# Command Dialogues

Katie Atkinson[1], Rod Girle[2], Peter McBurney[1], and Simon Parsons[3]

[1] Department of Computer Science, University of Liverpool, UK
{K.M.Atkinson,mcburney}@liverpool.ac.uk
[2] Department of Philosophy, University of Auckland, New Zealand
r.girle@auckland.ac.nz
[3] Department of Computer and Information Science, Brooklyn College, New York, USA
parsons@sci.brooklyn.cuny.edu

**Abstract.** We propose a representation of imperatives in computational systems, and a multi-agent dialogue protocol to argue over these. Our representation treats a command as a presumptive argument for an action to be executed by a designated agent, together with a set of associated critical questions whose answers may defeat the presumption. The critical questions enable the identification of attacks on the uttered command, and so can be used to specify a dialogue game protocol for participants to argue over the command. We present a formal syntax for part of the protocol, called **CDP**, and outline denotational semantics for both commands and for the protocol.

**Keywords:** agent communications, argument schemes, commands, dialogue games, imperatives, interaction protocols, semantics

## 1 Introduction

Computational processes may malfunction or they may interfere with the successful operation of other processes, whether intentionally or not. If a process in a single, centralized computer system malfunctions, the thread that is in overall control of the system can de-activate or delete the malfunctioning process. In a computer system with multiple threads of control, such as a distributed e-commerce system, there may be no central thread with the power to de-activate or delete malfunctioning processes on remote machines; at best, a co-ordinating thread may instruct the relevant remote thread controlling a malfunctioning process to de-activate or delete it. If the remote entities in the distributed system are autonomous, such an instruction may or may not be obeyed. If these remote entities are intelligent, they may also question or seek justification for an instruction prior to deciding whether or not to obey it. Humans do this constantly, even in organizations with strong hierarchical command structures, such as the military.

How should such instructions be formally represented and issued? And once issued, how should they be questioned or challenged, and how justified? Girle [7] presented the syntax of a dialogue protocol, **IDL3**, which enables arguments over commands. The protocol was based on a Hamblin-style dialogue game, **DL3** (also defined in [7]), with additional locutions for issuing and responding to instructions. If challenged, an instruction may be justified by reference to some proposition (indicating some fact about the world), and/or by reference to some further action which the commanded action is intended to enable. For many applications, such as the computer-aided instruction systems to which dialogue games were first applied [4, 14], this level of granularity of justification may be sufficient, especially if participants share over-arching goals. However, multi-agent computational applications typically involve participants with possibly-conflicting beliefs, goals, and even values.[4] We therefore believe a more finely-grained approach is required, in order that a dialogue over commands is better able to elucidate the differences between participating agents. Moreover, [7] gives no semantics for **IDL3**, although an axiomatic semantics defined in terms of the pre- and post-conditions of dialogue utterances would not be difficult to define.

What makes command utterances different to other locutions regarding actions, such as proposals, promises, requests or entreaties? Firstly are the pre-conditions of the utterance: the valid utterance of a command pre-supposes the existence of a regulatory environment or a social context in which one party (the speaker) has some right to require another party (the hearer) to perform some action at the behest of the first party, something Habermas has called the *normative rightness* of a directive [8]. Such an environment or context, if it exists, creates a *prima facie* obligation on the hearer to obey any legally-issued commands by the speaker. Of course, whether such a context actually pertains between the parties may be a matter of disagreement between them, and thus itself open to question and challenge. A protocol for multi-party dialogues over commands should be sufficiently expressive to permit such disputation. Secondly are the conditions of revocation of the utterance: normally only the issuer of a command utterance has the contextual power to retract the utterance and thus to revoke the requirement that the action stated be performed by the recipient of the command. This dialogic aspect of the semantics of a command — that revocation privileges are limited only to the initial speaker, even after a command may be accepted by a recipient — distinguishes commands from many other types of action locutions, such as promises, proposals and entreaties, as two of us explain in greater detail in [13].

Thirdly, are the possible forms of challenge to a command utterance: questioning or challenging a command may involve disagreement not only with the appropriateness, suitability or feasibility of a proposed action, as in the protocol for arguments over action of [2], but also disagreement with the normative rightness of the command utterance. As Habermas argues, *"refusing imperatives normally means rejecting a claim to power"* [8, p. 325]. Thus dialogue over commands may involve discussions over the right of the agent issuing a command to do so, or to do so at the particular time of the utterance, or to do so to the intended recipient. Any protocol for command dialogs needs, therefore, to enable arguments over such issues.

