentence

$$\forall_{A,B}\; \exists_Q\; \forall_S \text{ holds}(S,Q) \Leftrightarrow$$
$$\text{holds}(S,\text{elevator\_at}(A)) \wedge \text{holds}(S,\text{on\_elevator}(B))$$

is an axiom. That is, for any agent A and package B, there is a fluent Q corresponding to the condition that the elevator is at A and contains B. Q can then be the content of an "inform" act.

Example 5: The formula "$\exists_{S1,S2}\; S1 < S2 < S \wedge$ occurs(do(A,load(B)),$S1,S2$)" satisfies the conditions on Φ in axiom 4. Therefore, the sentence

$$\begin{array}{l} \forall_{A,B}\ \exists_{QC}\ \forall_{S}\ \text{holds}(S,QC) \Leftrightarrow \\ \exists_{S1,S2}\ S1 < S2 < S \wedge \text{occurs}(\text{do}(A,\text{load}(B)),S1,S2) \end{array}$$

is an axiom. That is, there is a fluent which asserts of situation S that A has loaded B onto the elevator prior to S.

Example 6: The formula "$S1 < S2 < S \wedge$ occurs(do(A,load(B)),$S1, S2$)" violates the conditions of axiom 4, as it contains free variables $S1$ and $S2$ of sort "situation". Therefore axiom 4 does not apply. If this exclusion were not made, then an inform act could refer *de re* to two particular situations $S1, S2$ in the same way that it refers *de re* to agent A and package B. It is not at all clear what such a *de re* reference would mean.

Example 7: Let $\Phi(S)$ be the formula

$$\begin{array}{l} \exists_{Q}\ \forall_{S2}\ [\exists_{E}\ \text{instance}(E,\text{inform}(\text{a2},\{\text{a1},\text{a2}\},Q),S2) \wedge \text{occurs}(E,S,S2)] \Rightarrow \\ \qquad \forall_{S2A}\ \text{k\_acc}(\text{a1},S2,S2A) \Rightarrow \\ \qquad\qquad [\text{holds}(S2,\text{on\_elevator}(\text{b1})) \Leftrightarrow \text{holds}(S2A,\text{on\_elevator}(\text{b1}))] \end{array}$$

That is, Φ holds on S if there is something that agent a2 can tell a1 in S which will cause a1 to know whether or not package b1 is on the elevator. Note that this fluent both quantifies over fluents and refers to future informative events. Further, more natural, examples of this kind are presented in [9].) Φ satisfies the conditions of axiom 4. Therefore there is a fluent q1 such that holds(S,q1) $\Leftrightarrow \Phi(S)$ and q1 can be the content of an inform act. That is, for example, a3 can inform a2 that there is something that agent a2 can tell a1 in S which will cause a1 to know whether or not package b1 is on the elevator.

Axiom 5 (Comprehension axiom schema for simple plans) *Let* $\Gamma(E, S0, S1)$ *and* $\Delta(S0, S1)$ *be formulas in* $\mathcal{L}^1(D)$ *such that*

1) *E is a free variable of sort "action" and* $S1$ *and* $S2$ *are free variables of sort "situation".*

2) *The variable* PS *does not appear free in* Γ *or* Δ.

Then the closure of the formula

$$\begin{array}{l} \forall_{A}\ \exists_{PS}\ [\forall_{E,S0,S1}\ \textit{next\_step}(E,PS,S0,S1) \Leftrightarrow \textit{action}(E,A) \wedge \Gamma(E,S0,S1)] \wedge \\ \quad [\forall_{S0,S1}\ \textit{succeeds}(PS,S0,S1) \Leftrightarrow \Delta(S0,S1)] \end{array}$$

is an axiom. The variable PS *above has sort "simple_plan".*

That is, you can use any formulas in $\mathcal{L}^1(\mathcal{D})$ of the appropriate sort to define the next step and the success conditions for a simple plan for actor A.. Note that the subformulas Γ and Δ are in language $\mathcal{L}^1(\mathcal{D})$; the axiom as a whole is in language $\mathcal{L}^2(\mathcal{D})$.

