Towards a Graphical Model of Remote Control of Machines

Gareth TYSON [a], Nicolas TROQUARD [b], Simon PARSONS [c] and Peter MCBURNEY [a,1]

[a] *King's College London, Strand, London WC2R 2LS, UK*
[b] *ISTC-CNR, Laboratory for Applied Ontology, Trento, Italy*
[c] *Department of Computer and Information Science, Brooklyn College, NY 11210, USA*

Abstract. In this paper we present a novel computational model of control of a machine, device or software programme by an autonomous agent, which may be remote. The model allows for different degrees of remote and mediated control, and seeks to represent a common situation with the remote use of scientific instruments. We present the model both as a logical language and through a formal diagrammatic semantics, in the form of encapsulated and annotated graphs. The model supports automated reasoning over access powers, including the automated allocation of powers to a remote user by the owner of a device. In addition, the model enables an intelligent agent to select appropriate multi-agent protocols to engage in argument about access to a device or the details of actions to be undertaken using the device. The key value of this work is in automating access and in enabling automated reasoning over complex networks of devices.

Keywords. remote control, multi-agent dialogs, commands, command dialogs, access dialogs, distributed systems

1. Introduction and motivation

Electron microscopes use beams of electrons either to view or to probe some object of study, called the sample. The output of such operations may be an image, data generated by the probe, or both. High-end electron microscopes, such as Scanning Transmission Electron Microscopes (STEMs), typically require trained human technicians to operate them, and also to prepare and position the sample object inside the microscope. Preparation may entail prior modifications to the sample such as staining, dehydration, or chemical etching; these actions may be done on-site, or at some other location. The local technical staff, however, are not usually the initiators or end-users of the research undertaken using the microscope; instead, the end-users are other people or teams.

Because of their high-cost, funders of STEM machines, as with funders or owners of other expensive scientific equipment, are usually keen to promote wide use of the equipment. The rise of the Internet has naturally led owners of such machines to seek to allow remote users to access their devices [6]. This is not necessarily difficult to enable techni-

[1] Corresponding Author: Peter McBurney, King's College London, Strand, London WC2R 2LS, UK; E-mail: peter.mcburney@kcl.ac.uk

cally,[2] but there may be policy reasons why particular remote users should or should not have remote access. Access to a remote user may be granted for some actions using the device, but not others, or at some times and not others. In any case, some activities, such as preparing samples for examination under the microscope, generally require a trained person on-site, so that use of the microscope typically involves a sequence of actions, some necessarily local and some not. Indeed, STEM machines usually require vacuums and vibration-free locations to operate successfully, so even on-site end-users will not actually be present in the room when the microscope is in use.

Allowing remote access to specific users for specified actions at certain times will be a matter of policy for the owners of specific devices. Automating remote access and the resulting interactions between remote users and a device is a challenge for Artificial Intelligence (AI). Some aspects of this challenge are settled. For instance, once a user (remote or local) has access, his or her interactions may be automated by means of the XML-type scripting languages developed in recent years for various scientific devices, for example telescopes [24] and microscopes [5]. Likewise, automation of decisions about whether particular users may or may not have access to a device can utilize the extensive computational models of trust developed recently in computer science [1]. In between these two stages — between deciding which users may have remote access and allowing authorized users to control the device through programs — lies an unsolved challenge to AI: how to automatically allocate access rights to trusted users.

To achieve this, we need a computational model of the possible powers of a remote user over a device. Motivated by the STEM example but generalizing from it, in this paper we present what we believe is the first such model. Our model assumes a remote user, an intelligent agent, with fewer or greater powers to manipulate a machine or device, possibly in concert with a second intelligent agent, local to the device, called *the technician*. In any actual instance, the powers granted to a remote user will depend precisely on what actions the user may do with the device at specific times; our model formally incorporates actions and times. To automate access controls, we need to be able to reason about the types of powers that potential users may have over the actions (or equivalently, the states) of the device, possibly mediated through the technician.

