PHILOSOPHICAL TOPICS
VOL. 45, NO. 2, FALL 2017

A Theory of Practical Meaning

Carlotta Pavese

Duke University Philosophy
University of Turin, Philosophy, LLC.

ABSTRACT. This essay introduces and explores the notion of practical meaning by looking at a certain kind of procedural system—the *motor system*—that plays a central role in computational models of motor behavior. I suggest that a semantics for motor commands has to appeal to a distinctively practical kind of meaning. Defending the explanatory relevance of motor representation and of its semantic properties in a computational explanation of motor behavior, my argument suggests that practical meanings play a role in an adequate explanation of motor behavior that is based on these computational models. In the second part of this essay, I generalize and clarify the notion of practical meaning, and I defend the intelligibility of practical meanings against an important objection.

1. INTRODUCTION

Suppose that agents and, more generally, systems come, at any given time, with a fixed set of *elementary operations*.[1] An operation is elementary for a system if the system can perform it but cannot perform a proper part (Fodor (1968, p. 629)).[2] In other words, the system performs an elementary operation *directly* (or *immediately*). Call an *ability* of a system elementary at a time *t* if it is an ability to perform an operation that is elementary for that system at *t*.

---

[1] Here I am using "systems" standardly as it has been used in the philosophical literature since Dennett 1971: as anything—be it a human being, a machine, or an alien—whose behavior we are trying to explain in terms of attributions of mental properties (i.e. mental states and their content or dispositions to behavior).

2 I will clarify this definition later in the text (cfr. §3.2).

On the suppositions that systems come, at any given time, with a fixed set of elementary operations and of elementary abilities, that different systems possibly come with different sets of elementary abilities, and that even the same system may change its stock of elementary abilities through time, we may relativize an assignment of meaning to an instruction for a system at a time *t* so that the assignment takes into account that system's elementary abilities at *t*. An assignment of *practical meaning* is an assignment of meaning to an instruction that is relative in a distinctive way to systems' elementary abilities at particular times.

In the first part of this essay (§2), I introduce and illustrate the notion of practical meaning by looking at a particular procedural system (the motor system) that plays a central role in computational models of motor behavior. I argue that we need to appeal to a distinctively practical kind of meaning in order to give a satisfactory account of the meaning of the representations that, on these models, are computed by the motor system (*motor commands*). If this is correct, a satisfactory psychological explanation of motor behavior based on those computational models is one that assigns an explanatory role to practical meanings. In the second part (§3), I clarify and generalize this notion of practical meaning, I review the standard argument for the existence of elementary operations and I defend the intelligibility of practical meanings against an important objection.

2. MOTOR SYSTEMS AND PRACTICAL MEANING

2.1 MOTOR COMMANDS AND THE DENOTATIONAL MODEL

The computational approach to the study of motor behavior is one of the most successful areas of research in cognitive psychology and cognitive neuroscience. According to computational models of motor behavior, a motor task such as, for example, the task of pouring wine in a glass (Figure 1) involves a series of sensorimotor transformations that translate the intentions of the

Figure 1: Example of a motor task (morguefile.com)

agent together with visual and other sensory information about the location of the targeted objects (bottle and glass) and of the limbs into a series of *motor commands*. Such motor

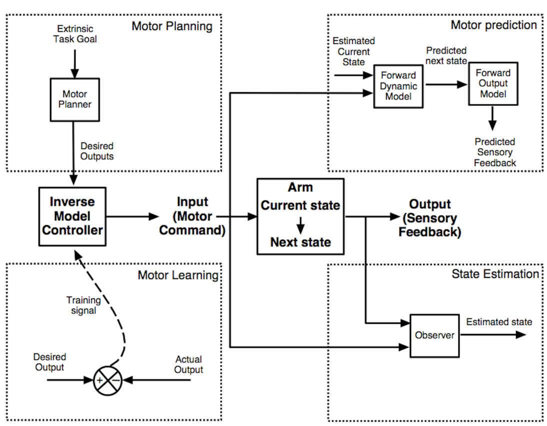

Figure 2: Wolpert's representation of the motor system (1997, 209-210)

commands are fed into the motor system that executes them sequentially to produce an output—e.g. the movement of the hand that pours the wine in the glass (Miall and Wolpert 1996, Wolpert 1997, Kawato and Wolpert 1998, Kawato 1999, Wolpert and Ghahramani 2000, Trappenberg 2009).

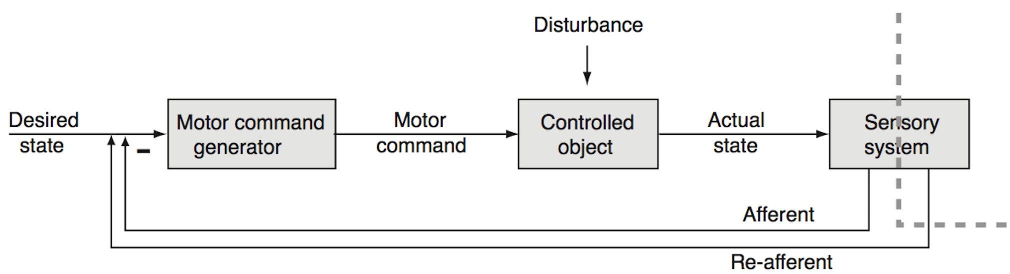

Figure 3: Trappenber's representation of the motor system (2009, 271)

Although these computational models differ in detail, they all share some important features. First of all, they all take motor commands to be the outputs of the so-called *motor planning*. Motor planning is a process by which an extrinsic task goal (cfr. Figure 2) such as a desired trajectory (cfr. Figure 3) or desired state of the arm (cfr. Figure 4) are translated into commands that can then be executed by the motor system.

Secondly, in these computational models, motor commands figure as inputs to the computation performed by the motor system that may change the state of the arm (cfr. Figure 2), align the trajectory of the hand with the agent's intentions (cfr. Figure 3), or achieve the control of an object as the agent desired (cfr. Figure 4). As made explicit by Figure 2, these outputs

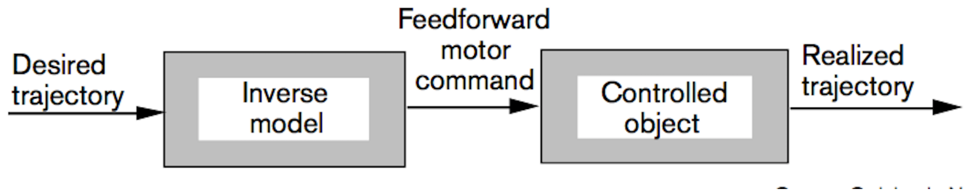

Figure 4: Kawato's representation of the motor system (1999, 719)

generate sensory feedback that is then taken into account in the generation of the next motor command. Thus, for example, in the task of lifting a can to one's lips, a desired state might be the acceleration of the hand's speed as registered by the sensory feedback, or the change of trajectory to reach the lips in response to an obstacle (Wolpert & Kawato 1998, p. 1317).

In these computational models, the role of motor commands can then be characterized as twofold:

> (i) motor commands translate desires and intentions that the agent might have into a representation that can then be interpreted and executed by the motor system;
>
> (ii) motor commands prescribe the execution of a given motor task.

At a slightly less intuitive, and slightly more abstract, level, we can say that the functional role of a motor command in these models is that of being the output of a first computational process—i.e. motor planning—and that of being the input to a second computational process—i.e. that by the motor system leading to the execution of the motor task.

Qua outputs of the motor planning and *qua* inputs to the computation by the motor system, it is plausible to take motor commands to be representations *of sort.*[3] Indeed, it is quite natural to think of motor commands as linguistic representations, on the model of programming languages' commands. However, for the purpose of this discussion, I do not want to lean on the assumption that motor commands must be linguistic. I want to allow that motor commands might be more akin to imperatival pictures such as architectural plans or road-sidewarning signs than they are to linguistic representations.[4]

Figure 5: Example of imperatival pictures

Nonetheless, it seems appropriate to at least assume that for them to play the role they are supposed to play in these computational models, motor commands must be sorts of representations. Call this assumption according to which motor commands can figure in a satisfactory computational explanations of motor behavior only if they are sorts of

---

[3] For a similar argument from the role of motor commands in motor planning to their representational nature, see also Butterfill and Sinigaglia 2014, pp. 123-124.

[4] For the notion of 'imperatival pictures', see Kjørup 1978, pp. 64-66. For examples of imperatival pictures, see Greenberg manuscript, p. 6.

representations the '*Explanatory Constraint*' for it puts a representational constraint on the explanatory power of computational approaches to motor behavior. The *Explanatory Constraint* can be justified in a variety of ways to which I will return later.[5] For the moment, I will take it as a plausible working hypothesis.

Just like imperatives in public languages such as English or like pictorial instructions such as road-sidewarning signs or architectural plans, motor commands are not representations in the sense that they are correct or incorrect, true or false.

Nonetheless, they are species of representations: they are, to use the words of psychologist and neuroscientist Tulving (1985, 387–8), "*prescriptive* representations." Qua representations, motor commands must have meaning. If so, then it makes sense to ask what their meaning is.