Example 8: The plan p1 defined on page 25 exists and is a simple plan. Define $\Gamma(E, S0, S1)$ and $\Delta(S0, S1)$ as

$$\begin{array}{l} \Gamma(E,S0,S1) \equiv \\ [[S1 = S0 \Rightarrow E{=}\text{do}(\text{joe},\text{put\_on}(\text{a,b}))] \wedge \\ [\text{occurs}(\text{do}(\text{joe},\text{put\_on}(\text{a,b}))S0,S1) \Rightarrow E{=}\text{do}(\text{joe},\text{put\_on}(\text{c,d}))]]. \end{array}$$

$$\begin{array}{l} \Delta(S0,S1) \equiv \\ \text{holds}(S1,\text{on}(\text{a,b})) \wedge \text{holds}(S1,\text{on}(\text{c,d})). \end{array}$$

Clearly, Γ and Δ are both in the language $\mathcal{L}^1(\mathcal{D})$.

Then axiom 5 asserts that there exists a simple plan PS such that next_step$(E, PS, S0, S1) \Leftrightarrow$ action(E,joe) $\wedge\, \Gamma(E, S0, S1)$ and succeeds$(PS, S0, S1) \Leftrightarrow \Delta(S0, S1)$.
Example 9: It is a consequence of axiom 5 that, for every AZ, the plan "el2(AZ)" defined in tables 23 and 24 is a simple plan. The definitions introduced in table 23 can all be expanded out into formulas in $\mathcal{L}^1$. The constant symbol "max_el2b_time" can be replaced by an existentially quantified variable M satisfying axiom X.8. Thus axiom 5 can be applied to assert that, for every agent AZ and duration M there exists a plan whose "next_step" and "succeeds" relations satisfy the appropriate conditions, stated in formulas Γ and Δ expressed in $\mathcal{L}^1(\mathcal{D})$.

Axiom 6 (Comprehension axiom schema for multi-agent plans) *Let $\Psi(P, A)$ be a formula in $\mathcal{L}^2(\mathcal{D})$ such that*

1) P is a variable of sort "simple plan" and A is a variable of sort "agent".

2) The variable R does not appear free in Ψ.

Then the closure of the formula

$$[\forall_A \exists_P\ \Psi(P, A)] \Rightarrow \exists_R\ \forall_A\ \Psi(assignment(R, A),A).$$

is an axiom. That is, if there exists a plan P satisfying $\Psi(P, A)$ for every agent A, then there is a multi-agent plan R that assigns to A a plan P satisfying $\Psi(P, A)$.

Axiom 7 (Comprehension axiom schema for complex plans) *Let $\Gamma(E, S0, S1)$ and $\Delta(S0, S1)$ be formulas in $\mathcal{L}^3(\mathcal{D})$ such that*

1) E is a free variable of sort "action" and $S1$ and $S2$ are free variables of sort "situation".

2) The variable P does not appear free in Γ or Δ in $\mathcal{D}$.

Then the closure of the formula

$$\forall_A\ \exists_P\ \forall_{E,S0,S1}\ [next\_step(E, P, S0, S1) \Leftrightarrow action(E, A) \wedge \Gamma(E, S0, S1)] \wedge \\ [succeeds(P, S0, S1) \Leftrightarrow \Delta(S0, S1)].$$

is an axiom. The variable P is of sort "plan".

That is, you can use any formulas in $\mathcal{L}^2(\mathcal{D})$ of the correct sort to define the next step and the success conditions for a plan.
Example 10: Let p1 be the plan that a1 will broadcast a request that all the agents in set u2 should clap their hands at the first opportunity. Define Γ_2 and Δ_2 be the following formulas in $\mathcal{L}^1(\{\text{"clap"}\})$:

$\Gamma_2(E, S0, S1, AZ) \equiv$ first_opportunity($S1, AZ$,a1,$S0$,true) $\Rightarrow E$=do(AZ,clap).
$\Delta_2(S0, S1, AZ) \equiv \exists_{S2}$ occurs(do(AZ,clap),$S2, S1$).