The approach adopted in this paper begins by presenting a novel formalism for representing commands, similar to the computational representation of proposals for actions given in [2]. That representation construed an action proposal as an argument scheme for practical reasoning, i.e., as a presumption for action along with a collection of relevant critical questions whose answers may defeat the presumption, following the

[4] Values, as used in this sense, represent the social interests promoted through achieving the goal. Thus they are qualitative measures of the desirability or non-desirability of achieving a goal.

account of Walton [19]. The current paper extends that representation to deal with command utterances, and identifies the specific critical questions appropriate for response to such commands. Section 2 presents our argument schemes for commands, and the associated critical questions are articulated in Section 3. These argument schemes enable the specification of a dialogue protocol, called **CDP**, in which commands can be issued, rationally questioned, challenged and justified, and accepted or rejected. The syntax for **CDP** is presented in Section 4, and we outline denotational semantics for commands and for the protocol in Section 5. Our semantic framework draws on recent work by Reed and Norman [18] formalizing Hamblin's Action-State Semantics for imperatives [9] and by two of us in the semantics of action dialogues [12, 13]. The paper concludes with a brief summary of our contribution and a discussion of future work.

## 2 Representing Commands

We first propose a representation of commands. We assume two dialogue participants, called *Commander* and *Receiver*, respectively. After opening, a dialogue between the two begins with Commander issuing an instruction or command to Receiver to execute some action. In the spirit of Hamblin [9], we allow this instruction to either specify the action to be executed or to specify a (partial or complete) world state to be achieved as a result of successful execution of an action, or both. For simplicity, we assume that actions either successfully achieve their intended goals, or they do not, and, apart from this binary outcome, we do not represent any degree of uncertainty regarding the execution or the effects of actions. We assume that both Commander and Receiver model time as discrete, and represented by the positive integers, and that they share a common clock (and thus always agree on the current time).

Our approach treats commands as presumptive arguments for action by Receiver, represented as argument schemes with an associated list of critical questions, as [2] does for action proposals. Accordingly, we have two basic representations for commands, either as an instruction to Receiver to perform at a certain time a certain action (*Command Argument Scheme for Actions, no. 1*, or *CAS-A1*) or as an instruction to Receiver to make true at a certain time a certain set of propositions describing a world-state (*Command Argument Scheme for States, no. 1*, or *CAS-S1*):

CAS-A1: Commander instructs Receiver to perform, at time $t$, action $\alpha$.
CAS-S1: Commander instructs Receiver to bring about, at time $t$, state $S$.

An instruction issued according to CAS-S1 allows Receiver the freedom to choose whatever action Receiver believes can best achieve state $S$ at time $t$. We number these schemes because each has a variant, in which Commander provides a justification for the instruction. Building on the account of [2], a justification for an instruction could include statements regarding: the current situation in which the instructed action is to be performed (represented by the symbol $R$ in our notation below); an indication of the social context ($X$) enabling Commander to issue instructions to Receiver; the state of affairs ($S$) expected to be achieved by the performance of the instructed action ($\alpha$);

the goal ($G$) of the action, meaning the features of the state of affairs desired by Commander; the value ($v$) to be realized or enhanced by the achievement of the goal, which provides a reason why those features are desirable. Thus, goals are (partial or complete) states of the world, represented by conjunctions of propositions, while values are functions over these states. As such, goals have truth values, and these values may (at least, in principle) be verified objectively, independently of the particular agents engaged in the dialogue. In contrast, values, not being propositions, do not have truth values, and only have value with reference to a particular agent or group of agents; they may be evaluated differently by different agents, or differently by any one agent at different times. The division of the consequences of an action into a set of implied logical statements (the goal $G$) and the impact of realization of these statements on some value $v$ means that dialogue participants can potentially separate objective from subjective assessments of the new circumstances arising from successful performance of the action.