In order to reach a preliminary answer to this question, let me describe in some more detail the workings of the motor system. As already noted, motor commands are supposed to translate one's desires and intentions in a form that enables them to be processed by the motor system—i.e. are supposed to translate goals, desires, and intentions *procedurally* (the *procedural hypothesis*). Whose goals, desires, or intentions? Presumably, the agent's. If so, the input to the translation by the motor planning must be some content available at the personal level (the

---

[5] According to Fodor (1981), there is no computation without representation: a computation in the relevant sense is a causal chain of computer states and the links in the chain are operations on semantically interpreted formulas in a machine code. To think of any system as performing a computation is, in Fodor (1981, 180)'s words "to raise questions about the nature of the code in which it computes and the semantic properties of the symbols in the code." This Fodorian idea that computation requires representations—that is, that computation requires the states of the computation to have semantic properties—has fallen in disgrace in recent times. There are ways of understanding it on which it is clearly false (Egan 1995, Piccinini 2008, Chalmers 2011). As I explain later, my argument does not rely on this Fodorian claim.

personal level input hypothesis).[6]

Furthermore, these intentions are procedurally translated *bit by bit*—the bits being the smallest parts of the complex intentions (the *discrete hypothesis*). The smallest parts of a complex intention are *basic intentions*: an intention is *basic* if it is an intention to perform a task/ action that is basic for the agent—a task/action is basic for an agent if and only if the agent can perform it intentionally without performing intentionally any proper part (Danto, 1965). An intention is complex just in case it is not basic. Importantly, the notion of basic task/action is not the same as the notion of elementary operation. Recall that an elementary operation for a system s at a time t is one that s can perform at t but of which s cannot perform at t a proper part. By contrast, a basic task/action for a system s at a time t is one that s can perform intentionally at t, but of which s cannot intentionally perform at t a proper part. Hence, although a basic task/action does not have any other task/action as part, it might have as parts elementary operations.

According to this way of cashing out the *discrete hypothesis*, the smallest bits of the complex intention that are translated by the motor system are basic intentions.[7] Figure 6 illustrates the model: a complex intention to pour wine in a glass divides into parts—i.e. basic intentions—and each of these basic intentions is mapped into a motor command.

[6] This assumption comes to the fore when we are told that the sensory feedback that is produced by the execution of a motor task is to be "registered" by the subject, who thereby updates their intentions and feeds them again into the Motor planning. For example, Wolpert, Doya, and Kawato (2003, p. 594) write: "The motor control loop (a) involves generating motor commands that cause changes in the state of my own body. Depending on this new state and the outside world I receive sensory feedback. The social interaction loop (b) involves me generating motor commands that cause communicative signals. These signals when perceived by another person can cause changes in their internal mental state. These changes can lead to actions which are, in turn, perceived by me." Although this personal level input assumption is implicit in many descriptions of the motor system and it both simplifies and makes more perspicuous my discussion, it is not needed for my argument: it might very well be that what is translated into motor commands are desires and intentions that are not accessible at the personal level and that there is a further layer between the level of the agent and the motor system at which it makes still sense to talk of intentions.

[7] The discrete assumption may be doubted on philosophical ground. See Thompson 2008, pp. 107-8 for an argument that could be used against the discrete hypothesis. Lavin (2013) has argued against the necessity of positing basic actions. Unfortunately, I cannot consider Thompson's or Lavin's argument here. For a critical response to Thompson's argument, see Setiya 2012, pp. 288-9.

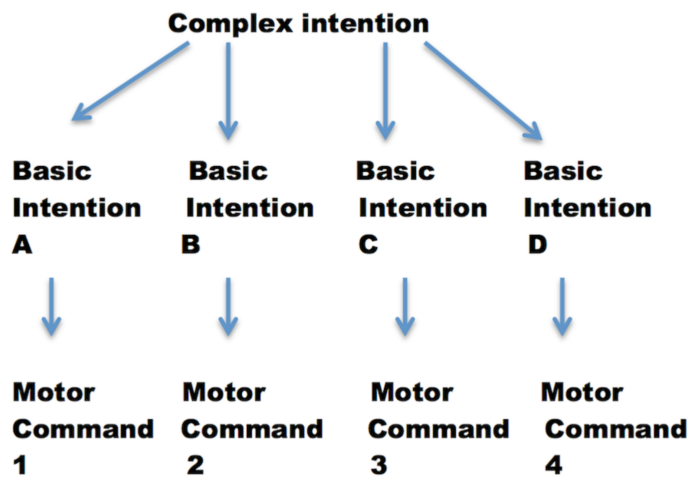

Figure 6: The Model

With these clarifications in play, combining together the procedural hypothesis, the personal level input hypothesis, and the discrete hypothesis, we can describe the working of the motor system as follows: each basic part of an agent's intention to perform a complex task is translated procedurally into a motor command. The motor system then executes the task that is prescribed by the input motor command (Figure 7: left arrow = input; right arrow = output):[8]

Figure 7: Input-output

Returning to our main question—what is the meaning of a motor command?—our

[8] This picture simplifies things a bit, for it ignores the function of the motor system that consists in taking in the sensory feedback and in responding to such feedback with the production of new motor commands. As far as I can see, this simplification is immaterial to the main argument in this section.

discussion thus far suggests that we describe the function of a motor command as issuing a prescription whose content is a task (Figure 8: large arrow = denote).

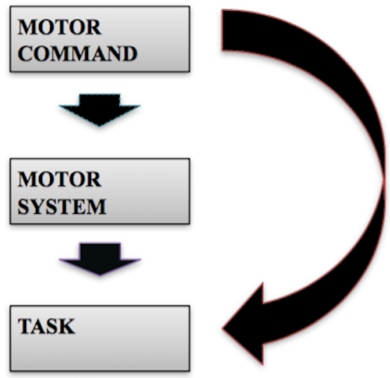

Figure 8: The Denotational Model

Call this the *denotational model* of the meaning of motor commands: according to it, the procedural translation of ones' intentions to perform a task τ is a representation—a motor command—that denotes τ. So, on this denotational model, a motor command denotes a task—i.e. the task that the motor command prescribes to the motor system.

The denotational model dovetails well with a particular approach to the semantics of imperatives that has been put forward in recent years (Lascarides and Asher 2003, Barker 2012), according to which the meaning (or denotation) of an imperative such as (1) is an *action*—i.e., (1) means, or denotes, the action of dancing and (2) denotes the action of dancing and singing:[9]

(1) Dance!

[9] This is by no means the only possible semantic treatment of imperatives in natural language, although it is the one that makes it easier for me to introduce the notion of practical meanings. See Charlow 2014b for a helpful overview on the semantics for imperatives and Charlow 2014a for possible problems with the sort of semantics I consider in the text. Still other approaches to the semantics of imperatives are the performative view (Lewis 1970), the modal approach (cfr. Grosz 2009, Kaufmann 2011), and the preference based approach (cfr. Starr ms).

(2) Dance and sing!

What is an action? As Barker (2012, 1) puts it:[10]

> Actions change the world. This means that actions can be characterized by before-and-after pictures, that is, by a picture of the world before the action is performed, and a picture of the world afterwards. Technically, then, an action will be a relation over worlds, a set whose elements are ordered pairs $<w, w_i>$ where *w* is the world before the action and w_i is the world after the action in question has been performed.

According to Barker (2012, 4), actions can be modeled as sets of ordered pairs of inputs and outputs—their inputs being possible states of the world before the action is performed and their outputs being possible states of the world that result from performing the action. Thus, for example, the meaning of an imperative such as (1) is the set of world pairs in which the second world is a continuation of the first world in which the addressee dances and the meaning of an imperative (2) is the set of world pairs in which the second world is a continuation of the first world in which the addressee dances and sings. Call this view of the meaning of imperatives '**Action Semantics**'.

Two observations about **Action Semantics**. First of all, according to it, the denotation of (1) is the action of the addressee's of dancing *in some way or other*. Thus, the denotation of (1) will encompass ordered pairs in which the second world is one where the addressee dances *tango* as well as ordered pairs in which the second world is one where the addressee dances *salsa*. This seems correct as a hypothesis about the meaning of imperatives in English: for example, (1) does not specify which particular method one is to use in order to comply with the prescription it issues.

Secondly, according to Barker's semantics, the denotation of an imperative is an action to

[10] See also Pavese 2015, 2-3.

be performed *by the addressee.* This can be modeled by making the relevant action *centered* on the addressee: instead of thinking of w and w_i simply as worlds, we think of them, following Lewis (1979), as *centered worlds* (or *situations*)—indicated as $<w, c>$ and $<w_i, c>$—where the center is the addressee at a particular time and location. The result is a semantics that more perspicuously models the role of the addressee in the denotation of an imperative such as (1) or (2): on this semantics, both a simple imperative such as (1) and a complex imperative such as (2) denote a set of ordered pairs of the form $<<w, c>, <wi, c>>$. Extending Barker's proposal to the semantics of motor commands, we may then think of a motor command as denoting the task τ that it prescribes, where a task is modeled as the set of centered world-pairs in which the second centered world is a continuation of the first centered world in which the motor system executes the task of τ-ing.

All in all, the denotational model sketched thus far is a plausible semantics for motor commands. In what follows, I will argue, however, that though partially correct, the denotational model of motor commands is incomplete: the denotation of motor commands—i.e. the motor task they denote—cannot be the only dimension that there is to their meaning.

2.2 TOWARDS A TWO-DIMENSIONAL MODEL

In a nutshell, my argument for thinking that the denotational model is incomplete goes as follows. Tasks can be performed in different ways and in accordance with different methods. Now, in these computational models, the motor planning is the process by which a task intended by the agent is translated into a motor command and by which the particular method by which a task is to be performed by the motor system is selected across a variety of different options. If motor commands are to be the outputs of this process of motor planning, they must bear record of the method by which the task they represent is to be performed. Hence, the task they denote cannot exhaust their meaning.

Let me now go through this argument carefully. The first premise is that tasks can be performed in accordance with different methods. This statement sounds like a platitude, but it is helpful, nonetheless, to draw it out with an example. Consider again the motor task that consists in moving the hand to a target location. There are a number of possible paths that the hand could move along, and for each of these paths there are an infinite number of velocity profiles (trajectories) the hand could follow. Even after having specified the hand path and velocity, each location of the hand along the path can be achieved by multiple combinations of joint angles, and each arm configuration can be achieved by many different muscle activations (Wolpert 1997, p. 2). In this sense of 'method', the same motor task can be performed by a variety of different methods.