Let $\Psi(P, AZ)$ be the following formula in $\mathcal{L}^2$:

$\Psi(P, AZ) \equiv$
$\forall_{E,S0,S1}$ [next_step$(E, P, S0, S1) \Leftrightarrow \Gamma_2(E, S0, S1, AZ)] \wedge$
[succeeds$(P, S0, S1) \Leftrightarrow \Delta_2(S0, S1, SZ)$]

Define Γ_1 and Δ_1 to be the following formulas in $\mathcal{L}^3$:

$$\begin{aligned}
&\Gamma_1(E,S0,S1) \equiv \\
&S0 = S1 \Rightarrow \exists_R\ [\forall_A \text{ assignment}(R,A){=}P \Leftrightarrow \Psi(P,A)] \wedge \\
&\qquad\qquad \text{instance}(E,\text{broadcast\_req(a1,u2,}R),S1).
\end{aligned}$$

$$\begin{aligned}
&\Delta_1(S0,S1,AZ) \equiv \\
&\exists_R\ [\forall_A \text{ assignment}(R,A){=}P \Leftrightarrow \Psi(P,A)] \wedge \text{occurs(broadcast\_req(a1,u2,}R),S0,S1)
\end{aligned}$$

Then axiom 7 asserts

$$\begin{aligned}
&\exists_P\ \forall_{E,S0,S1}\ [\text{next\_step}(E,P,S0,S1) \Leftrightarrow \Gamma_1(E,S0,S1)] \wedge \\
&\qquad\qquad [\text{succeeds}(P,S0,S1) \Leftrightarrow \Delta_1(S0,S1)].
\end{aligned}$$

Here Γ_2 and Δ_2 characterize the plans requested from the other agents; Ψ characterized the content of the broadcast; and Γ_1 and Δ_1 characterize a1's plan of broadcasting the request.

Example 11: Expanding out the definitions XD.1–XD.13 in table 23, axiom 5 states that there do exist simple plans satisfying axioms X.5 and X.6. Axiom 6 states that there exists an assignment satisfying axioms X.4. Axiom 7 asserts that there exists a plan satisfying axioms X.1 and X.2.

It is certainly possible to find quite natural examples of plans that fall outside the scope of these comprehension axioms, or their natural extension to additional levels of stratification. For example, suppose that each agent can only communicate directly with the agent immediately above or below him. Then an agent who wishes to get a package on an unknown floor will ask the other agents to keep passing the request upward until it reaches the agent who has the package. This involves an imbedding of requests of unbounded depth, and so cannot be handled by theory stratification such as we have described. Indeed, it is not even clear how one would extend the representation language to describe such a plan. Another problem for another day.

9 Consistency

We have shown that, with the above comprehension axioms, our theory can side-step these three paradoxes. How do we know that the next paradox won't uncover an actual inconsistency in the theory? We can eliminate all worry about paradoxes once and for all by proving that the theory is consistent; in fact, that it is consistent with a very broad class of physical theories and problem specifications. The previous paper [10] presents, discusses, and proves a consistency theorem for our theory of knowledge and informative acts. It turns out that it is straightforward to extend this proof to a proof of the consistency of the theory in this paper, which includes multi-agent planning as well. Therefore, in this section we will give the minimum discussion needed to correctly state this consistency result. Appendix A gives a brief sketch of how the proof given in [10] can be modified for the new theorem.

Theorem 12 below states that the axioms in this theory are consistent with essentially any physical theory that has a model over discrete time with a starting point state and physical actions.