With such a structure, each argument scheme above has a variant (numbered 2) which provides a justification for the instruction, as follows:

CAS-A2: Given the social context $X$,
In the current circumstances $R$,
Commander instructs Receiver to perform action $\alpha$ at time $t$,
Which will result in new circumstances $S$,
Which will realize goal $G$,
Which will in turn promote value $v$.

CAS-S2: Given the social context $X$,
In the current circumstances $R$,
Commander instructs Receiver to bring about state $S$ at time $t$,
Which will realize goal $G$,
Which will in turn promote value $v$.

Denoting Commander by $Com$ and Receiver by $Rec$, we can represent these as:

CAS-A2: $[Com:\ Rec]:\ X, R \overset{\alpha,t}{\rightarrow} S \models G \uparrow v$

CAS-S2: $[Com:\ Rec]:\ X, R \rightarrow (S,t) \models G \uparrow v$

Our notation also allows for commands to be justified on the basis of some value $v$ being *demoted*, denoted $\downarrow v$. If we allow the action notation $\alpha$ to include representation of negative actions (i.e., not doing something), then demotion of values also permits representation of commands in which Receiver is instructed not to do some action because doing so may be deleterious for some value or may actively promote some value. For simplicity, we ignore issues regarding relationships between the timing and durations of actions on the one hand, and the timing of goal- and state-realizations, on the other.

## 3 Critical Questions

The justifications for instructions given by CAS-A2 and CAS-S2 provide a basis for elaborating a set of associated critical questions. In this paper, we present only the questions for instructions under Scheme CAS-A2, and these questions fall naturally into the following categories:

- Questions regarding the selection of the action
- Questions regarding the selection of Receiver to perform the action
- Questions regarding the authority of Commander to issue the instruction to Receiver
- Questions regarding performance of the action, including questions regarding its timing.

### 3.1 Questioning the choice of action

Most of the critical questions related to the selection of the action $\alpha$ stated in instruction CAS-A1 or CAS-A2 are the same as the questions already articulated in [3]. We list these again here (numbered *CQA1*, . . ., *CQA18*, to indicate that these are Critical Questions regarding the Action). For simplicity of presentation, we delete references to the time $t$ specified in the command.

CQA1: Are the believed current circumstances true?
CQA2: Assuming this, will the action bring about the stated new circumstances?
CQA3: Assuming all of these, will the action bring about the desired goal?
CQA4: Does the goal promote the value intended?
CQA5: Are there alternative ways of realizing the same new circumstances?
CQA6: Are there alternative ways of realizing the same goal?
CQA7: Are there alternative ways of promoting the same value?
CQA8: Does doing the action have a side effect which demotes the value?
CQA9: Does doing the action have a side effect which demotes some other value?
CQA10: Does doing the action promote some other value?
CQA11: Does doing the action preclude doing some other action which would promote some other value?
CQA12: Is the action possible?
CQA13: Are the current circumstances as described possible?
CQA14: Are the new circumstances as described possible?
CQA15: Can the desired goal be realized?
CQA16: Is the value indeed a legitimate value?
CQA17: Do the new circumstances already pertain?
CQA18: Has the action already been performed?

The last two questions are additional to those in [3]. A reason not to obey a command is that the action being commanded has already been performed at an earlier time and the resulting world-state already brought into being. This may be a fact known to Receiver but not yet known to Commander, especially if they are situated in a distributed computational environment.

Commander may respond to any of these questions with further elaboration and justification of the instruction, or with its withdrawal. The type of justification dialogue between Commander and Receiver will differ, according to the particular critical question posed by Receiver; these different dialog types are listed in [3].

### 3.2 Questioning the choice of Receiver to perform the action

This second category deals with questions regarding why Receiver, *in particular*, was selected by Commander to perform the action stated in the instruction. Here, we are dealing with questions regarding availability and capability of agents other than Receiver to execute the instruction, and the criteria used by Commander in selecting Receiver. The critical questions are:

CQR1: Is there another agent available to perform the action at the stated time?
CQR2: Is there another agent with sufficient knowledge to perform the action at the stated time?
CQR3: Is there another agent with access to the resources required to perform the action at the stated time?
CQR4: Is there another agent with sufficient experience to perform the action at the stated time?
CQR5: Is there another agent with the appropriate skillset able to perform the action at the stated time?
CQR6: Is there another agent who can see to it that the action is performed at the stated time?
CQR7: Is there another agent more suitable than Receiver to perform the action at the stated time?