We first present our model conceptually, in Section 2.1, in terms of statements about the relationships, if any, that exist between a remote user and a local technician, and their respective degrees of influence over a device. We then convert these statements into a formal logical language in Section 2.2. We then present, in Section 2.3, a diagrammatic semantics for this language, with statements in the language mapped to encapsulated and annotated graphs. As mentioned, our models apply generally to the remote operation of any machine (or device or software object), not only electron microscopes. We also present rules to allow instantiations of the model to be composed in various ways so as to represent more complex systems of interactions between users and devices. In addition to representing real phenomena, our model enables automated reasoning about the powers of a remote user in such systems. This ability will be of particular value in situations involving multiple devices or large-scale and complex interactions.

One motivation for this work is to identify the system-level pre-conditions for particular types of multi-agent argumentation dialogs over access to resources, following some earlier work on automated frameworks for such dialogs [25]. The different relationships

[2]For example, Costello [5] describes a successful implementation of remote access to a STEM machine using mostly off-the-shelf equipment and commercial software packages.

between remote and local users and devices in our model lead naturally to different types of dialogs between the participants, which we present in Section 3. Essentially, our key contribution in this paper is a formal model which allows the extension of automated computational argument to a greater range of real distributed systems. We then discuss related research work in Section 4 and conclude the paper in Section 5.

2. Conceptual, Syntactical and Diagrammatic Models

2.1. Conceptual Model

How may we model the relationships between a user and machine or device? For any particular desired state of the machine, a remote end-user (call him or her *"Agent A"*) may have direct control over the machine him or herself, or may have to interact through a local technician (*"Agent B"*), depending on the access privileges granted by the device owner to A. Interacting via B, A may be empowered to instruct or command B to control the machine, or may, alternatively, only be able to request B to do so. In yet another alternative, control of the machine may only be able to be exercised by A and B acting jointly, in concert, with neither agent being able to control the machine alone. Control of the launch of nuclear missiles, for example, typically requires simultaneous joint actions by two or more physically-separated human operators, with neither operator being able to launch the missile on his or her own [10, Chapter 5].

For any action α at time t to be executed by a device X, we may therefore distinguish the following cases pertaining to Agent A's influence over X. Note that we assume the local agent B always has unilateral power over X to do α at time t, except in Case 4 where B must act jointly with A.

Case 1: Remote agent A has the power to make device X execute action α at time t.

Case 2: Remote agent A has the power to command local technician B to make device X execute action α at time t.

Case 3: Remote agent A has the power to request (but not command) local technician B to make device X execute action α at time t.

Case 4: Remote agent A and local technician B, acting in concert, have the joint power to make device X execute action α at time t. Neither agent has this power alone.

Case 5: Remote agent A has no power to make device X execute action α at time t.

These various powers of agent A are summarized in Table 1, which also refers to the arrows in Figure 1. Here, we have assumed the power by an agent over a device X is the ability to make that device execute a particular action at a specified time.[3] In addition, for simplicity, we have ignored any additional agents or social structure in agent B's organization.

---

[3]Alternatively, we could conceive of this power to be an ability to make the device achieve a certain state. Following Hamblin [12], however, we believe that explicit representation both of actions and of states is necessary for concise modeling of imperatives and their effects. Accordingly, for the first power listed (Case 1), strictly speaking we should write: *Remote agent A has the power to make device X, assumed to be in state R, execute action α at time t, so as to achieve resulting state S.* For brevity, we ignore the state representations in the remainder of this paper.

We can represent the various powers of agents A and B over the device X by the diagram of Figure 1. In Section 2.3 below, we will define such diagrams formally, while here we present an informal description. Agents and devices (objects) are represented by nodes in the figure — agents with a single circumference, and objects with a double circumference. The labels shown on the edges correspond to those in column 1 of Table 1. A direct power by agent A to make device X execute action α at time t (Case 1) is indicated by a full arrow, labeled "1", from node A to node X. A power to command (Case 2) is indicated by full arrow, while a power only to request (Case 3) is shown by a dotted arrow. A joint power by A and B (Case 4) is shown by the three arrows labeled "4". Because our concern here is with the powers of agent A, we do not have a case where agent B has power over X, shown by the arrow labeled "0". Because we seek to represent large-scale and complex control structures, each agent has a control arrow pointing to it from outside the boundary, and each object has a control arrow pointing from it outside the boundary. Such boundary-crossing arrows will enable these diagrams to be composed, as in [8].