Definition 8 *A* physical language *is a first-order language containing the sorts "situations", "agents", "physical actionals", "physical actions", "physical fluents", "clock times"* *and "u-intervals"; containing all the non-logical symbols introduced in tables 1-8 and table 12; and excluding all symbols introduced in tables 9-11 and 13-18.*

Definition 9 *Let $\mathcal{L}$ be a physical language, let $\mathcal{T}$ be a theory over $\mathcal{L}$. $\mathcal{T}$ is an* acceptable physical theory *(i.e. acceptable for use in theorem 12 below) if there exists a model $\mathcal{M}$ and an interpretation $\mathcal{I}$ of $\mathcal{L}$ over $\mathcal{M}$ such that the following conditions are satisfied:*

1. *$\mathcal{I}$ maps the sort of durations to the integers, the duration constant 0 to integer zero, the relation $D1 < D2$ on clock times to the usual ordering on integers, and the function $D1 + D2$ on clock times to the usual addition on integers,*
2. *The axioms of time, events, and actions, TD.1, TD.2, T.1 — T.18, EVD.1, EVD.2, EV.1, AD.1 — AD.3, A.1 — A.7, and U.1 — U.3 hold in $\mathcal{M}$ under $\mathcal{I}$.*
3. *Theory $\mathcal{T}$ is true in $\mathcal{M}$ under $\mathcal{I}$.*
4. *The theory is consistent with the following constraint: In any situation S, if any communication act is feasible, then infinitely many physically indistinguishable communication acts are feasible.*
5. *If α is a predicate symbol in $\mathcal{L}$ with more than one situational argument, then $\alpha(X_1 \ldots X_k)$ holds only if all the situations among $X_1 \ldots X_k$ are ordered with respect to $<$. (Note that this condition holds both when α is "$<$" and α is "occurs".) If $\beta(X_1 \ldots X_k)$ is a function symbol, then the above condition holds for the relation $X_{k+1} = \beta(X_1 \ldots X_k)$.*
6. *There are finitely many agents.*

Condition (4) no doubt seems complex, strange, and restrictive. But in fact any physical model can be easily transformed into one satisfying this condition: take the original model and, wherever a communicative act occurs, make an infinite number of identical copies of the subtree following the branch where the act occurs. (The exact definition of "physically indistinguishable" is given in [10].) Moreover, most reasonable physical theories $\mathcal{T}$ will accept this transformation, or can be straightforwardly transformed into theories that will accept this transformation. In fact, therefore, condition (4) is not a substantial restriction on $\mathcal{T}$. The reason it is needed is that, without this condition, the physical theory could include an axiom like, "In any situation S there is only one situation $S1$ such that occurs(communicate(AS, U),$S, S1$)" whereas our theory demands that there must exist many such situations corresponding to the different informative acts and requests possible in S.

Condition (5) is a technical one needed for the proof. We do not know of any causal theories that contain predicates that do not satisfy this condition and cannot be defined in terms of simpler predicates that satisfy this condition.

Definition 10 *Let $\mathcal{L}$ be a physical language. The* type-1 social language over $\mathcal{L}$ *is equal to $\mathcal{L}$ together with the symbols, "govern", "reserved", "reserved_block", "min_reserve_block" and "delay_time".*

Definition 11 *Let $\mathcal{L}$ be a physical language, and let $\mathcal{T}$ be an acceptable physical theory over $\mathcal{L}$. Let $\mathcal{M}$ be a model and let $\mathcal{I}$ be an interpretation satisfying the conditions of definition 9. Let $\mathcal{L}'$ be the type-1 social language over $\mathcal{L}$ and let $\mathcal{I}'$ be an extension of $\mathcal{I}$ that provides an* *interpretation for the additional symbols such that axioms QD.1 and Q.1 — Q.4 are satisfied. A theory $\mathcal{T}'$ over $\mathcal{L}'$ that extends $\mathcal{T}$ and that is true under $\mathcal{I}'$ is called an* acceptable type-1 social theory.

It is trivial to show that any acceptable physical theory can be extended to an acceptable type-1 social theory.

Theorem 12 *Let $\mathcal{L}$ be a type-1 social language, and Let $\mathcal{T}$ be an acceptable type-1 social theory over $\mathcal{L}$. Let $\mathcal{L}'$ be equal to $\mathcal{L}$ together with all the general non-logical symbols in this paper (i.e. all those not specific to particular domains and problems). Let $\mathcal{U}$ be $\mathcal{T}$ together with all the general axioms in this paper. Then $\mathcal{U}$ is consistent.*

In order to verify the consistency of our theory of the elevator domain and our problem specification, it is necessary to strengthen theorem 12 by adding in domain-specific axioms of knowledge acquisition, plus conditions on the initial knowledge and ignorance of the agents.