In responding to such questions, Commander may provide information on the expected consequences of Receiver performing the action and/or Receiver not performing the action. These consequences could be for Receiver, for Commander, for other agents in the system as individuals, or for the system as a whole. Statements by Commander of such consequences can be understood as elaborations of the Consequential State ($S$), the Goal ($G$), and the Value ($v$) components of CAS-A2. If the answer to any of the questions CQR1 – CQR7 is YES, then the action to be performed may be re-assigned to another agent, not Receiver. In that case, Commander will need to withdraw the instruction issued to Receiver, and issue a new instruction to the other agent. Note that answers to questions CQR1 – CQR6 may all be YES, and yet the answer to question CQR7 may be NO: Receiver may be assessed by Commander as the best, or most suitable, by some criteria, of all those agents able, willing and available to do the task.

### 3.3 Questioning Commander's authority to command Receiver

The third category of critical questions are those regarding the authority of Commander to issue an instruction to Receiver within the social context stated in CAS-A2. They deal with issues such as: the social status of Commander; the social roles of the two agents; and the social or individual consequences of obeying or disobeying orders. The critical questions that arise here are:

CQX1: Under what authority does Commander have the power to issue such an instruction to any agent?
CQX2: Under what authority does Commander issue the instruction to Receiver?
CQX3: Under what authority does Commander issue the instruction to Receiver to perform the stated action?
CQX4: Under what authority does Commander issue the instruction to Receiver to perform the stated action at the stated time?
CQX5: What are the consequences of compliance with the instruction?
CQX6: What are the consequences of non-compliance with the instruction?

Responses to these questions will depend upon the nature of the social context (denoted by $X$ in CAS-A2), which links Commander and Receiver. As with the previous category of questions, the consequences of compliance or non-compliance with the instruction may impact upon the Receiver, the Commander, other agents in the system, and/or the system as a whole; similarly, statements by Commander of such consequences in response to Questions CQX5 or CQX6 can be understood as elaborations of the Consequential State ($S$), the Goal ($G$), and the Value ($v$) components of CAS-A2. Implementing computational entities able to entertain and respond to questions such as these would, of course, require a formal representation of the social context connecting the agents, as for example in the work of Karunatillake [11], or in multi-agent systems with explicitly-defined roles, rights and responsibilities, such as those developed using software engineering methodologies such as *Gaia* [20].

### 3.4 Questions regarding the performance of the action

The final category of questions relate to the performance and timing of the action. As with the questions above, the answers to these questions may inhibit an agent's ability or make it impossible to execute an instructed action, and so may serve to defeat the command. However, these questions may also be asked by a Receiver willing to obey the command, in order to clarify the instruction. In this category are the following questions:

CQP1: By what time should Receiver complete the action?
CQP2: How long should the action take to complete?
CQP3: At what place should Receiver perform the action?
CQP4: How (by what methods) should Receiver perform the action?

CQP5: May Receiver delegate the performance of the action?
CQP6: May Receiver perform the action in concert with other agents?
CQP7: For joint actions, with whom should Receiver perform the action?

Joint actions and delegated actions lead to consideration of issues such as: the division of actions into constituent tasks; co-ordination and timing of these tasks; the willingness and availability of other agents to execute their assigned tasks; etc. For reasons of simplicity, we ignore these issues in this initial account.[5]

## 4 Protocol Syntax

We now present the syntax of a multi-agent dialogue protocol to enable agents to issue, accept, reject, question, challenge, justify and retract commands. We call the protocol the *Command Dialogue Protocol* (**CDP**). The syntax is based to some extent on that of the **PARMA Protocol** for arguments over proposals for action [2], extended to deal with commands and associated questions and challenges. As is standard in the agent communications literature, we represent agent utterances by a two-layer syntax: an inner, or content, layer and an outer, or wrapper, layer. The outer layer comprises locutions which express the illocutionary force of the inner content. The inner layer includes the following elements (expressed in suitable formal representations): a time-stamp; an identifer for the speaker of the utterance; an identifier for the intended recipient of the locution; and the conversational content of the utterance.[6]

Generic (uninstantiated) locutions are denoted with just the wrapper as, for example, in WITHDRAW(.), while instantiated locutions are denoted with both wrapper and contents shown, as in WITHDRAW($t, Ag1, Ag2$), where $t$ is a time-stamp, and $Ag1$ is an agent identifier indicating the speaker, and $Ag2$ the intended recipient of the utterance. For simplicity, we assume there are just two participating agents, Commander and Receiver, denoted $Com$ and $Rec$ as before. We also assume, as in [13], that CDP contains standard control locutions for participants to initiate, enter into and withdraw from dialogues using the protocol.