Case	Arrow (Figure 1)	Label	Agent Power (for action α at time t)
	0	DO	B may control X
1	1	DO	A may itself control X
2	2	TELL	A may command B to control X
3	3	ASK	A may request B to control X
4	4	JOINT-DO	A and B may jointly control X
5		NO-DO	A has no powers over X

Table 1.: Powers of agents A and B over device X

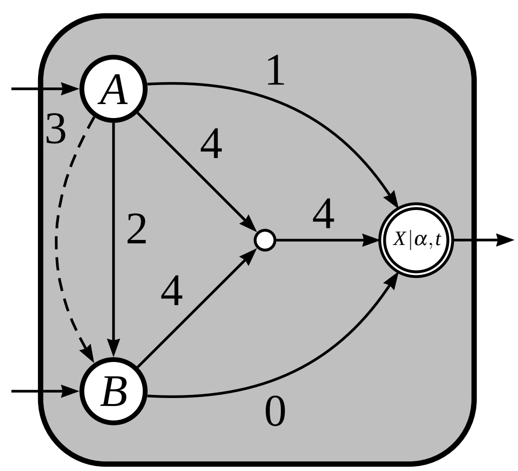

Figure 1.: Control powers of agents A and B over device X for action α at time t

How may we distinguish the power to command from the power to request? In other words, what characteristics distinguish, semantically and/or pragmatically, a TELL speech act from an ASK speech act? Speech acts about actions are social acts, requiring at least two parties, a speaker and an intended hearer; a commitment to action generally only follows once the initial speech act is accepted, or *uptaken* [30]. Uptake is usually, but not necessarily, in the power of the hearer. Once uptaken, a speech act may typically only be revoked or retracted under special circumstances, by a designated participant or participants. Thus, differences between speech acts over actions lie in the force of the respective utterances, in the powers of uptake by the intended hearer, and in the power of revocation. Although commands may be questioned or contested [3], a legitimate command by an authorized agent would ordinarily be expected to be both accepted and obeyed by the intended hearer. A request, in contrast, provides much greater freedom of volition to the party of whom the request is made, agent B in this case. In this sense, a command has greater force than a request. Another pragmatic difference is in the power of retraction: Only the speaker of a command would ordinarily have the power to revoke it, once it had been accepted. In contrast, a request may potentially be revoked by the agent which had uptaken it, depending on the social relationship between the agents [19].

2.2. Formal Syntax

In this section, we present a formal syntax for a logical language describing the powers of remote agents. We assume two finite collections — one, $\mathscr{A}$, of autonomous agents, each denoted A, B, etc, and a second, $\mathscr{D}$, of devices (or objects), each denoted X, Y, etc. The difference between the two types of entities is in their respective autonomy: When instructed to undertake an action at a specified time by an authorized agent or device, a correctly-functioning device always executes the action as instructed. An agent, similarly instructed by an authorized agent or device, however, may or may not execute the action as instructed. In other words, agents have some greater or lesser degree of autonomy; an autonomy which devices do not have. We denote atomic actions by lower-case Greek letters, α, β, γ etc, while times are denoted by the letters r, s, t etc. For simplicity, we assume the actions are independent of one another and may not be combined.

Case	Agent Power (for action α at time t)	Formal syntax
1	A may itself control X	$\Gamma(A,\text{DO},X\,|\,\alpha,t)$
2	A may command B to control X	$\Gamma(A,\text{TELL},B|X\,|\,\alpha,t)$
3	A may request B to control X	$\Gamma(A,\text{ASK},B|X\,|\,\alpha,t)$
4	A and B may jointly control X	$\Gamma(\{A,B\},\text{JOINT-DO},X\,|\,\alpha,t)$
5	A has no powers over X	$\Gamma(A,\text{NO-DO},X\,|\,\alpha,t)$

Table 2.: Syntax for powers of agents A over device X

We assume that these entities have control relationships of the five types listed in Table 1, corresponding to the five cases of Section 2.1. We denote each type of control relationship with a tuple-based formalism, as shown in column 3 of Table 2, where the symbol Γ is a label for a tuple. Each statement is a 3-tuple, where the slots of the tuple are as follows:

Position 1: An agent or set of agents, whose power is described by the statement.

Position 2: A type of power, one of the 5 types listed in column 3 of Table 1.