Now, suppose that, in accordance with the denotational model, a motor command's meaning were simply its denotation and that its denotation were a task or an action that the motor system could execute by means of at least two different methods. In this case, the input provided by the motor command would be ambiguous: it would not provide all the information needed by the motor system for an unambiguous computation. This restriction suggests that if a motor command is to represent a task, for it to provide an unambiguous input to the motor system, the motor command would have to represent a task *as to be executed in accordance with a particular method*.

That motor commands also must prescribe the method by which the task is to be performed in addition to the task itself is shown in these computational models by the fact that in them, motor commands are the outputs of the process of motor planning (e.g. Wolpert 1997, Figure 2) which consists in figuring out a solution to the problem of how to perform a particular task (Figure 9, as before orange = output; blue = input).[11]

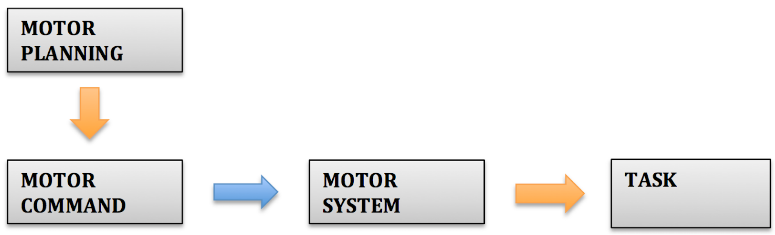

Figure 9: The role of motor planning

As Wolpert (1997, 2) puts it:

> Motor planning can be considered as the computational process that consists in selecting a single solution or pattern of behavior at the levels in the motor hierarchy, from the many alternatives which are consistent with the task.

Figure 10 (from Wolpert 1997, 3) shows the motor hierarchy. In it, the same task goal— e.g.

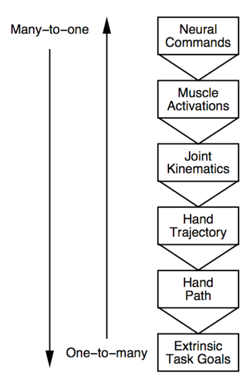

Figure 10: The Motor Hierarchy

[11] Sometimes, motor planning is called 'motor command generator'. For example, see Trappenberg 2009, 271 and Figure 3.

reaching for the glass on the table—corresponds to different paths the hand could take, which, in turn, correspond to different possible trajectories that can be executed by different movements of the joint, and these movements, in turn, correspond to different muscle activations that can be prescribed by still different neural commands.

In this sense, motor planning is the process by which a single task goal is mapped into a motor command by making a choice at each level of the motor hierarchy. If a motor command is to be the output of motor planning so conceived, then it must encode the solution to the problem tackled by the motor planning process. Hence, it must record the sort of method that the motor planning has arrived at through its selection through the motor hierarchy. This requirement suggests that a motor command does not simply denote a task but it represents a task *as to be executed in a particular way*.

So, for example, the motor task of moving the hand to a target location will be represented as to be executed along a certain path, at a certain trajectory, through a certain combination of joint angles, by different muscle activations, and so on. In this way, the fact that the motor command is the output of the motor planning ensures that its input be not ambiguous but be instead, univocal, for the instruction can now be executed only in one way.

So, the denotational model is incomplete. Motor commands do not just denote, or just represent, tasks. A specification of their meanings must include mention of the *methods* in accordance with which they prescribe that a task is to be performed. Now, as we have just seen, a method stands to a task in a many-to-one relation: the same task can be performed by more than one method. Moreover, a method is always a method to perform a specifiable task (Girard 1989, Pavese 2015, 2–5); finally, the execution of a method M outputs the task that M is a method to perform. In this sense, a method *fixes*, or *determines*, that task.[12]

Because methods stand to tasks into a many-to-one relation and can be said to determine tasks, several people have pointed out (Girard 1989, chapter 1, Moschovakis 1994, p. 17, Muskens 2005, and Pavese, 2015, 3) that methods stand to tasks as Fregean meanings (or senses) stand to their denotations (or referents). Consequently, methods are plausible candidates for being the meanings (or senses) of motor commands.

This conclusion suggests a more sophisticated picture of the semantics of motor commands. On this picture, we want to distinguish between the *denotation* (or *referent*) of a motor command—or a task—and something we might call the *meaning* (or *sense*) of a motor command (Figure 11: top right = expresses; bottom right = fixes/determines).

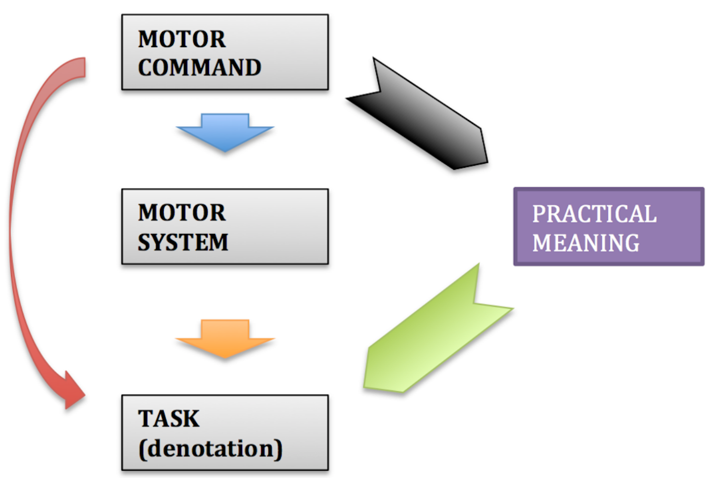

Figure 11: The Two-Dimensional Model

[12]One might think that *probabilistic methods* are a counterexample to this "determination" claim, for they enable the execution of a task *only with a certain probability of success*. However, the determination claim can still be upheld by being careful about what task it is which a probabilistic method determines or fixes: a probabilistic method for F-ing with X% probability of success *determines the task of F-ing with X% probability of success.*

On this *two-dimensional model*, a motor command has a meaning (or sense) as well as a denotation (or referent). Just like on the denotational model, on the two-dimensional model, a motor command still denotes (represents) the task that it prescribes. In addition, it prescribes (or represents) that task *as to be performed in a certain way*. This further aspect of the meaning of a motor command is captured by the two-dimensional model by a further layer of meaning: a motor command expresses a meaning (or sense), its meaning (or sense) being the method by which that task is to be performed in the sense of 'meaning-referent', or 'sense-referent' that goes back to Frege (1948).[13]

2.3 THE NEED FOR DISTINCTIVELY PRACTICAL MEANINGS

The picture is incomplete unless we clarify what a method is. So what is a method?

Different methods can be thought of *as different ways of breaking down a task into sub-tasks or sub-operations* (Pavese 2015, 2–3 and Pavese 2017). This idea can be illustrated by a couple of examples.

Consider ordering alphabetically a list of names. One method is to scroll through the whole list and move to the top of the list the items that are first in alphabetical order among the items of the whole list. Another method consists in sorting into alphabetical order every successive two members-subset of the list. These two different methods break down the problem of ordering the names alphabetically into different parts. In the first case, the main parts of the task will be (roughly): 1) scroll down the list until you find the item that comes first alphabetically; 2) move

13 If the motor command is a picture, as previously supposed, does it make sense to talk of its sense? It does. The sense or meaning of a motor command is, on my view, akin to Greenberg's (manuscript) notion of "attributive content" of a picture. For views of the meaning of picture, see Greenberg 2013, Giardino and Greenberg 2015, and Greenberg manuscript.

that item to the top; and 3) repeat the operation until the whole list is alphabetically ordered from bottom to the top. In the second case, the main parts of the same task will be instead: 1)* divide the list into every possible combination of two successive items; 2)* for any of those parts, order them alphabetically; and 3)* continue for every part of the list.

Figure 12: Directions

Consider two different sets of directions to reach a same destination X, as shown by Figure 12. The first set of directions (dotted line) breaks down the task of reaching Mole Antonelliana from Piazza Vittorio Veneto in Turin into a sequence of tasks that consists in taking via Po until one reaches the intersection with via San Massimo and then in turning right until one almost reaches via Gaudenzio Ferrari. The second set of directions (solid line) breaks down the same task into a different set of subtasks, consisting in taking via Po, in turning right on via Sant'Ottavio, taking a left on via Verdi, and then again turning right on via S. Massimo. These two sets of directions are, in the sense of 'method' explored here, two different methods, for they break down the same task—reaching the Mole Antonelliana from Piazza Vittorio Veneto—into different sequences of sub-tasks.

Finally, consider the task of calculating the sequence of Fibonacci numbers where each number in the sequence is the sum of the fib(n) for a certain input n:

0, 1, 1, 2, 3, 5, 8, 13, 21 . . . .

In order to calculate this sequence, one could use a recursive solution or a closed solution (Pavese 2015, 2–3). The task of calculating the sequence of Fibonacci numbers is broken down by a recursive method into many more parts than by a closed solution, as can be seen from the fact that, in order to calculate fib(n) for any n different from 0 or 1, a recursive method would require that one first calculate fib($n - 1$) and fib($n - 2$). By contrast, a closed solution permits calculating fib(n) directly (Abelson and Sussman 1983, Secs. 1.2.2–3).

Hence, quite generally, different methods of performing a task will break down a task into a different sequence of parts. In this sense, methods can be thought of *as different ways of breaking that task into parts*. If so, then we might think of the meaning (or sense) of a motor command as a way of breaking down the task that it denotes into subtasks. For example, the different paths the hand could take, the different possible trajectories, the different movements of the joint, and the different muscle activations will break down that task of moving a hand to a target into different sequences of operations.