After the opening, a command dialogue using CDP begins with Commander issuing a command to Receiver to perform a specified action at a specified time; Table 4.1, discussed below, presents the locutions available to Commander for this utterance. Following this utterance, CDP then permits Receiver to respond to the command, using the locutions in Table 4.2, discussed below. If Receiver issues a question or challenge to Commander in response to the command, Commander may respond in turn, using the locutions in Table 4.1 to answer a question, or to state or justify an aspect of the command, or to retract some aspect of the command. The next four sub-sections discuss the utterances, responses and counter-responses in more detail.

[5] Note that the argument scheme for action proposals of [2] has recently been extended in [1] to deal with joint actions.

[6] We assume faultless transmission, so that the intended recipient is also always the hearer of the utterance.

### 4.1 Issuing or retracting a command

As in Section 3, we assume argumentation scheme CAS-A2 is used to issue instructions. Table 4.1 shows the locutions available to Commander for stating the instruction and for stating the elements of its justification, and for retracting any of these statements after their utterance. We suppress the agent identifiers and time-stamp from these expressions, except in the case of the utterances indicating or cancelling the action to be undertaken. We also assume, for simplicity, that the value $v$ is one promoted by the action. The time variable $t$ specified in the following locutions refers to the start time of execution of the action, $\alpha$. Note that, in this protocol, only the agent issuing an instruction has the dialectical power to retract it and thus to revoke the command it embodies.

**Table 1. Locutions to issue or revoke a command, or elements of its justification**

| Command & Justification Locutions | Retraction Locutions |
|---|---|
| State_context($X$) | Deny_context($X$) |
| State_circumstances($R$) | Deny_circumstances($R$) |
| State_action_command($Com, Rec, \alpha, t$) | Retract_action_command($Com, Rec, \alpha,$ t) |
| State_consequences($\alpha, t, R, S$) | Deny_consequences($\alpha, t, R, S$) |
| State_logical_ consequences($S, G$) | Deny_logical_ consequences($S, G$) |
| State_purpose($v$) | Deny_purpose($v$) |

### 4.2 Responding to a command

We now now list the high-level locutions available to Receiver to respond to an instruction from Commander. These are shown in Table 4.2. Receiver may accept or refuse the instruction issued by Commander, with the first or the second locutions shown there. The third locution, *Question_action_command(.)*, allows Receiver to indicate a desire to seek clarification or further information from Commander. As stated earlier, such clarification may be sought whether Receiver intends to obey or not to obey the instruction. The fourth locution, *Challenge_action_command(.)*, indicates that Receiver wishes to challenge the instruction or some aspect of its justification. These two locutions will then lead to subsequent utterances by Receiver with specific questions or challenges, as discussed in Section 4.3 below.

The locution *Done_action_command($Com, Rec, \alpha, t$)*, uttered by Receiver, indicates that Receiver has executed at time $t$ the action $\alpha$ which was assigned by Commander in the command utterance. It may be that Receiver has only partially executed action $\alpha$, or has executed it at some other time, or has executed some other action whose performance obviates the need to execute action $\alpha$, or whose performance precludes the execution of $\alpha$. In such cases, Receiver may respond with an appropriate instantiation of the final locution in Table 4.2, namely *Action_done($\beta, s$)*, where $\beta$ denotes some action, and $s$ a time-point. This locution can also be used if action $\alpha$ or some other action $\beta$ has been executed by another agent, not Receiver, and such execution obviates or precludes now the need to execute $\alpha$ at time $t$. Thus, this final locution in Table 4.2 permits Receiver to make an initial response to an instruction in the case where the commanded action does not need to be undertaken.