Position 3: An agent or device, upon whom the power in Position 2 is acted by the agent named in Position 1. Following the name of the agent or device and a vertical stroke the name of an action α and a time point t, separated by a comma, are given. If a chain of such agents or devices is acted upon, the names of these are separated by vertical strokes prior to the action and time.

We label tuples by upper-case Greek letters, Γ, Δ, Π, etc. We assume a symbolic language $\mathscr{L}$ of statements of the form expressed by these 3-tuples (and shown in column 3 of Table 2); we call such statements *atomic*. Where there is no confusion, such statements are referred to by their labels, Γ, Δ, etc. We assume further that such statements may be conjoined (&), with the obvious meaning. In addition, we define sequential combination, denoted $\Gamma; \Delta$, of two statements Γ and Δ which reference different times s and t respectively, with $s < t$, as the control power referenced in the first statement followed by the control power referenced in the second. Similarly, we define parallel combination, denoted $\Gamma \parallel \Delta$, for two statements Γ and Δ which reference the same time t. At present, we do not permit negation or disjunction of statements, because of the subtleties of dealing with negative and choice powers. In summary, the grammar of the language $\mathscr{L}$ is:

$$\mathscr{L} ::= \Gamma | \Gamma \& \Delta | \Gamma; \Delta | \Gamma \parallel \Delta.$$

2.3. Diagrammatic Semantics

We now outline a diagrammatic semantics for the formal language defined above, in the form of encapsulated and annotated graphs. It is important to note that these graphs are not merely illustrations of statements about the powers of agents over devices, but are also a denotational (i.e., mathematical) semantics for such statements. In other words, the diagrams described here constitute a basis for assigning meaning to the statements and, as defined mathematical objects, may themselves be reasoned about. In this respect, our diagrammatic model has the same formal status as commutative diagrams in category theory (for example, [15]) or other formally-defined visual languages, e.g., [7,8,34]. Indeed, our coherence theorem below implies that neither the logical syntax of the previous section nor the visual language of this section is superior to the other, when viewed as mathematical objects.

We assume a collection $\mathscr{G}$ of encapsulated and annotated graphs, labeled G, H, J, etc, each having a boundary, two types of nodes, and six types of directed edges, denoted as follows:

$$\mathrm{G}(\delta, N_{Agents}, N_{Devices}, E_{DO}, E_{TELL}, E_{ASK}, E_{JOINT-DO}, E_{in}, E_{out})$$

Here, the graph G has boundary δ, a set of agent nodes N_{Agents} and a set of device nodes $N_{Devices}$, and four sets of directed edges labeled respectively with the types DO, TELL, ASK, and JOINT-DO, and a set of incoming edges E_{in} and a set of outgoing edges E_{out}. All diagrams have boundaries, but some may not have all types of nodes or all types of edges. Diagram elements (boundaries, nodes, edges) may be annotated with their labels; agent and object nodes may additionally be annotated with the names of actions and times. For each well-formed statement in the language defined in the previous section, there is a corresponding diagram. Thus, for instance, the statement indicating that agent A may command agent B to make device X execute action α at time t, namely:

$\Gamma(A, \text{TELL}, B|X|\alpha, t)$

is mapped semantically to the following diagram (where the symbol $\rightarrow$ represents a directed edge):

$$\text{G}(\delta, \{A, B\}, \{X|\alpha, t\}, \emptyset, \{A \rightarrow B, B \rightarrow X\}, \emptyset, \emptyset, \{\rightarrow A, \rightarrow B\}, \{X \rightarrow\})$$

This diagram is shown in Figure 2. We thus assume a semantic mapping from atomic statements (i.e., those without conjunction, sequence or parallel combination) in the logical language $\mathscr{L}$ to graphs in the collection $\mathscr{G}$. For a statement $\Gamma \in \mathscr{L}$ with graph semantics $G_\Gamma \in \mathscr{G}$, we write $\Gamma \models G_\Gamma$. For non-atomic statements, the semantic mapping operates in the obvious ways, which we present here informally. Thus, for conjunction of statements we can readily define the semantics as the combined diagram, having all the nodes and edges of each constituent; likewise, for two parallel statements (those referencing the same time points), the two diagrams are stacked (placed vertically) on one another and then combined into one. For sequential statements (those referencing different time points), the two diagrams are placed horizontally next to one another and then combined into one. In all three types of combinations, nodes having the same label are overlayed, so that agents or objects referenced in each of the combined statements (and appearing in each of the combined graphs) only appear once in the conjoined diagram.