We have now reached a crucial juncture in my argument. Thus far, I have shown that the denotational model is incomplete and suggested that we replace it with the two-dimensional model. According to the two-dimensional model, there are two dimensions to the meaning of a motor command—one dimension being its denotation and the other dimension being its sense. Moreover, I have argued that we think of the sense of a motor command as a way of breaking its denotation (a task) into parts. Now, the crucial juncture is this: these ways of breaking a task down into subparts must come to an end *at some point*. They cannot divide into subtasks indefinitely. That is so because a method for a system s to perform the task τ must answer the question "How can s perform τ?"—i.e. it must provide an explanation of how s can execute τ. And a satisfactory explanation of how s can execute τ must come to an end at some point—it cannot go on ad infinitum. Hence, methods cannot divide tasks into sub-operations indefinitely.[14] If methods for performing a task cannot divide it into subtasks indefinitely, their division of tasks into parts must reach a set of "elementary" subtasks—ones that have no further proper parts.

Now, either the set of elementary sub-tasks is relative to a system, or it is not relative to a system. Suppose the latter is the case—i.e. that the set of elementary sub-tasks is not relative to a system. In this case, we must suppose that a task together with a method will specify a sequence of elementary sub-tasks *absolutely*—it will specify a sequence of tasks that does not vary from system to system. The problem with this supposition is that it is not clear that the notion of an absolute elementary subtask even makes sense. An elementary task is, by definition, one that *a system* (or a set of systems which certain commonalities, cfr. Fodor (1968, p. 629)) *can perform directly, without thereby performing any other task as proper part*. So, the very notion of an

14 This argument to the effect that methods cannot divide tasks into sub-operations indefinitely closely resembles Fodor's (1968, p. 629) argument against the objection from the "proliferation of homunculi," to which I will return in section 3.2. Like Fodor's, my argument focuses on the need for a satisfactory explanation (e.g., of how a system s performs a task) to be finite. Another possible line of argument focuses on whether a finite physical system could execute a task through a method that divides it into an infinite number of subtasks. This latter line of argument is akin to one of Zeno's paradoxes on infinite divisibility. A prominent solution of Zeno's paradox (originally due to Aristotle) states that a system may be able to execute a task with infinitely many parts within a finite time, if the time of execution of each part decreases in a suitable way (cfr. Priest 1999). However, some worry that this solution is not completely satisfactory (cfr. Huggett 2010). Others still worry that this solution of Zeno's paradox of infinite divisibility is not applicable to finite systems. If it is not, then Zeno's paradox might provide yet another avenue to argue against the infinite divisibility of tasks by methods. Moreover, even if this sort of solution of Zeno's paradox is applicable to finite systems, one could argue that such a system could not perform the infinite number of subtasks within a minimal time, and argue for the need of a finite set of elementary operations on those bases. Because of the complexities surrounding Zeno's paradox, however, I choose to use Fodor's line of argument in the main text for the need of a finite set of elementary operations. Thanks to Alexis Burgess, Katrina Elliott, and Karl Schafer for helpful discussions of these issues.

elementary task makes reference to a *system*, *subject*, or *agent*. Indeed, it seems highly questionable that there could be a set of elementary tasks that is common to every system, subject, or agent—i.e. that is absolute. The very same operation may be elementary for a system at a time, and not elementary for another system at that time. Or it may not be elementary for a system at a time and become elementary for that very same system at another time. Because of this, it is not clear that we would be speaking intelligibly if we speak of a way of breaking down a task into a set of elementary operations that are common to *every* system.

Let me back up this claim, according to which what counts as an elementary operation may vary with the system and with the time, by introducing the notion of *chunking*, that plays an important role in psychological theories of motor behavior.[15] Chunking is a process by which a sequence of elementary operations gets "chunked" into parts that then can be executed as unified wholes (Verwey 1996, Verwey 2001, Sakai, Kitaguchi, Hikosaka 2003). For example, through chunking, a sequence of elementary operations A, B, C, D, E, and F can get chunked into two big parts [A, B, C] and [D, E, and F]. By definition, the chunks [A, B, C] and [D, E, and F] are now new elementary operations for the system. For through chunking, the sequence A, B and C loses, so to say, *theoretically interesting structure*: although the sequence used to have A, B and C as parts, the system has now come to execute it directly. In psychological theories of motor behavior, chunking plays an important role in explaining improvement at a motor task through practice: it is widely thought that practice makes improvement of performance possible precisely through chunking, for chunking makes the processing of a motor sequence more efficient (Verwey 2010; Verwey and al. 2011, p. 407).

[15] The label "chunking" seems to go back to Miller (1956).

If chunking is possible, then the set of elementary operations of a system must change over time, because in virtue of their lack of structure, the new chunks qualify to be included in the list of newly acquired elementary operations. Hence, it is not only plausible that what counts as an elementary operation varies across systems; it is also plausible that *the same system varies its elementary operations over time*.

If what counts as an elementary operation is relative to systems and times, and if a method is a way of breaking down a task into operations that are elementary, then methods must be relative to systems and times too. In other words, whether a way of breaking down a task into subtasks is a method for performing that task will depend on the system one considers and on its stock of elementary operations. Accordingly, because an assignment of meaning or sense to a motor command is an assignment of a method for the relevant system to perform the task, we should think of an assignment of meaning or sense to motor commands as being relative to the relevant motor system's elementary abilities.

2.4 DOES STRUCTURE OBVIATE THE NEED FOR PRACTICAL MEANINGS?

At this point, one might object to my argument as follows. Presumably, motor commands must be structured, syntactically or otherwise. At least in the case of *linguistic* motor commands, the need for structure can be argued as follows. It cannot be the case that every basic intention translates into an unstructured motor command. That is so because for that command to be processed by the motor system, it would have to be decoded, and for that purpose, a decoder would need a "look-up" table. But there are in principle an infinite number of possible basic intentions that the motor system may be asked to translate, and so the look-up table would need to include an infinite number of entries, which is not possible for a finite system.[16] Hence, the motor command that translates a basic intention must be structured out of a finite number of elementary motor commands.

Now, suppose that a basic intention is translated into a complex representation, made out of an arrangement of elementary motor commands, each corresponding to an elementary operation of the system. One might think that if motor commands are structured in this way, practical meanings are dispensable, for this complex representation can simply be assigned a structured task as its denotation—a structured task whose building blocks are operations that are in fact elementary for the system at a time.17 The structure of the motor command suffices to make its referent more fine-grained. And by making the denotation of the motor command more fine-grained, we might be able to dispense with senses. Or so one might argue.

This way of looking at things is misleading but it provides me with the opportunity to clarify my proposal. The short answer to the objection is that structure does not obviate the need for practical meanings, as I understand them. Rather, under certain plausible assumptions, practical meanings are to be identified with that structure. Let me clarify.

Even if motor commands are syntactically (or otherwise) structured, the question still arises: why does the motor planning map a certain basic intention into that particular structured representation? Part of the answer to this question must be that that particular structured

---

16 It is important to notice that this problem does not arise if a syntactically unstructured motor command is an imperatival picture, for a picture would not need for its interpretation a look-up table. But of course, although pictures are not *syntactically* structured, they may be otherwise structured—for example geometrically. See Greenberg 2013, Giardino and Greenberg 2015, and Greenberg forthcoming. Here, I am very indebted to Gabe Greenberg for discussion.

17 These structured tasks could then be modeled in different ways: as sets of ordered pairs of input centered worlds and output centered worlds where the addressee performs a certain motor task in a very particular way—through a certain particular joints displacement, a certain configuration of muscles contractions, and so on. For this purpose, we may let the output centered worlds encompass a very detailed history—whatever set of steps that is required for the motor system to go through in order for the task to be executed. Alternatively, we could let motor commands denote syntactically structured tasks, similar to linguistically structured propositions.

representation represents a task in terms of operations that the system can elementary perform. Hence, although the meaning of this complex representation can be thought of as a structured task, structured out of those elementary operations, what determines that it have as its meaning that particular structured task is the particular way in which the task is broken down by the motor planning into operations that are elementary for the system. But recall that a way of breaking down a task into operations that are elementary for the system is nothing else but a method—i.e., a practical meaning.

Now, a method determines the structure of a task that is executed in accordance with it. In this sense, a method—i.e., a way of breaking down a task into elementary parts for a system—determines a structured task. Alternatively, we might think of the method itself as the result of breaking down a task into elementary parts, in which case we want to identify the method with the very same structured task. In this sense, a practical meaning (i.e., a method) can be thought to either determine a structured task or to be identical with that structured task.

This discussion helps me clarify in what sense practical meanings are relative. A motor command and its meaning are the values of a function implemented by the motor planning—a function that takes as argument the system's stock of elementary abilities, together with the task that the agent intends to perform, and returns a complex motor representation of that task. This function is not constant: had that stock of elementary abilities been different, its value—i.e., the representation and how that representation represents the motor task—would be different. It is in this sense that the practical meaning of a motor command is relative to the system's stock of elementary abilities.

Hence, far from being dispensable, practical meanings take central stage in a view that assigns structured tasks as their meanings to motor commands. For an explanation of why a representation produced by the motor planning has a certain structured task as its meaning must appeal to the system's stock of elementary abilities. And as I have clarified, these very same structured tasks themselves can be thought of as ways of breaking down an unstructured task into elementary parts (i.e., as practical meanings) or as what is determined by those ways of breaking down the unstructured task.

One might nonetheless object that if a structured task is (or is determined by) the practical meaning of a motor command, then it is not clear that we need the motor command to have two layers of meaning, rather than just one—i.e., that we need a sense in addition to a referent. Could not we just say that the motor command's content (or denotation) is the structured task and banish altogether the level of sense?

The resulting picture would have the drawback of not vindicating a very natural way of describing the workings of the motor system. It is very natural to describe two motor systems as executing the same motor task, coarsely individuated, even though the methods that they employ are widely different; and it is natural to speak as if two motor commands may prescribe the same motor task, even though in different ways. Consequently, it is natural to follow **Action Semantics** in taking coarsely individuated tasks to be what both syntactically simple commands (such as (1)) and syntactically complex commands (such as (2)) prescribe and in identifying these coarse-grained tasks with the motor commands' denotation. The required fineness of grain is then reached by adding a further layer of meaning—a sense in addition to the denotation.