**Table 2. High-level locutions in response to a command**

| |
|---|
| Accept_action_command($Com, Rec, \alpha, t$) |
| Refuse_action_command($Com, Rec, \alpha, t$) |
| Question_action_command($Com, Rec, \alpha, t$) |
| Challenge_action_command($Com, Rec, \alpha, t$) |
| Done_action_command($Com, Rec, \alpha, t$) |
| Action_done($\beta, s$) |

### 4.3 Questioning or challenging a command

The content of the various questions and challenges available to Receiver in response to an instruction issued by Commander are those listed in Sections 3.1 to 3.4. For each of the questions listed there, we may specify two associated dialogue locutions, in a similar manner to that adopted for the specification of the **PARMA Protocol** for the first 16 critical questions of Section 3.1 [2]. One of these two dialogue locutions is intended to seek from Commander further information or a justification of the relevant issue. The other locution is intended to deny a justification provided by Commander of the relevant issue. Consider, for example, critical question CQR1 in Section 3.2, which asks if there is another agent available to perform the action stated in the command at the time specified. Receiver may respond with a question to Commander, *asking* if there is another agent available to perform the stated action at the stated time. Or, Receiver may respond with a challenge to Commander, *denying* the assertion that no other such agent exists. The dialectical force of these two utterances is, of course, very different: a challenge, unlike a question, is an attack by Receiver on the dialectical position of Commander.

Accordingly, corresponding to the 38 critical questions presented in Sections 3.1–3.4, **CDP** has 76 locutions available to Receiver for questioning or challenging the command and its justification issued by Commander.[7] For reasons of space we do not list all of these here. Instead, Table 4.3 provides an example of these locutions, giving the

[7] In case this number of locutions is thought prolix, recall that **CDP** is intended for machine-to-machine communications; for comparison, the machine interaction protocol, Hypertext Transfer Protocol (HTTP), defines 41 standard status-code responses to a GET command, and allows for several hundred additional non-standard codes [15].

14 locutions corresponding to the seven critical questions CQR1–CQR7 of Section 3.2; these seven critical questions relate to the choice of agent selected by Commander to perform the action specified in the instruction.

**Table 3. Locutions to question or attack the choice of agent to perform the action**

| CQ | Locutions to question the choice of agent | Locutions to challenge the choice of agent |
|---|---|---|
| CQR1 | Ask_if_other_agents_available(.) | Deny_no_other_agents_available(.) |
| CQR2 | Ask_if_other_agents_with_knowledge(.) | Deny_no_other_agents_with_knowledge(.) |
| CQR3 | Ask_if_other_agents_with_resources(.) | Deny_no_other_agents_with_resources(.) |
| CQR4 | Ask_if_other_agents_with_experience(.) | Deny_no_other_agents_with_experience(.) |
| CQR5 | Ask_if_other_agents_with_skillset(.) | Deny_no_other_agents_with_skillset(.) |
| CQR6 | Ask_if_other_agents_can_see_to_it(.) | Deny_no_other_agents_can_see_to_it(.) |
| CQR7 | Ask_if_other_agents_more_suitable(.) | Deny_no_other_agents_more_suitable(.) |

### 4.4 Responding to a question or challenge to a command

If Receiver utters a question or challenge to a command locution uttered by Commander, Commander has several alternatives in responding to it. Commander may say nothing at all or may withdraw from the interaction, as may happen at any time in any dialogue between autonomous agents. Commander may provide answers to questions or provide justifications to challenges from Receiver by stating (or re-stating) some aspect of the justification for the command, using the locutions in the left-hand column of Table 4.1. In the case where these justifications have already been uttered in the dialogue, and Receiver questions or challenges them, Commander may utter supporting evidence or arguments, again using the locutions in the left-hand column of Table 4.1, but instantiated with new content. In response to a question or challenge, Commander may also retract the initial command, using the *Retract_action_command(.)* locution of Table 4.1. Commander may also simply re-state the command.

The Command Dialogue Protocol has no guarantee of eventual termination or even of loop-freeness (i.e., freeness from circularity), since Commander may simply repeatedly restate a command which Receiver repeatedly rejects, or which Receiver repeatedly accepts but never executes. Whether such features are considered desirable or not will depend upon the goals and values of the dialogue participants. For automated applications, it may be useful to explore modifications to **CDP** to avoid or ameliorate such protocol features, perhaps using the *instruction completion blocks* of **IDL3** [7, p. 259].