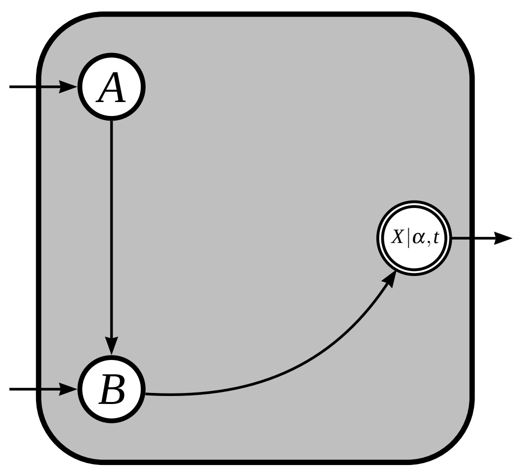

Figure 2.: Graph of power of A to command B to make X do action α at time t

We can demonstrate soundness and completeness of the syntax with the diagrammatic semantics by means of the following theorem, called a *Coherence Theorem* in category theory (see [34]). This theorem says that smoothly transforming one diagram into another in certain constrained ways (including not changing the number or types of nodes and edges, and not detaching nodes from their incoming or outgoing edges) means that the statements corresponding to the two diagrams in the logical language both represent the same types of power relationships between the agents and devices, and conversely.

Theorem 1: *Two statements $\Gamma, \Delta \in \mathscr{L}$ represent the same types of power relationships between agents and objects if and only if their corresponding graphs $G_\Gamma, G_\Delta \in \mathscr{G}$ are planar isotopic.*

Proof: The proof is straightforward, following from related coherence theorems in category theory; see [14, Theorem 1.5], [15, Theorem 1.2], and [34]. □

3. Relationship to agent dialogues

The model presented in the section above enables representation and automated reasoning over the powers of remote agents to control (or not) the actions of devices and objects. An intelligent agent equipped with that model could compare the powers it has in a particular instance with those it requires to achieve specific goals. If the powers that the agent has are not sufficient to achieve its specified goals, it could engage in automated dialog and argument with whatever agent (human or software) with the power to allocate control to the device in question. For example, a remote agent A with no powers to control a machine X may seek to have permission to request a local technician agent B to make X do specific actions. Moreover, an intelligent agent with a particular power may not necessarily just exercise that power, if it needs to interact with other intelligent agents to exercise its power. For example, although an agent A with the power to command a local agent B can just issue commands to B, there may be good reason for B to engage in dialog with A, for example, to clarify the instruction or to question its feasibility, legality or timing, even when B fully accepts the authority of A to issue commands, and intends to obey this particular command. An intelligent agent having command power over another intelligent agent should thus expect to have to engage in a command dialog.

One attractive feature of our model is the natural relationship revealed to different types of multi-agent dialogs, with each type of control corresponding to particular dialog types. These are shown in Table 3. Research on some of these dialog types and articulation of protocols for them has been undertaken in AI within the last several years; for instance, Persuasion dialogs [26,31,36], Negotiation dialogs [28], Command dialogs [3], Information-Providing dialogs [35], Deliberation dialogs [17], and Propose-Action dialogs [2]. Dialogs of different types may also be embedded in these dialog types. For example, an agent A having only the power to request an action from agent B may seek to persuade B through a Persuasion Dialog. To consider this request, B may seek information from A through an embedded Information-Seeking Dialog, or the two may even enter into an embedded or subsequent Negotiation Dialog. Formal models allowing automation of such dialog combinations exist, e.g., [18,29].

As mentioned earlier, having a formal model allows for reasoning about the powers of agents over devices and over other agents. Thus, vicarious requests or commands may be reasoned about in a more sophisticated manner. For instance if agent B can control device X, and agent A may command B, and agent G may command A, then commands from B to X may be in fact originate with G. If the policy of the owners of X were not to accept control by G then such a chain of commands could be identified and precluded. Such policies would be important in situations, such as financial trading, where separate control paths are needed to prevent fraud or conflicts of interest.