The idea that motor representations denote actions or action outcomes (as opposed, for example, to methods for performing those actions and as opposed even to very fine-grained actions that include bodily movements and muscle contractions) is actually very popular in the cognitive sciences (Rizzolatti et al. 2001, Gallese 2000, Gales and Metzinger 2003, Butterfill and

Sinigaglia 2014, Levy 2016). Gallese and Metzinger (2003, 372) distinguish between BODILY MOVEMENTS—which are "simple physical events, and they can be represented accordingly;" BEHAVIORS—which are "movements that are goal-directed, i.e. which can meaningfully be described as directed towards a set of satisfaction conditions, but without necessarily being linked to an explicit and conscious representation of such conditions;" and ACTIONS—which are a "specific subset of goal-directed movements: a series of movements that are functionally integrated with a currently active goal representation as leading to a reward constitute an action." Gallese and Metzinger (2003, 383) argue that motor representations are representations of actions *in this latter sense*. Butterfill and Sinigaglia (2014, 120–1) concur; they argue that there must be motor representations of actions and action outcomes, as opposed to representations of joint displacements, because the same joint displacements may be involved in different actions marked by the presence or absence of a targeted object and because the same action outcome may correspond to several variations of the kinematic and dynamic features of the action. Relying on a variety of psychological and neuroscientific studies, Butterfill and Sinigaglia (2014) claim that there is evidence that some markers of motor processing are correlated with action outcomes (extrinsically individuated) rather than narrowly individuated kinematic or dynamic features of an action:

> For any given marker of motor processing (such as a pattern of neuronal discharge or motor-evoked potentials), how can we test whether that marker carries information about action outcomes? The basic principle is straightforward: vary kinematic and dynamic features while holding constant an action outcome; and, conversely, vary action outcomes while holding kinematic and dynamic features constant. In practice researchers have devised many ingenious ways to achieve this. In order to vary kinematic and dynamic features while holding action outcomes constant, in some studies a single action outcome is achieved using different effectors, hand, mouth or foot, say (Rizzolati *et al.* 1988, 2001; Cattaneo *et ai* 2010). A variation on this approach is to contrast performing a grasping action with different tools, so that the same action outcome might require closing or opening the hand depending on the tool used (Umhta *et al* 2008; Cattaneo *et*

> *al.* 2009; Rochat *et al.* 2010). In order to vary action outcome while holding kinematic and dynamic features constant, researchers have contrasted grasping movements with different distal outcomes such as eating and placing (Fogassi *et al.* 2005; Bonini *et al.* 2010; Cattaneo *et al.* 2007). Another approach is to contrast the same grasp-ing movements performed in the presence or manifest absence of a target object (Umilta *et al.* 2001; Vilhger *et al.* 2010). A related alternative is to contrast the same grasping movements in the presence of objects which could, or manifestly could not, be grasped by means of such movements (Koch *et al.* 2010). In each of these cases there is evidence that some markers of motor processing are correlated with action outcomes rather than narrowly kinematic or dynamic features of action.

Thus, the hypothesis that motor representations denote action outcomes, and that these may come apart from narrowly kinematic or dynamic features of the action, is very popular. If we are to vindicate it, we are better off letting actions and tasks (coarsely individuated) be the referents of motor commands and assigning to methods or structured tasks the role of motor commands' senses.

One might nonetheless insist that we could let motor commands have two different kinds of denotations, one corresponding to the more fine-grained (structured) task and one corresponding to the coarse-grained (unstructured) task. On this "multiple-denotations" picture, senses are not needed. This picture would also vindicate the intuition that two different motor commands might have the same content, in some sense of 'content', in that they might prescribe the same (coarse-grained) motor task. And so this picture would overcome my main objection to dispensing altogether with the level of sense.

The chief problem with the current proposal is that it fails to do justice to the different theoretical roles of the two layers of meanings that are being posited. In describing thwe more fine-grained level of meaning as a sense, we highlight that it stands in a sense-to-reference relation to the coarse-grained level of meaning—in that not only does it stand in a many-to-one relation to it but it also determines it. This theoretical role of the fine-grained level of meaning seems to me obliterated on the multiple-denotations picture.

Even if one is convinced that something like the two-dimensional model is correct, one might still quibble on whether an assignment of meaning to a motor command must be an assignment of practical meaning—i.e., on whether the assignment must be thought to be relative to the system's set of elementary operations. My argument to this effect consisted in pointing out that what counts as a method depends on the system's set of elementary operations, for methods are ways of breaking down a task into operations the system can elementarily perform, and those may vary from system to system. But it is an assumption of this argument that an assignment of meaning to a motor command for a system s must be an assignment of a method for s to perform the relevant task. One might think that this assumption should not be granted: could not such an assignment be an assignment of *some* way of decomposing the task, even if that way of decomposing the task does not constitute a method for the system to perform that task, as it does not break down the task into parts that are elementary for the system?

In response, note that such an assignment of meaning would not capture what we have seen to be the motor command's functional role. The functional role of a motor command is that of representing a task in a particular way—in terms of operations that the system can elementary perform. That is the function played by a motor command within the motor system. To the extent to which we think that at least some level of meanings of mental representations must capture their functional role (cfr. Field 1977, Block 1998) and to the extent to which we think that what a representation represents and how it does depend on the function of that representation within the systems that produce or use it (cfr. Millikan 1984, Neander 1995, 2006), we should recognize the theoretical benefits of a semantics that assigns to motor commands methods (i.e., practical meanings) as their meanings, rather than any other way of decomposing a task into subparts.

In this section, I have argued as follows. The structure of motor commands does not allow us to dispense with practical meanings. The functional role of motor commands is to represent a task in terms of the system's elementary operations. And practical meanings are just that: ways of breaking down a task into parts that a system can elementarily perform. We might think of these ways as structured tasks themselves or as what determines these structured tasks; what makes these structured tasks practical meanings is that they are the outputs of a function taking as argument the system's stock of elementary abilities—a function implemented by the motor planning. Hence, their relativity.

In addition to practical meanings, I have argued that motor commands have another layer of meaning—i.e., they in addition denote coarse-grained motor tasks. Because practical meanings stand to those tasks in a sense-reference relation and because practical meanings accurately describe the functional role of motor commands, it seems to me that the two-dimensional model proposed here stands out as the most plausible model for the semantics of motor commands.

2.5 THE EXPLANATORY CONSTRAINT

In conclusion, the two-dimensional model proposed here provides a more natural answer to the question "What is the meaning of motor commands?" than its obvious alternatives. To summarize, my argument went as follows. According to the *Explanatory Constraint*, if motor commands are to play a role in satisfactory computational explanations of motor behavior, then motor commands must be representations, albeit prescriptive ones. *Qua* representations, they must have semantic properties. Hence, it makes sense to ask what their meaning is. I suggested that a motor command represents a task as to be performed in a particular way—i.e., *as to be* *performed in accordance with a certain method*. This conclusion suggests that the two-dimensional model is correct: a motor command denotes a task *and* expresses a meaning (or sense) that determines that task—their meaning being a method for performing that task. But methods are ways of breaking down a task into parts. And this structure must bottom out at some point—it must reach a set of elementary operations. Because this notion of a method and elementary operations is relative, assignments of meanings to motor commands must be relative to the relevant system's set of elementary abilities. An assignment of meaning to a command that is relative in this way to a set of elementary abilities is an assignment of *practical meaning*. In virtue of its practical meaning, relative to a system, a motor command represents a task in terms of operations that are elementary for that system.

My argument relied on the *Explanatory Constraint*. It is time to defend it. According to the Explanatory Constraint, motor commands ought to be representations and as such they ought to have content for them to play a role in a satisfactory psychological account of motor behavior. The *Explanatory Constraint* may seem problematic for it appears to rely on Fodor (1981)'s claim that there is no computation without representation. And this Fodorian claim has been widely criticized. Several people have pointed out that computation can be given a purely formal characterization. Egan (1995) has argued in favor of a purely formal and mathematical individuation of computational states and of computation. Piccinini (2008) has argued in favor of a functionalist understanding of computation that does not invoke any semantic properties. Finally, Chalmers (2011) proposes a causal individuation of states implementing a computation, and he points out that such implementing states may or may not have semantic properties.

I am sympathetic to these criticisms of the (unqualified) Fodorian claim that computation (*qua* computation!) requires representation. But let me emphasize that the *Explanatory*

Constraint is independent of this Fodorian claim. One might agree that computation can be given a purely formal characterization yet also insist that cognitive systems compute representations (Newell and Simon 1976, Fodor and Pylyshyn 1988, Peacocke 1995, Peacocke 1999, Rescorla 2012, Rescorla 2014) on the ground that only computation over *representations* suffices for cognition and mentality. Or one might follow Chalmers (2011) in taking computational psychological explanations to involve the physical states that implement a computation, formally characterized, and at the same time letting these physical states to have semantic properties.

To my mind, both of these approaches offer valuable ways of motivating and understanding the *Explanatory Constraint*. On these two ways of understanding the *Explanatory Constraint*, however, the question does arise: why think of the motor system as a *cognitive* system—i.e. as something whose output has mental and semantic properties?

The answer to this question is that psychological explanations where the motor system features are supposed to be explanations of *actions*—or tasks. Actions or tasks cannot be explained simply as the outcome of a purely formally characterized computation. Actions are *physical events* that are *goal-directed*. As such, they are intentional under some description. (Recall Gallese and Metzinger's 2003 characterization of the output of the motor system as actions mentioned above; they argue, "specific subset of goal-directed movements [are] a series of movements that are functionally integrated with a currently active goal representation as leading to a reward constitute an action.") Hence, no purely formal characterization of the computation that outputs them can explain them, for it would miss out on their semantic properties. Thus, it is because of the nature of the outputs of the motor system that psychological explanations of these outputs need to appeal to the semantic properties of the states involved in the explanation. Motor commands must be representations, thus have meaning, if they can be employed in an adequate psychological explanation of its intended *explananda*.