## 5 Outline of Semantics

We now present a semantic framework for **CDP**, which we merely sketch, for space reasons. We require both a semantics for command statements, and for the CD Protocol overall. For commands, we can utilize the double-possible-worlds semantics recently proposed by Reed and Norman [18] to formalize Hamblin's Action-State Semantics for imperatives [9]. This formal semantics includes representation both for world-states and for events (including actions); both are needed to accommodate actions which do not achieve their intended outcome-state, and states arising not through any deliberate action. Reed and Norman propose a ternary relationship across states and events, indicating that a given world-state is accessible from another by means of a particular event. Additional accessibility relationships encode time relationships between world-states, events and one another, in such a way that alternative models of time are possible. We have designed our representation for commands in order to map neatly into this semantic framework.

For the semantics of the protocol **CDP**, it would be straightforward to define an axiomatic semantics of pre- and post-conditions for each legal utterance in terms of their effects on the state of the dialogue. Given a suitable mental model of the participating agents, for example, in terms of the Beliefs, Desires and Intentions of each agent, it would be possible to extend these pre- and post-conditions to include the mental states of the participants, as does the Semantic Language *SL* of the FIPA Agent Communications Language *ACL* [5].[8] Instead of this, we adopt the denotational trace semantics approach presented for dialogues over action in [12], as refined in [13]. In this approach, participants in a multi-agent dialogue are viewed as jointly creating and manipulating objects in a shared conceptual space of action-intentionality tokens, analogously to the joint creation of natural language semantics in Discourse Representation Theory in linguistics [10]. Each token in the shared space represents a possible action, tagged with meta-information providing information about the preferences and intentions of agents regarding that action. Examples of such meta-information includes: the identity of the dialogue participant who first proposed the action; the identity(ies) of the intended executor(s) of the action; the identities of dialogue participants, if any, who have currently endorsed the proposal; and the identities of participants with the power to revoke the proposal. The tags on tokens enable the shared token space to be partitioned into sub-spaces, with different read-, write-, and delete-permissions applying in different sub-spaces. These sub-spaces may also be viewed as generalizations of the participants' Commitment Stores. Preferences between different actions expressed in the dialogue by participants are represented by labeled arrows between tokens.

In the case of commands, only the proposer of an instruction, Commander, has the power to revoke it, which simplifies the tags (and hence the sub-space partition structure). Moreover, **CDP** does not (currently) permit expression of preferences, so there are no arrows created between distinct tokens. Each command utterance in a dialogue using **CDP** causes the creation of an action-intentionality token in an appropriate subspace of the shared space. Utterances to indicate acceptance or rejection of a command

[8] However, such an extension to include mental states would not, in general, be verifiable.

likewise lead to tagging of the associated token, or equivalently, the creation of tokens in other sub-spaces of the overall space. A complete articulation of the mapping between **CDP** utterances and objects in the token-space semantics will be presented in later work.

## 6 Conclusion

We have a proposed a novel representation for commands in computational systems, with each command represented as a presumptive argument scheme for action by a designated agent along with a list of critical questions, in the manner of [2]. We have then used this representation to specify a multi-agent dialogue protocol for command dialogues, **CDP**. A denotational semantic framework for this protocol has also been outlined, with commands interpreted via the Action-State double-possible-worlds semantics of [18], and dialogue utterances understood as intended to manipulate action-intentionality tokens in a shared conceptual space, following [13]. Our work lies in the field of software engineering of multi-agent systems using argumentation: we have presented a principled framework for the design of a machine-to-machine dialogue protocol which would enable commands to be issued, questioned, challenged and justified between autonomous software agents. To the best of our knowledge, **CDP** is the first computational framework to enable multi-agent command dialogues.