Specifically, we define a knowledge acquisition axiom as follows: (see [10] for more extensive discussion.)

Definition 13 *A* knowledge acquisition axiom *has the form*

$$\begin{array}{l}\forall_{A,S}\ [[\forall_{SA}\ k\_acc(A,S,SA) \Rightarrow \Phi_i(A,S)] \lor \\ \qquad [\forall_{SA}\ k\_acc(A,S,SA) \Rightarrow \neg\Phi_i(A,S)]]\end{array}$$

where $\Phi(A,S)$ satisfies the following conditions:

- *The only free variables in $\Phi(A,S)$ are A and S.*
- *If $S1$ is a quantified variable other than S appearing in Φ, and $S1$ is used as either the second-to-last or last argument for either k_acc or ck_acc, then the quantification of $S1$ imposes the restriction $S1 < S$.*
- *If $S1$ is a quantified variable other than S appearing in Φ, and $S1$ is not used as an argument for either k_acc or ck_acc, then the quantification of $S1$ imposes the restriction $S1 \leq S$.*

These last two conditions mean that a knowledge acquisition axiom may specify that an agent may gain knowledge about the physical state of the world in the past and present but not (except by inference) in the future, and may gain knowledge about the knowledge states of other agents in the past but not in the present or future. Axioms E.19, E.20, and E.21 are examples of such axioms: the robot knows whether the elevator is on his floor, whether he has package B, and whether [the elevator is on his floor and package B is on the elevator].

Theorem 14 *Let $\mathcal{T}$ be an acceptable type-1 social theory, and let $\mathcal{U}$ be the union of:*

A. $\mathcal{T}$;

B. All the general axioms in this paper.

C. A collection of domain-specific knowledge acquisition axioms.

D. Any set of axioms $\mathcal{K}$ specifying the presence or absence of k_acc relations among situations at time 0 as long as:

i. The axioms in $\mathcal{K}$ do not refer to any situations of time later than 0.

ii. The axioms in $\mathcal{K}$ are consistent with $\mathcal{T}$, axioms K.1 — K.3, K.5 (as regards knowing the feasibility of actions at time 0); and the axioms in (C).

Then $\mathcal{U}$ is consistent.

The proof of theorems 12 and 14 is sketched in appendix A
http://cs.nyu.edu/faculty/davise/elevator/commplan-appa.ps and .pdf.

It is straightforward to verify

- That the physical elevator axioms E.1 — E.18 and X.9 satisfy the conditions of an acceptable physical theory;
- That adding axioms X.7, X.8, and E.22 to the above satisfy the conditions of an acceptable type-1 social theory;
- That axioms E.19, E.20, and E.21 satisfy the conditions in definition 4 for knowledge acquisition axioms.
- Using lemma B.38 of appendix B, one can establish that there exists a socially possible situation s0 satisfying conditions X.10 — X.13.
- As discussed above, it follows from axioms F.1 — F.4 that a plan exists satisfying the conditions on el1 in axioms XD.1 — XD.11, X.1 — X.6.

Therefore, the axiomatization of the elevator world and of the problem statement are consistent with the remaining axioms in this paper. Hence the proof of the correctness of plan el1 in appendix B is non-vacuous. (If this theory were not consistent, then of course *any* statement can be proven from it, so the fact we can prove that el1 is correct would not be very impressive.)

10 Related Work

The foundations of this work come from Moore [19], which makes the key proposals that:

1. Temporal situations [16] could be identified with epistemic possible worlds [13].
2. A first-order language, in which situations were first-class entities, could be used to represent statements about knowledge and time. Reasoning about knowledge and time could be modelled as deduction in this language.
3. Agent A *knows how to do* action E in situation S if, in S, A knows a standard identifier for E.