3. THE INTELLIGIBILITY OF PRACTICAL MEANINGS

3.1 AN EXAMPLE OF PRACTICAL MEANINGS

In the last section, I have argued that computational explanations of motor behavior must appeal to practical meanings, for the best account of the meaning of motor representations is in terms of practical meanings. My argument consisted in pointing out that the functional role of motor commands in computational models of the motor system is that of representing a motor task in terms of the operations that the motor system can elementarily perform. And these ways of representing tasks in terms of operations that the motor system can elementarily perform are nothing else but practical meanings.

According to this definition, the notion of practical meaning is general and not restricted to representations of motor tasks. The definition can then be extended to representations of tasks such as adding and multiplying or of any other task, including those that are elementary for a system (in this case, a practical meaning is a way of breaking the task into *no parts*). Because of this generality, practical meanings promise to play a role in computational explanations of tasks that involve procedural systems other than the motor system (cfr. Pavese 2018).

In Pavese 2015, 6–9, I have argued that operational approaches to the semantics of programming languages, famously proposed and developed by Plotkin (1981, 2004), are an example of assignments of practical meanings to programming texts. What is distinctive about these sorts of semantics is that they assign to linguistic instructions (such as programming texts) complex semantic values that describe a task in terms of operations that a system can elementarily perform. Hence, they potentially assign different semantic values to the same linguistic instruction depending on the relevant set of elementary abilities.

For example, consider the following piece of programming text:

PROGRAM TEXT

MULT(n,m)

Suppose a system's elementary operations include addition (ADD), assignment (SET v TO v^{l}), and sequencing (;). If so, then an operational semantics will assign PROGRAM TEXT a complex semantic value that describes multiplication in terms of those operations, as **R-MULT** does:

$$\textbf{R-MULT=}\quad \begin{cases} \text{if } \langle m{=}0, f\rangle \Longrightarrow \mathrm{T}, \\ \langle \mathrm{MULT}(n, m), f\rangle \longrightarrow \langle res, f[0/res]\rangle; \\ \text{if } \langle m{=}0, f\rangle \Longrightarrow \mathrm{F}, \\ \langle \mathrm{MULT}(n, m), f\rangle \longrightarrow \langle \mathrm{SET}\ v\ \text{to}\ \mathrm{MULT}(n, m\text{-}\ 1);\mathrm{SET}\ res\,\text{to}\,\mathrm{ADD}(n, v);res, f^{*}\rangle \end{cases}$$

where $f^{*} = f[n \times (m - 1)/v][n + (n \times (m - 1))/res]$.

R-MULT is an *update function*—i.e. a function that maps a configuration of a system into another configuration[18] that results from updating the input configuration by assigning 0 to the result of multiplying n by m, if m is 0 (**R-MULT1**), or else by assigning the value of v to the

[18] Configurations are indicated by an ordered pair of an instruction and an assignment function f assigning values to variables in the instruction—e.g. $\langle$MULT (n, m), $f\rangle$.

result *res*, after performing a series of additions of *n* to itself *m* − 1 times (**R-MULT2**).[19]

Consider now a different system, whose set of elementary abilities only includes the operation **ADDn** that updates a configuration in such a way to assign to the variable *res* the result of correctly adding *n* to itself *m* – 1 times in accordance with the rule **R-ADDn**:

$$\textbf{R-ADDn} = \langle ADD_n(m), f\rangle \longrightarrow \langle res, f^*\rangle \text{ where } f^* = f[n_0 + \ldots + n_{m-1}/res]$$

Then an operational semantics will assign PROGRAM TEXT the following operational semantic value that describes multiplication in terms of ADDn (*m*):

$$\textbf{R-MULT*} = \text{If } \langle ADD_n(m), f\rangle \longrightarrow \langle res, f^*\rangle, \text{ then } \langle MULT(n, m), f\rangle \longrightarrow \langle res, f^*\rangle$$

By possibly assigning different operational semantic values to the same instruction depending on a system's elementary abilities, this sort of semantics illustrates the relativity of practical meanings.

Let me highlight a further feature of this semantics. Consider again R-MULT. R-MULT is a complex update function that is composed out of simpler update functions (ADD, SET *v* TO v1, and ;) that the system has to compute in order to multiply two numbers. So, in effect, such an operational semantic value is a way of breaking down multiplication into sub-tasks that the system can elementarily perform. Hence, operational semantic values are structured in such a way that, relative to a stock of elementary operations, the operational semantic value of a complex instruction is function of the operational semantic values of simpler instructions

---

[19] In Pavese 2015, 7-8, I characterized operational semantic values as inference rules. That way of thinking of operational semantic values is encouraged by their employment in proof systems for proving certain structural and semantic features of programs. Here, I describe them instead as update functions from configurations to configurations. There is no contradiction, as inference rules can themselves be thought of as update functions of sort. See Pavese 2016.

together with the structure of those complex instructions. By satisfying the conditions of structure and relativity, operational semantics for programming languages provides an example of an assignment of practical meanings.[20]

3.2 ELEMENTARY ABILITIES

My characterization of practical meanings essentially appeals to the notion of elementary abilities. But is positing such elementary abilities legitimate?

Fodor (1968) invoked elementary operations and elementary abilities in a defense of the intelligibility of computational explanations of behavior against the *proliferation of homunculi* objection. According to this objection, a computational explanation of, for instance, how we tie our shoes that appeals to sub-systems that do it for us is problematic, for it invites the further question: "How do those subsystems do it?," leading to positing a further layer of subsystems, and so on *ad infinitum*. Fodor (1968, p. 629) responded to this sort of objection that the proliferation of homunculi is to be stopped by positing elementary operations, where an elementary operation is one with "no theoretically relevant internal structure," that a system performs "in no way at all" and for which "certain kinds of 'how to' questions cannot arise about it:"

> […]If every operation of the nervous system is identical with some sequence of elementary operations, we get around proliferating little men by constraining a completed psychological theory to be written in sequences of elementary instructions (or, of course, in abbreviations of such sequences).

[20] It is an important question whether ways of breaking down a task into parts are fully, in part, or not at all combinatorial. The question is important because abilities to perform motor tasks do not seem to be productive in the way one would expect they would be, were practical meanings to be fully combinatorial. This issue will have to wait future discussion (cfr. Pavese 2018).

Following Fodor, elementary operations and elementary abilities are theoretical posits that are needed if we are to make sense of the possibility of computational explanations of behavior. Because of this, it seems appropriate to appeal to such theoretical posits, if our goal is to provide a theory of the meaning of motor commands *as they figure in computational models of explanation*. Hence, it seems appropriate to appeal to elementary abilities in our semantics of motor commands.

Given that we need elementary operations in order to stop the proliferations of homunculi/ sub-systems and hence in order to stop an infinite regress, a second question arises: what sorts of abilities must a system's set of elementary abilities include in order for them to play this theoretical role?

Suppose the system can elementarily perform A and can elementarily perform B but does not know how to perform A and B in a sequence. In this case, again, a regress is triggered: the system would need to be told how to combine A and B and, if that method itself has parts, the system will need to know how to combine those parts in order to combine A and B. And so on. In order to stop this new sort of regress, we must add, to the elementary operations of a system, its *primitive modes of combination*. These modes of combination have a structure by which elementary tasks of the system are combined into complex ones. For example, computer's primitive modes of combination usually include *sequencing*, as well as *loops*, *if-then* commands, and *while*-commands:[21]

(i) *sequence*: execute A; B; by executing A; then executing B;

[21] The goal of this list is just provide an example of primitive modes of combination. I am not claiming that this list exhausts the primitive modes of combinations for humans' motor systems. Nor am I claiming that the set of primitive modes of combinations for motor systems is fixed once and for all.

(ii) *loop*: execute A; B; by executing A; then executing B;

(iii) *if-command*: execute A if C obtains;

(iv) *while-command*: execute A while executing B.

Accordingly, we should think of methods as ways of breaking down tasks into parts that are elementary for a system and that are combinable in accordance with that system's primitive modes of combinations. I call the set of abilities to perform elementary tasks and to combine elementary tasks in accordance with a system's primitive modes of combinations a system's set of *primitive* abilities. Only methods thus conceived are plausible candidates for being practical meanings.

A final complexity should be noted. Recall chunking, the process by which sequences of elementary operations (such as A, B, C) are chunked, to produce a new elementary operation (the chunked operation [A, B, C]). This is a new elementary operation for the system, as the system can now execute it directly, having its parts lost computational significance. Note, however, that that does not necessarily mean that the parts of the sequence A, B, and C stop being elementary for the system once the sequence is chunked, for the system can still execute, say, A in isolation. If so, then now consider the command to execute the sequence A, B, C, D. It seems that there is more than one possible way of decomposing it into elementary operation for the system. For example, both [A, B, C][D] and [A][B][C][D] count (assuming that D is also elementary). But then the task of executing the sequence A, B, C, and D can now be decomposed into more than one way, compatibly with the same set of elementary operations. In this case, the system's set of elementary operations does not guarantee a unique decomposition of a task into elementary operations. Hence, without further restriction on the assignment of practical meaning, we are left with the possibility that there might be more than one practical meaning for any given motor command, given the same set of elementary abilities.

This problem can be overcome by imposing that a practical meaning of a command for a system s at a time t be not just any way of decomposing a task into parts that are elementary for s at t. It must be the most efficient (for s at t) way of decomposing the task into parts.[22] An assignment of practical meaning to a motor command c that procedurally translates the task τ is an assignment to c of the most efficient (for s at t) way of decomposing τ into parts that s can elementarily perform at t.