Power	Initial Dialog Type	Possible Continuation Dialogs
DO	None	Information-Providing
TELL	Command Dialog	
ASK	Request-Action Dialog	Persuasion Information-Seeking Negotiation
JOINT-DO	Deliberation Dialog	Persuasion Information-Seeking Negotiation
NO-DO	Propose-Action Dialog	Persuasion Information-Seeking Negotiation

Table 3.: Dialog types for different powers

4. Related work

Recent work in multi-agent and distributed systems includes various formalizations of delegation and power, but none of these models have exactly what is needed for the situations we seek to represent. We discuss the main related work, from the closest outwards.

Models of delegation: Reasoning about delegation (influence, imperatives, etc) has been a prominent topic of investigation in *logics of agency* that identify an action with its results [4,16,32,33]. In this tradition, instead of formalizing "John kills the coyote", one writes a formula $\mathsf{S}_{John}\mathtt{dead}$ that stands for "John brings about (sees to it) that the coyote is dead". There is always some idea of responsibility of an agent for some state of affairs that she brings about. Thus, expressing a delegation in logics of agency generally follows the pattern "the delegat*or* brings about that the delegat*ee* brings about that ϕ". In formula $\mathsf{S}_{or}\mathsf{S}_{ee}\phi$.

As logics of agency concentrate on the results of actions, they generally are not fit to talk about the means of an activity. Killing a coyote might well be an event of shooting, but that would be no different from an event of strangling. This, of course, is often problematic for reasoning about "real-world" delegation procedures. Just as it might be preferable to shoot a coyote rather than to strangle it, delegation instructions in agent organizations are often about one delegator asking or telling the delegatee to perform some specific action in order to achieve some expected result.

Norman & Reed [23] propose a logic of agency and action evaluated on models inspired by Hamblin's model [12]. As in logics of agency, $\mathsf{S}_i\phi$ reflects the responsibility of i for ϕ. But an additional operator T_i is introduced and action terms are first class citizens of the object language. The formula $\mathsf{T}_i\alpha$ means that agent i is responsible for action α. Therefore, additional modes of delegation can be formulated, e.g., $\mathsf{S}_{or}\mathsf{T}_{ee}\alpha_{\mathsf{shoot}}$, or $\mathsf{S}_{or}\mathsf{S}_{ee}\mathsf{T}_{ee}\alpha_{\mathsf{strangle}}$. This still presents a certain limitation since one cannot nest a modality in the scope of a T_{or} modality, be it either a S_{ee} or T_{ee}.[4] In particular, one cannot formalize a sequence of actions that would represent a long-term plan of a remote control instruction, with delegation, and command or persuasion dialog. For this purpose, Dynamic Logics [13,21] are more suitable, but they are logics of events rather than logics of action. As such, they generally fail to capture a notion of agentive responsibility that is adequate to reason about delegation in organizations.

Arguments over access: In earlier work, one of us considered automated multi-agent arguments over access to resources or devices [9,25]. The focus of that work was the creation of a computational framework to support automated multi-agent interactions enabling a supplicant agent, agent *A*, to argue with a resource controller so as to move from a NO-DO access-state to a DO access-state, with respect to a particular resource at a particular time. We did not consider the other types of power presented here, nor the notion of a local agent able to control the resource through whom the powers of the remote agent may need to be mediated.

Distributed systems architectures: An applicable area of our work is distributed system architectures; by this we mean the interaction paradigms in which distributed components cooperate. Currently, two prominent models exist: *client-server* and *peer-to-peer*. In a client-server model, one or more client machines *request* a server machine to provide a service or a digital object to the client, as in, for example, Hyper-Text Transfer Protocol (HTTP) [22]. In contrast, a peer-to-peer (P2P) model involves the decentralization of this server functionality in that each participant can both consume *and* provide services. These models do not encompass the level of detail that our model does. Most prominently, in both models, components can only ever request services; a distinguished concept of command does not exist. Instead, the distributed systems classification focusses on *who* rather than *what*. As such, our model extends these paradigms to place constraints on exactly *what* each component can do, as well as the expectations of the other components around it.

Visual reasoning: Formal reasoning over diagrams has a long history in pure mathematics, dating from Euclid's geometry to mid-20th-century commutative diagrams in category theory. Recently, in addition to mathematics, such formal visual reasoning has become important in areas of computer science, systems biology, and theoretical physics, for example [7,8,15,34]. The diagrammatic semantics presented in this paper is different in detail from any of that work, but presented in the same spirit.