Moore further proposed that the "knowledge preconditions" problem for plans, (discussed briefly in [16]) could be solved by using recursive rules over the form of the plan, where rule (3) above is the base of the recursion. For example, agent A knows in S how to do the plan "if (Q) then do $P1$ else do $P2$" if either [A knows in S that Q is true and knows in S how to do $P1$] or [A knows in S that Q is false and knows in S how to do $P2$].

This analysis of knowledge preconditions for plans is extended and improved in [8] where it is shown that

1. Moore's recursive rules constitute sufficient, but not necessary conditions.
2. A general rule that covers all cases is the following: Agent A has enough knowledge to execute plan P starting in situation S if and only if the following holds: If A has begun the execution of P from S to $S1$, then in $S1$, A will know whether he has completed the execution of P and, if he has not completed P, then he will know how to execute some next step of P.
3. Moore's recursive rules can be proven as theorems from the general rule (2).

The theory of planning put forward in this paper is a direct extension of the theory of [8] to the case of multi-agent plans.

Morgenstern [20, 21] recasts Moore's theory in a "syntactic" theory of knowledge, in which the content of knowledge is viewed as a string of characters. This move was motivated by the desire to avoid using a modal logic of knowledge with possible worlds semantics, and its attendant and sometimes undesirable consequences, such as all agents knowing all axioms. To avoid Liar-style paradoxes, it uses a three-valued logic. This work also extends Moore's theory to deal with general multi-agent plans. It examines specific actions such as agent A1 informing A2, A1 querying A2 (i.e., requesting that A2 inform him about some particular fact), and A1 commanding A2 to perform a particular action. The theory of delegation is substantially less restrictive than the theory developed in this paper. For example, delegation can be done micromanager-style or executive-style: A1 can ask A2 to execute a highly specific plan, or just ask that A2 achieve some goal in whichever way A2 prefers (see section 11). In addition, the notion of plan execution is not tied to specific protocols of agent interaction.

Steel [31] recasts Moore's theory using a combination of dynamic and epistemic modal logic.

The literature on theories of time and knowledge is immense (e.g. [29, 12]). In [10] we give a detailed analysis of the relation between some of the particularly relevant theories in this area and our theory of time and knowledge. Wooldridge [34] chap. 12 includes a short but very useful survey of the applications of these theories to multi-agent systems, and includes an extensive bibliography.

There is also a large literature on "BDI" (belief, desire, intention) models, ranging from logical analyses, which are relevant here, to implementations, which mostly are not. The general BDI model was first proposed by Cohen and Perrault [4]; within that model, they formalized illocutionary acts such as "Request" and "Inform" and perlocutionary acts such as "Convince" using a STRIPS-like representation of preconditions and effects on the mental states of the speaker and hearer. Cohen and Levesque [3] extend and generalize this work using a full modal logic of time and propositional attitudes. Here, speech acts are *defined* in terms of their effects; a request, for example, is any sequence of actions that achieves the specified effect in the mental state of the hearer. Rao [25], despite the title, is more nearly a Prolog implementation of BDI than a logical theory; in particular, his theory does not have an explicit representation of time. A major problem in developing a logic of planning based on BDI is that it is extremely difficult to formulate universally valid rules that relate the desires or goals of an agent to its actions.

Update logic (e.g. [23, 33]) combines dynamic logic with epistemic logic, introducing the dynamic operator $[A!]\phi$, meaning "ϕ holds after A has been truthfully announced.". The properties of this logic have been extensively studied. Baltag, Moss, and Solecki [2] extend this logic to allow communication to a subset of agents, and to allow "suspicious" agents. Colombetti [5] proposes a *timeless* modal language of communication, to deal with the interaction of intention and knowledge in communication. Parikh and Ramanujam [22] present a theory of *messages* in which the meaning of a message is interpreted relative to a protocol.

A number of researchers have applied modal logics of knowledge to the analysis and implementation of multi-agent systems. For example, Sadek et al. [28] present a first-order theory with two modal operators $\mathrm{B}_i(\phi)$ and $\mathrm{I}_i(\phi)$ meaning "Agent i believes that ϕ" and "Agent i intends that ϕ" respectively. An inference engine has been developed for this theory, and there is an application to automated telephone dialogue that uses the inference engine to choose appropriate responses to requests for information. However, the temporal language associated with this theory is both limited and awkward; it seems unlikely that the theory could be applied to problems involving multi-step planning. (The dialogue application requires only an immediate response to a query.)