3.3 MODELING PRACTICAL MEANINGS

Practical meanings are ways of breaking down a task in terms of parts that a system can elementary perform and puts together in accordance with primitive modes of combination. But how can these ways of breaking down a task be modeled formally?

By analogy with operational semantic values, we could model them as *complex update functions*—composed out of simpler update functions—in which the output configuration results from updating the input configurations with a series of operations. Such update functions are to be finely individuated in that their identity depends not only on their inputs and outputs but also on their structure—i.e., on how they are composed out of simpler update functions.

An *intensionalist* understanding of functions is particularly suited for our purposes. According to *extensionalists*, functions are just sets of ordered pairs of inputs and outputs. According to *intensionalists*, functions' types are to be individuated more finely in terms of their

22 In case of ties—i.e., if there is more than one most efficient way of decomposing a task into parts—we may suppose that the assignment randomly chooses one among the ties.

structure—i.e., in terms of how they describe the way the output is to be reached. For example, the function $x + 2$ and the function $x + 1 + 1$ are different, even though they are extensionally equivalent (Church 1940; Church 1973; Church 1974), for their structure is different: while the former is identical to the operation of adding 2, the latter is composed out of two successive operations of adding 1. That holds for operational semantic values too: **R-MULT** is a different operational semantic value from **R-MULT***, for it is composed out of different update functions. Understood as such, ways of describing a task in terms of its parts record the task's structure in the way desired.

To sum up: an assignment of practical meanings will assign an instruction a way of breaking down a task in terms of subtasks that a system can elementarily perform and that can put together in accordance with a system's primitive modes of combination. Practical meanings can be formally modeled as update functions, provided that we construe these update functions in sufficiently fine-grained fashion and not simply as sets of input and output conditions.

3.4. TWO CONSTRUALS OF PRACTICAL MEANINGS AND THE PROBLEM OF UNDERSTANDING

The relativity of an assignment of practical meaning to sets of primitive abilities can be conceived of in two different ways.

We may think of this set of primitive abilities as if it were a sort of *context* with respect to which a linguistic instruction can be semantically interpreted in a way analogous to how, in a Kaplanian semantics, sentences are assigned truth-conditions relative to a context.[23] In

[23] Cfr. Kaplan 1989.

determining the practical meaning of an instruction the set of primitive abilities plays, on this construal, a role analogous to that of a Kaplanian context (Kaplan 1979, 1989). This reading would amount to a *context-relative* construal of practical meanings.

According to a *content-relative* construal of practical meanings, on the other hand, the set of a system's primitive abilities is not a context relative to which practical meanings are assigned. Instead, such a set of primitive abilities should be thought of as sorting practical meanings into *types*. What does that mean? Consider again the case of operational semantic values. As noted, operational semantic values can be thought of as update functions, and I gave some reasons in favor of an intensionalist construal of these update functions. But even among intensionalists about functions, there might be disagreement as to how finely functions are to be individuated. Responses to this question vary but it is not outlandish to think that functions should be typed by the set of primitive abilities that would be needed to compute them. In fact, it is quite usual among theoretical computer scientists to type functions on the basis of the number of steps that a machine would take to compute them (cfr. Girard 1989). Because the number of steps will vary as a function of a machine's primitive abilities, this way of typing would amount to individuating functions by sets of primitive abilities.

I want to be neutral between the context-relative and the content-relative construal of practical meanings. But let me point out that the two construals are quite substantially different. A way to highlight their difference is by looking at how these two construals deal with a possible objection to the intelligibility of practical meanings.

In order to introduce the objection, let me consider an argument in Pavese 2015, according to which what makes practical meanings distinctively practical is that one cannot understand them in the relevant sense without being endowed with the *ability* to perform a certain task. For example, a system cannot understand **R-MULT** without being endowed with the ability to multiply two numbers, precisely because such a rule breaks down the task into parts that a system can primitively perform.

One might object to this claim. Is it really true that a system cannot understand practical meanings without acquiring the ability to perform the relevant task? Take, for instance, operational semantic values that I claimed to be examples of practical meanings. Programmers use operational semantic values all the time, so they, plausibly, understand them. Yet, they are not necessarily enabled to perform the same task that computers they program can perform. So, one might object on this ground to the intelligibility of practical meanings.

The response to this objection depends on one's preferred construal of practical meanings. First, consider the context-relative view. One adopting this construal of practical meanings can respond that although programmers can understand operational semantic values without being endowed with the ability to perform the corresponding task, they cannot *practically* understand operational semantic values—cannot understand-them-under-a-certain-set-of-primitive-abilities—without being endowed with that ability. Consider the analogy with a Kaplanian semantics, which assigns a sentence "I am Italian" a certain proposition *that x is Italian* for a given contextual assignment of the speaker to x. Suppose the relevant proposition is the proposition *that Giorgio is Italian*. One may understand that same proposition without being the speaker of the context, as when Ale thinks of Giorgio's nationality. But one cannot understand-it-under-the-character-of-the-context unless one is the speaker of the context. The same holds for this context-relative understanding of practical meanings: on this construal, practical meanings are semantic values of instructions that can be understood by systems which do not have the relevant set of primitive abilities but that cannot be understood-under-those-primitive-abilities—in other words, they cannot be *practically understood*—by systems that do not have a certain set of primitive abilities.[24] However, *if* they are so practically grasped, they do endow one with the ability to perform the corresponding task.

The relativist's response is different. A relativist will deny that, in the circumstances envisaged, programmers necessarily understand the same function computed by the machines. For according to the content-relative view of practical meanings, the type of the function will be different if the set of primitive abilities is different because the type of the function itself is determined by that set. If so, then although certainly programmers grasp *some* function when they manipulate operational semantic values, they do not necessarily grasp the *same* function that is computed by the machines they program. By individuating functions finely in the way described above, we get a sort of content that is relative to sets of primitive abilities and is fine-grained enough to guarantee that if one can understand that content at all, then one is endowed with the ability to perform the relevant task. Given these identity conditions, one can understand practical meanings at all only if one "practically" understands them.

4. CONCLUSIONS AND OPEN ISSUES

Motor commands figure prominently in computational explanations of motor behavior. Following Tulving (1985, 387–8), we can think of motor commands as *prescriptive representations*. That raises the question: What is the best way of thinking of the meaning of these prescriptive representations?

---

[24] In this sense, my account of practical meaning, when understood in this context-relative sense, provides a more rigorous characterization of Stanley and Williamson (2001) and Stanley's (2011) notion of practical modes of presentation.

In the first part of this essay, I have argued that, given the functional role of motor commands in computational models of motor behavior, an adequate semantics for motor commands is two-dimensional. In addition to denoting tasks, motor commands have a further dimension to their meaning, if they have to play the functional role that these computational models assign to them—if they are to be the output of motor planning and if they are to prescribe to the motor system the execution of the task. I argued that the right way to think of this further dimension of their meaning is in terms of *practical meanings*. That means that motor commands represent motor tasks in terms of operations that the system can elementarily perform and can be put together through its primitive modes of combinations. Hence, by assigning a central role to motor commands and motor representations, complete psychological explanations of motor behavior based on the computational models described here must invoke practical meanings.

In the second part of this essay, I clarified this notion of practical meaning, I generalized it, and I defended it against an important objection. Having defended the explanatory relevance of semantic properties to a computationalist explanation of motor behavior (the *Explanatory Constraint*), my argument concludes that a psychological theory of motor competences based on the computational models described here must assign a central explanatory role to practical meanings.

Several issues are left open for future discussion (cfr. Pavese 2018). One is the following: one might wonder whether *motor commands* and *motor representations* themselves are really needed in an explanation of motor behavior. As we have seen, computational models of motor behavior do posit motor commands and so do posit motor representations. But could not we envisage computational models of motor behavior that assign *no role* to motor commands and to motor representations? On such models, presumably, the explicit representation of the agent's intentions would be processed by the motor system into the execution of the action without producing a motor representation as an intermediary step.

Other philosophers have recently defended the need for motor representation, over and above the explicit representation of the subject's intentions, for a satisfactory explanation of motor behavior (Butterfill and Sinigaglia 2014, Levy 2016, Fridland 2017). An exhaustive review of this literature is beyond the scope of this essay. Let me just mention what I take to be the most promising line of argument in support of the indispensability of motor representation. Motor representation could be shown to be indispensable in an explanation of motor behavior if instances were observable of goal-directed yet involuntary motor behavior. In this case, an explanation of the goal-directedness of the motor behavior would demand positing a representation of the motor goal that is, nonetheless, distinct from the agent's intentions and volitions. Recent empirical work on motor skill suggests that we do observe instances of dissociation between an agent's intentions and goal-directed motor behavior (Mazzoni and Krakauer 2006). I leave a more careful discussion of these empirical findings for future work (cfr. Pavese 2018). I also leave it to future work to discuss whether practical meanings could appear as components of thoughts,[25] whether and to which extent they are combinatorial and productive, and to discuss their role in an account of the interaction between procedural systems and declarative systems—that is, in a solution of Butterfill and Sinigaglia (2014)'s *interface problem*.

ACKNOWLEDGMENTS

[25] At least on some conception of what propositions are, like on the Fregean conception. Cfr. Gallese and Lakoff (2005), Pavese (2015, 16-17), and Pavese 2017.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 609402 – 2020 researchers: Train to Move (T2M). For comments, thanks go to the audiences at the Metaphysics Through Time workshop at UCL and at the Mind, Metaphysics, and Psychology research seminar at King's College London, at the philosophy of language reading group at University of Edinburgh, at the Workshop "Meaning beyond truth conditions: Practical Meaning, Prejudice, and Language", at Humboldt University, and at the California Philosophy Workshop for comments. I am grateful to Ellen Fridland, Ephraim Glick, Jeremy Goodman, Harvey Lederman, Karl Schafer, Kevin Lande, Sam Cumming, Barbara Vetter, Ian Jakeway, Katrina Elliott, Diego Marconi, Bryan Pickel, Brian Rabern, Anders Schoubye, Will Starr, and Wolfgang Schwarz for comments. Finally, conversations with Josh Armstrong, Guillermo Del Pinal, Gabe Greenberg, and Alexis Burgess were immensely helpful.