5. Conclusions

In this paper we have presented a novel formal model of the types of power which an autonomous, intelligent agent may exercise over a device or software object, possibly

[4] But we can do this with compositions of diagrams.

mediated through another autonomous, intelligent agent. *Inter alia*, our model distinguishes the power to command another agent to do some action from the power merely to request that agent to do that action. This is a distinction we have not seen in any other formal model of agent power. The model was presented in the form of a logical language along with a formal diagrammatic semantics, and thus supports automated reasoning over such powers. The model is defined at the level of the atomic actions of the device at specific times, in a way which readily enables composition of statements in the syntax (equivalently, diagrams in the semantics) to represent large-scale or complex control flow situations.

One benefit of such a reasoning capability is that questions about systems of control with complex control flow may be interrogated, for example, to discover if a chain of controls requires permissions or joint actions at any point, or to determine which agents in a complex chain have volition not to act. Another benefit of being able to reason automatically about the powers of remote agents using such a model is that the allocation of such powers can thus, when combined with appropriate models of trust, be automated. A further benefit of this model is that it demonstrates a natural relationship between the types of control a remote agent may have over a device and the types of dialog that an agent may enter into with the owner or controlling authority for the device. This relationship can be used by both agents to guide their choice of dialog type when arguing about the level of control granted, or about the specific actions which the remote agent requires to be undertaken.

Possible applications of this work are to business process modeling, and to military communications and operations planning, particularly for coalitions of forces where multiple entities may have partial or shared control of resources. We intend to explore these in future work. We will also consider situations where the underlying atomic actions may be combined, as in [20], and situations with more than two agents. Finally, a major direction for future work is to understand the nature and types of joint action (i.e., Class 4). Researchers in AI and multi-agent systems have done much work on joint actions and shared intentions, for example the SharedPlans framework of Grosz and colleagues [11]. There is also a relationship between the types of joint action and theories of concurrency, and we believe that scope exists to represent the range of joint actions with a continuous mathematical model, as in Pratt's influential geometric model of concurrency [27].[5]

References

[1] D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Web Semantics*, 5(2):58–71, 2007.

[2] K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue-game protocol for multi-agent argument over proposals for action. *Autonomous Agents and Multi-Agent Systems*, 11(2):153–171, 2005.

[3] K. Atkinson, R. Girle, P. McBurney, and S. Parsons. Command dialogues. In I. Rahwan and P. Moraitis, editors, *Proceedings, Fifth International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2008)*, pages 9–23, Lisbon, Portugal, 2008. AAMAS 2008.

[4] B. F. Chellas. *The Logical Form of Imperatives*. Perry Lane Press, Stanford, CA, 1969.

[5] W. Costello. Remote microscopy and open source data structures. M.Phil., Department of Computer Science, University of Liverpool, Liverpool, UK, 2008.

[5] **Acknowledgments:** We are grateful for discussions on these topics with Andrew Bleloch, William Costello, Michael Fisher, Peter Goodhew, Jarred McGinnis, and the late Ray Paton.

[6] W. Costello, A. Bleloch, P. Goodhew, P. McBurney, and R. Paton. A new inclusive approach to remote microscopy (Extended Abstract). In *Proceedings, Thirteenth European Microscopy Congress (EMC 2004)*, Antwerp, Belgium, 2004.

[7] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.

[8] L. Dixon and A. Kissinger. Open graphs and monoidal theories. *Mathematical Structures in Computer Science*. *In press*.

[9] S. Doutre, P. McBurney, and M. Wooldridge. Law-Governed Linda as a semantics for agent interaction protocols. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *Proceedings, Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2005)*, pages 1257–1258, New York City, NY, USA, 2005. ACM Press.

[10] J. A. Engel, C. Slattery, M. Ebeling, E. Pogany, and A. R. Squitieri. *The Missile Plains: Frontline of America's Cold War*. US National Parks Service, Minuteman Missile NHS, SD, USA, 2003.

[11] B. J. Grosz and S. Kraus. The evolution of sharedPlans. In *Foundations and Theories of Rational Agency*, pages 227–262, 1998.

[12] C. L. Hamblin. *Imperatives*. Basil Blackwell, Oxford, UK, 1987.