The multi-agent communication languages KQML [11] and FIPA [1] provide rich sets of communication "performatives". KQML was never tightly defined [34]. FIPA has a formal semantics defined in terms of the theory of Sadek et al. [28] discussed above. However, the content of messages is unconstrained; thus, the semantics of the representation is not inherently connected with the semantics of the content, as in our theory.

Other modal theories of communication, mostly propositional rather than first-order, are discussed in [35, 15, 24].

11 Conclusions

The major accomplishments of this paper are:

1. The construction of a first-order language sufficient to characterize a multi-agent domain, integrating action, knowledge, plans, and communication.

2. The validation of a simple plan in a simple specific physical domain. It is quite clear that the general language of knowledge, planning, and communication can be applied to a very broad range of physical domains.

3. The identification of potential paradoxes in the theory analogous to Russell's paradox and the formulation of comprehension axioms that avoid the paradoxes but still allow a very expressive language of planning and communication. Some aspects of these comprehension axioms are standard; others are new and interesting.

4. The proof that the theory is consistent.

The theory has many significant limitations that we would like to overcome, and that we plan to address in future work:

1. The rather restrictive protocol for agent interaction is "hard-wired" into the definition of executing a plan. Presumably, some protocol is necessary if we wish to *prove* with certainty that the plan will be correctly executed, in this kind of general setting. However, it would be better if it were possible to separate out the specifics of the protocol from the general definition of correct execution. We do not see how to do this.

2. There are important classes of communicative plans that do not come within the scope of our comprehension axioms; for example, passing a message along until reaching the intended recipient.

3. The theory would be much more powerful if it were extended to allow plausible reasoning.

4. The theory here deals only with a rather restrictive and, indeed, rather micro-managing form of delegation: One agent can ask another to carry out a specified plan. A more general notion of delegation would allow one agent to request another to achieve a specified goal. For example, Morgenstern [21] discusses the following example:

   > Alice knows that Bob is able, one way or another, to find out Charley's telephone number. Alice can therefore ask Bob for Charley's telephone number and be sure that he will find it out and tell it to her.

   The theories in [21] or in [3] are sufficient to handle these kinds of problem. The theory here cannot handle this, because Alice is not specifying a plan for Bob to carry out. One might think that she could do this by making the request "Do whatever is necessary, then tell me Charley's telephone number". However, this is not a "plan" within the scope of our definition, because at no stage does it specify a next step to be carried out.

5. The theory of planning here, as presented in section 6 is largely a matter of formulating a definition of "knowing enough to execute a plan." The definition that we have constructed, in tables 17 and 18 are much too complicated and too dependent on the specifics of the protocol to command anything like immediate assent or support any claim of "self-evident truth". It would be well to put this on firmer ground. One promising possibility might be to show the definition of "executable plan" is in fact satisfied by some specific class of implementable agents, along the lines of the framework defined by Fagin et al. [12].

The paper is somewhat unusual methodologically in that it starts with examples that are comparatively natural and that, though simple, draw on many different aspects of multi-agent interactions. These examples drive the analysis of the concepts. By contrast many of the theoretical papers in this area start either with no specific examples, or with artificial and implausible puzzles, such as the "Muddy Children" problem. These papers start out with the concepts the authors wish to analyze, then design problems to specifically highlight these concepts; the concepts drive the examples. The hope is that our approach will lead to more natural and central problems and to raise earlier and more centrally the problem of integrating many different domains for reasoning. All example-driven analysis is, of course, open to the danger that the analysis of each individual domain will be more superficial, and may omit considerations that are in fact important but do not happen to arise in the particular example being studied.

References

[1] FIPA 2001. The foundation for intelligent physical agents.

[2] A. Baltag, L. Moss, and S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Tech. Report 534, Computer Science Dept., U. Indiana, 2000.