REFERENCES

Abelson, H. and Sussman, G. J. (1983). Structure and Interpretation of Computer Programs. The MIT Press, Cambridge, Mass.

Barker, C. (2012). Imperatives denote actions. In *Proceedings of* Sinn *und Bedeutung*, volume 16, pages 57–70. Citeseer.

Block, N. (1998). Conceptual Role Semantics. In Edward Craig (ed.), Routledge Encyclopedia of Philosophy. Routledge. pp. 242-256.

Butterfill, S. A. and Sinigaglia, C. (2014). Intention and motor representation in purposive action. *Philosophy and Phenomenological Research*, 88(1): 119–145.

Chalmers, D. J. (2011). A computational foundation for the study of cognition. *Journal of Cognitive Science*, 12(4): 323–357.

Charlow, N. (2014a). Logic and Semantics For Imperatives. *Journal of Philosophical Logic*, 43(4):617–664.

Charlow, N. (2014b). The meaning of imperatives. *Philosophy Compass*, 9(8):540–555.

Church, A. (1940). A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5(02):56–68.

Church, A. (1973). Outline of a revised formulation of the logic of sense and denotation (part i). *Noˆus*, pages 24–33.

Church, A. (1974). Outline of a revised formulation of the logic of sense and denotation (part ii). *Noˆus*, pages 135–156.

Danto, A. C. (1965). Basic actions. *American Philosophical Quarterly*, 2(2):141–148.

Dennett, D. C. (1971). Intentional systems. *The Journal of Philosophy*, 68(4):87–106.

Egan, F. (1995). Computation and content. *The Philosophical Review*, 104(2):181–203.

Field, H. (1977). Logic, meaning, and conceptual role. The Journal of Philosophy, 74(7): 379-409.

Fridland, E. (2017). Skill and motor control: intelligence all the way down, *Synthese*, 174(6): 1539–1560.

Fodor, J. (1968). The appeal to tacit knowledge in psychological explanation. *The Journal of Philosophy* 65.20: 627–640.

Fodor, J. (1981). The mind-body problem. In Heil, J., editor, Philosophy of Mind: A Guide and Anthology, pages 168–182. Oxford University Press, Oxford.

Fodor, J. A. and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71.

Frege, G. (1948). Sense and Reference. *The Philosophical Review*, 57(3):209–30.

Gallese, V. (2000). The inner sense of action. agency and motor representations. *Journal of Consciousness Studies*, 7(10):23–40.

Gallese, V. and Lakoff, G. (2005). The brain's concepts: The role of the sensory-motor system in conceptual knowledge. *Cognitive neuropsychology*, 22(3–4):455–479.

Gallese, V. and Metzinger, T. (2003). Motor ontology: the representational reality of goals, actions and selves. *Philosophical Psychology*, 16(3):365–388.

Giardino, V. and Greenberg, G. (2015). Introduction: Varieties of Iconicity.

Review of Philosophy and Psychology 1.6: 1-25.

Girard, J.-Y. (1989). Proofs and Types. Cambridge University Press, Harvard.

Greenberg, G. (2013). Beyond resemblance. Philosophical Review 122.2: 215-287.

Greenberg, G. (2017) Content and Target in Pictorial Representation, manuscript.

Grosz, P. (2009). German particles, modality, and the semantics of imperatives.

Huggett, N. (2010). Zeno's Paradoxes, The Stanford Encyclopedia of Philosophy (Winter Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/win2010/entries/paradox-zeno/>.

Kaplan, D. (1979) On the logic of demonstratives. *Journal of philosophical logic* 8.1: 81–98.

Kaplan, D. (1989). Demonstratives. In Themes from Kaplan, ed. by Joseph Almog, John Perry and Howard Wettstein, pages 481–563.

Kaufmann, M. (2011). Interpreting imperatives, volume 88. Springer Science & Business Media.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727.

Kawato, M. and Wolpert, D. (1998). Internal models for motor control. Sensory Guidance of Movement, 218:291–307.

Kjørup, S. (1978) Pictorial speech acts. *Erkenntnis* 12.1: 55–71.

Lascarides, A. and Asher, N. (2003). Imperatives in dialogue. Pragmatics and Beyond New Series, pages 1–24.

Lavin, D. (2013) "Must there be basic action?." *Noûs* 47.2: 273–301.

Levy, N. (2016). Embodied savoir-faire: Knowledge-how requires motor representations. *Synthese*, pages 1–20.

Lewis, D. (1970). General semantics. *Synthese*, pages 18–67. Reprinted in Lewis(1983), pp. 189–232.

Lewis, D. (1979) Attitudes de dicto and de se. *The Philosophical Review* 88.4: 513–543.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt and Co., New York.

Mazzoni, P. and Krakauer, J. (2006). An implicit plan overrides an explicit strategy during visual-motor adaptation, *The Journal of Neuroscience*, 26(14):3642–3645.

Miall, R. C. and Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural networks*, 9(8):1265–1279.

Millikan, R. (1984). Language, Thought and Other Biological Categories, Cambridge, MA: MIT Press.

Miller, G. A (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review* 63.2 :81.

Moschovakis, Y. (1994). Sense and Denotation as Algorithm and Value. In Logic Colloquium '90 (Helsinki 1990), Volume 2 of Lecture Notes in Logic, pp. 210–249. Berlin: Springer.

Muskens, R. (2005). Sense and the Computation of Reference. *Linguistics and Philosophy*, 28(4):473–504.

Neander, K. (1995). “Malfunctioning and Misrepresenting”, in Philosophical Studies, 79: 109–141.

Neander, K. (2006). “Content for Cognitive Science”, in G. McDonald and D. Papineau (eds.), Teleosemantics, Oxford: Oxford University Press, 167–194.

Newell, A. and Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. Communications of the ACM, 19(3):113–126.

Pavese, C. (2015). Practical Senses. *Philosophers' Imprint*, 15(29):1–25.

Pavese, C. (2016). Logic Inference and its Dynamics. In Roy, Taming, and Willer, editors, *Deontic Logic and Normative Systems: Proceedings of the 13th International Conference*, pages 203–219. College Publications, Milton Kaynes-GBR.

Pavese, C. (2017). Know how and Gradability. *The Philosophical Review*, 126(3): 345–83.

Pavese, C. (2018). On the Psychological Reality of Practical Representation. *Philosophical Psychology*. Forthcoming.

Peacocke, C. (1995). Content, computation and externalism. *Philosophical Issues*, 6:227– 264.

Peacocke, C. (1999). Computation as involving content: A response to Egan. *Mind & Language*, 14(2):195–202.

Piccinini, G. (2008). Computation without representation. *Philosophical studies*, 137(2): 205–241.

Plotkin, G. (1981). A Structural Approach to Operational Semantics. Tech. Rep. DAIMI FN-19, Computer Science Department, Aarhus University, Aarhus, Denmark.

Plotkin, G. (2004). The Origins of Structural Operational Semantics. *Journal of Logic and*

Algebraic Programming, 60–61:3–15.

Priest, G. (1999). On a version of one of Zeno's paradoxes. Analysis 59 (261): 1-2.

Rescorla, M. (2012) How to integrate representation into computational modeling, and why we should. Journal of Cognitive Science 13.1: 1-38.

Rescorla, M. (2014) The causal relevance of content to computation. Philosophy and Phenomenological Research 88.1: 173-208.

Rizzolatti, G., Fogassi, L., and Gallese, V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2(9): 661–670.

Sakai, K., Kitaguchi K., and Hikosaka O. Chunking during human visuomotor sequence learning. *Experimental brain research* 152.2 (2003): 229–242.

Setiya, K. (2012) XIV—Knowing How. *Proceedings of the Aristotelian Society*. Vol. 112. No. 3. Oxford University Press.

Stanley, J. (2011). Know How. Oxford University Press, Oxford.

Stanley, J. and Williamson, T. (2001). Knowing How. *Journal of Philosophy*, 98(8):411–444.

Starr, W. (ms). A Preference Semantics For Imperatives.

Thompson, M. (2008): *Life and Action*. Cambridge: Harvard University Press.

Trappenberg, T. (2009). Fundamentals of computational neuroscience. OUP Oxford.

Tulving, E. (1985). How many memory systems are there? *American psychologist*, 40(4): 385.

Verwey, W. B., Abrahamse, E. L., Ruitenberg, M. F., Jiménez, L., & de Kleine, E. (2011). Motor skill learning in the middle-aged: limited development of motor chunks and explicit sequence knowledge. *Psychological research*, *75*(5), 406–422.

Verwey, W.B. (2010). Diminished motor skill development in elderly: indications for limited motor chunk use. *Acta Psychologica,* 134, 206–214.

Verwey, W.B. (2001). Concatenating familiar movement sequences: the versatile cognitive processor. *Acta Psychologica*, 106:69–95.

Verwey, WB. (1996). Buffer loading and chunking in sequential keypressing. *Journal of Experimental Psychology Human Perception and Performance,* 22,544–562.

Wolpert, D. M. (1997). Computational approaches to motor control. *Trends in cognitive sciences*, 1(6): 209–216.

Wolpert, D. and Ghahramani, Z. (2000). Computational principles of movement neuroscience. *Nature Neuroscience*, 3:1212–1217.

Wolpert, Daniel M., Kenji Doya, and Mitsuo Kawato. (2003). A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences* 358.1431: 593–602.