[13] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.

[14] A. Joyal and R. Street. Planar diagrams and tensor algebra. Unpublished manuscript, 1988.

[15] A. Joyal and R. Street. The geometry of tensor calculus I. *Advances in Mathematics*, 88(1):55–112, 1991.

[16] E. Lorini, N. Troquard, A. Herzig, and C. Castelfranchi. Delegation and mental states. In *Proceedings, Fourth Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007)*, pages 622–624, 2007.

[17] P. McBurney, D. Hitchcock, and S. Parsons. The eightfold way of deliberation dialogue. *International Journal of Intelligent Systems*, 22(1):95–132, 2007.

[18] P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3):315–334, 2002.

[19] P. McBurney and S. Parsons. Retraction and revocation in agent deliberation dialogs. *Argumentation*, 21(3):269–289, 2007.

[20] R. Medellin Gasque, K. Atkinson, P. McBurney, and T. Bench-Capon. Arguments over co-operative plans. In S. Modgil, N. Oren, and F. Toni, editors, *Proceedings, First International Workshop on Theory and Applications of Formal Argumentation (TAFA 2011)*, Barcelona, Spain, 2011.

[21] T. Miller and P. McBurney. Propositional dynamic logic for reasoning about first-class agent interaction protocols. *Computational Intelligence*, 27(3):422–457, 2011.

[22] Network Working Group. Hypertext Transfer Protocol — HTTP/1.1. Technical Report RFC 2616, Internet Engineering Task Force, June 1999.

[23] T. J. Norman and C. Reed. A logic of delegation. *Artificial Intelligence*, 174(1):51–71, 2010.

[24] C. Pennypacker, M. Boer, R. Denny, F. V. Hessman, J. Aymon, N. Duric, S. Gordon, D. Barnaby, G. Spear, and V. Hoette. RTML – a standard for use of remote telescopes. *Astronomy and Astrophysics*, 395:727–731, 2002.

[25] L. Perrussel, S. Doutre, J.-M. Thevenin, and P. McBurney. A persuasion dialog for gaining access to information. In I. Rahwan, S. Parsons, and C. Reed, editors, *Argumentation in Multi-Agent Systems*, LNAI 4946, pages 63–79, Berlin, Germany, 2008. Springer.

[26] H. Prakken. Models of persuasion dialogue. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, chapter 14, pages 281–300. Springer, Berlin, Germany, 2009.

[27] V. R. Pratt. Modeling concurrency with geometry. In *Proceedings, Eighteenth Annual ACM Symposium on the Principles of Programming Languages*, pages 311–322. ACM, New York, USA, 1991.

[28] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and E. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4):343–375, 2003.

[29] C. Reed. Dialogue frames in agent communications. In Y. Demazeau, editor, *Proceedings, Third International Conference on Multi-Agent Systems (ICMAS-98)*, pages 246–253. IEEE Press, 1998.

[30] A. Reinach. Die apriorischen Grundlagen des bürgerlichen Rechtes. *Jahrbuch für Philosophie und phänomenologische Forschung*, 1:685–847, 1913.

[31] L. Riley, K. Atkinson, T. Payne, and E. Black. An implemented dialogue system for inquiry and persuasion. In S. Modgil, N. Oren, and F. Toni, editors, *Proceedings, First International Workshop on Theory and Applications of Formal Argumentation (TAFA 2011)*, Barcelona, Spain, 2011.

[32] F. Santos, A. Jones, and J. Carmo. Responsibility for action in organisations: a formal model. In G. Holmström-Hintikka and R. Tuomela, editors, *Contemporary Action Theory*, volume 1, pages 333–348. Kluwer, 1997.

[33] F. Santos, A. J. Jones, and J. Carmo. Action concepts for describing organised interaction. In *Proceedings, Hawaii International Conference on System Sciences (HICSS-30)*, volume 5, pages 373–382. IEEE, 1997.

[34] P. Selinger. A survey of graphical languages. In B. Coecke, editor, *New Structures for Physics*, pages 275–337. Springer, 2011.

[35] Y. Tang, T. Norman, and S. Parsons. Agent-based dialogues to support plan execution by human teams. In *Proceedings, Second Annual Conference of the International Technology Alliance*, London UK, 2008.

[36] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany, NY, USA, 1995.