[3] P.R. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[4] P.R. Cohen and C.R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979.

[5] M. Colombetti. A modal logic of intentional communication. *Mathematical Social Sciences*, 38:171–196, 1999.

[6] E. Davis. *Representations of Commonsense Knowledge.* Morgan Kaufmann, San Mateo, CA, 1990.

[7] E. Davis. Branching continuous time and the semantics of continuous action. In *AIPS-94*, pages 1–100, 1994.

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

[9] E. Davis. A first-order theory of communicating first-order formulas. In *Ninth International Conference on Principles of Knowledge Representation and Reasoning*, pages 235–245, 2004.

[10] E. Davis. Knowledge and communication: A first-order theory. *Artificial Intelligence*, to appear.

[11] T. Finin et al. Specification of the KQML agent communication language, 1993.

[12] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge.* MIT Press, 1995.

[13] J. Hintikka. Semantics for propositional attitudes. In L. Linsky, editor, *Reference and Modality*, pages 145–167. Oxford University Press, 1969.

[14] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94:79–97, 1997.

[15] A. Lomuscio and M. Ryan. A spectrum of modes of knowledge sharing between agents. In N. Jennings and Y. Lesperance, editors, *Intelligent Agents VI: Agent Theories, Architectures, and Languages*, Lecture Notes in Artificial Intelligence 1757, pages 13–26. Springer-Verlag, 2000.

[16] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, Edinburgh, 1969.

[17] D.V. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[18] R. Moore. Reasoning about knowledge and action. Note 191, SRI International, Menlo Park, CA, 1980.

[19] R. Moore. A formal theory of knowledge and action. In Jerry Hobbs and Robert Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. ABLEX Publishing, Norwood, New Jersey, 1985.

[20] L. Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 867–874, 1987.

[21] L. Morgenstern. *Foundations of a Logic of Knowledge, Action, and Communication.* PhD thesis, New York University, 1988.

[22] R. Parikh and R. Ramanujam. A knowledge-based semantics of messages. *Journal of Logic, Language, and Information*, 12(4):454–467, 2003.

[23] J. Plaza. Logics of public announcements. In *Proceeding of the 4th International Symposium on Methodologies for Intelligence Systems*, 1989.

[24] A.S. Rao. Decision procedures for propositional linear time belief-desire-intention logics. In Michael Woodridge, Jorg P. Muller, and Milind Tambe, editors, *Intelligent Agents II: Agent Theories, Architectures, and Languages*, Lecture Notes in Artificial Intelligence 1037, pages 33–48. Springer-Verlag, 1995.

[25] A.S. Rao. Agentspeak(l): BDI agents speak out in a logical computable language. In *Agents Breaking Away: 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Lecture Notes in Artificial Intelligence 1038, pages 42–55. Springer-Verlag, 1996.

[26] R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.* MIT Press, 2001.

[27] B. Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30:222–262, 1908.

[28] M.D. Sadek, P. Bretier, and F. Panaget. Artimis: Natural dialogue meets rational agency. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 1030–1035, 1997.

[29] R.B. Scherl and H.J. Levesque. Knowledge, action, and the frame problem. *Artificial Intelligence*, 144(1):1–39, 2003.

[30] M.J. Schoppers. Universal plans for reactive robots in unpredictable environments. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1039–1046, 1987.

[31] S. Steel. Action under uncertainty. *Journal of Logic and Computation*, 4(5):767–795, 1994.

[32] A. Tarski. The concept of truth in formalized languages. In *Logic, Science, and Metamathematics.* Oxford University Press, 1956.

[33] J. van Benthem. ‘one is a lonely number’: on the logic of communication. ILLC Tech Report 2003-07, Institute for Logic, Language and Computation, University of Amsterdam, 2003.

[34] M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.

[35] M. Wooldridge and A. Lomuscio. Reasoning about visibility, perception, and knowledge. In N. Jennings and Y. Lesperance, editors, *Intelligent Agents VI: Agent Theories, Architectures, and Languages*, Lecture Notes in Artificial Intelligence 1757, pages 1–12. Springer-Verlag, 2000.