Of course, much work remains to be done before this framework would be ready for production deployment. In addition to a full articulation of the semantics, and the development of a prototype implementation, several other aspects require further research. Firstly, a computational representation of the social context between the participants, as in [11], may be useful to provide a stronger semantic underpinning of those critical questions relating to the authority of the Commander (CQX1–CQX6). In some social contexts, of course, agents issuing commands may have access to arsenals of rewards or threats to ensure compliance; although a different topic to that of command dialogs, work in that area may be relevant, e.g., [17]. Secondly, we plan to refine our model of time, so as to enable more sophisticated treatment of the relationships between the timing and duration of actions and the timing of achievement of world-states and goals. We also plan to consider how to incorporate uncertainty regarding the success or failure of actions, and discussion of the relative costs and benefits of alternative actions, possibly drawing on the qualitative decision theory of [6]. Such work would support the development of appropriate critical questions for argument scheme CAS-S2, which we did not consider in this paper. Finally, recent work in multi-agent systems has looked at delegation and responsibility, e.g., [16]; it may be valuable to explore the relationships between that work and multi-agent dialogs over commands requiring joint action for their fulfilment.

**Acknowledgments** We thank the referees for their comments. Two authors (PM and SP) are grateful for partial financial support received from the EC *Information Society Technologies* (IST) programme, through project ASPIC (IST-FP6-002307). This work was also partially supported by the US Army Research Laboratory and the UK Ministry of Defence under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the US Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

## References

1. K. Atkinson and T. Bench-Capon. Action-based alternating transition systems for arguments about action. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*, pages 24–29. AAAI Press, 2007.
2. K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument for proposals over action. *Autonomous Agents and Multi-Agent Systems*, 11(2):153–171, 2005.
3. K. Atkinson, T. Bench-Capon, and P. McBurney. Computational representation of practical argument. *Synthese*, 152(2):157–206, 2006.
4. T. J. M. Bench-Capon, P. E. Dunne, and P. H. Leng. Interacting with knowledge-based systems through dialogue games. In *Proceedings of the Eleventh International Conference on Expert Systems and Applications*, pages 123–140, Avignon, 1991.
5. FIPA. Communicative Act Library Specification. Standard SC00037J, Foundation for Intelligent Physical Agents, 3 December 2002.
6. J. Fox and S. Parsons. Arguing about beliefs and actions. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*, LNAI 1455, pages 266–302. Springer, Berlin, 1998.
7. R. Girle. Commands in Dialogue Logic. In D. M. Gabbay and H. J. Ohlbach, editors, *Practical Reasoning: Proceedings of the First International Conference on Formal and Applied Practical Reasoning (FAPR 1996)*, LNAI 1085, pages 246–260, Berlin, 1996. Springer.
8. J. Habermas. *The Theory of Communicative Action: Volume 1: Reason and the Rationalization of Society*. Heinemann, London, 1984. Translation by T. McCarthy.
9. C. L. Hamblin. *Imperatives*. Basil Blackwell, Oxford, 1987.
10. H. Kamp and U. Reyle. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer, Dordrecht, 1993.
11. N. C. Karunatillake. *Argumentation-Based Negotiation in a Social Context*. Phd, School of Electronics and Computer Science, University of Southampton, Southampton, UK, 2006.
12. P. McBurney and S. Parsons. A denotational semantics for deliberation dialogues. In I. Rahwan, P. Moraitis, and C. Reed, editors, *Argumentation in Multi-Agent Systems*, LNAI 3366, pages 162–175. Springer, Berlin, 2005.
13. P. McBurney and S. Parsons. Retraction and revocation in agent deliberation dialogs. *Argumentation*, 21(3):269–289, 2007.
14. D. J. Moore. *Dialogue Game Theory for Intelligent Tutoring Systems*. PhD thesis, Leeds Metropolitan University, Leeds, 1993.
15. Network Working Group. Hypertext Transfer Protocol — HTTP/1.1. Technical Report RFC 2616, Internet Engineering Task Force, June 1999.

16. T. J. Norman and C. Reed. Delegation and responsibility. In Y. Lesperance and C. Castelfranchi, editors, *Intelligent Agents VII*, LNAI 1986, pages 136–149, Berlin, 2001. Springer.
17. S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. Negotiating using rewards. In H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, editors, *Proceedings of the Fifth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006)*, pages 400–407, Hakodate, Japan, 2006. ACM Press.
18. C. Reed and T. Norman. A formal characterisation of Hamblin's action-state semantics. *Journal of Philosophical Logic*, 36:415–448, 2007.
19. D. N. Walton. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, Mahwah, NJ, 1996.
20. F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multiagent systems: the Gaia methodology. *ACM Transactions on Software Engineering Methodology*, 21(3):317–370, 2003.