A Theory of Practical Meaning

Carlotta Pavese

Draft of May 1st 2017

Abstract

This essay introduces the notion of practical meaning by looking at a certain kind of procedural systems — *the motor system* — that play a central role in computational models of motor behavior. I argue that a satisfactory account of the content of the representations computed by motor systems (*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 concludes that practical meanings play a central 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 two important objections.

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)): the system performs an elementary operation *directly or immediately*, that is, without thereby performing any other operation as its proper part. 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*.

On the suppositions that systems come, at any given time, with a fixed set of elementary operations and 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, at a time *t*, to an instruction for a system 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 this 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

[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, 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).

one that assigns a central 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 abilities, 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

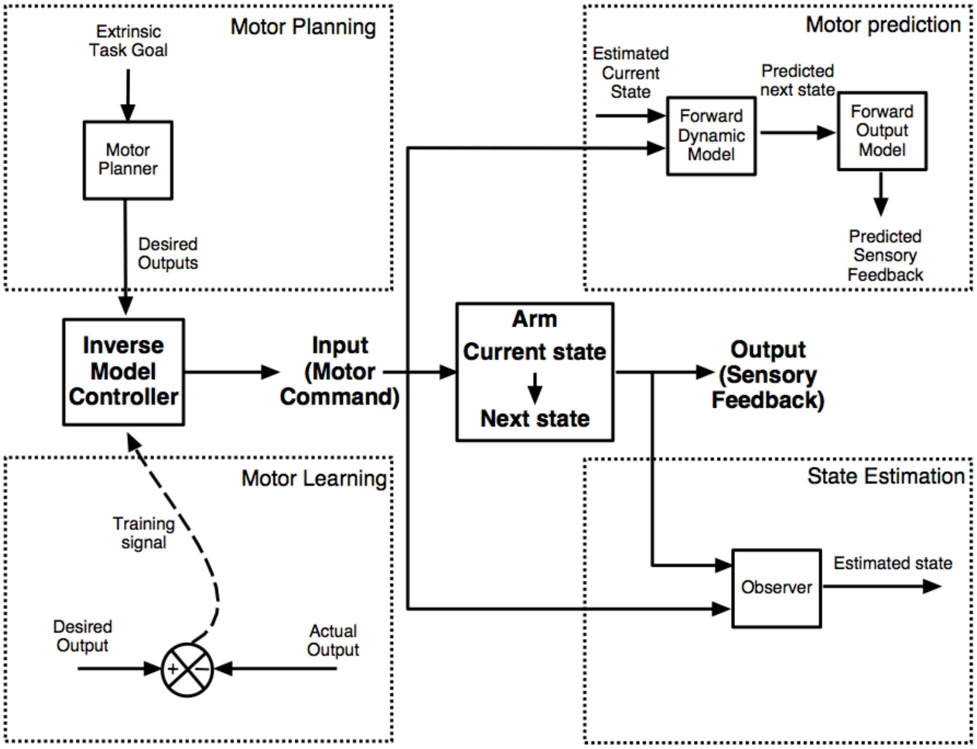
The computational approach to the study of motor behavior is one of the most successful areas of research in cognitive psychology and cognitive neuroscience. According to computational models of motor behavior, a motor task such as, for example, the task of pouring wine on a glass (Figure 1) involves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

Figure 1: Example of a motor task

objects (bottle and glass) and the limbs into a series of *motor commands*. Such motor 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 on the glass (Miall and Wolpert 1996, Wolpert 1997, Kawato and Wolpert 1998, Kawato 1999, Wolpert and Ghahramani 2000, Trappenberg 2009).

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

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



that can then be executed by the motor system; second, 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 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).

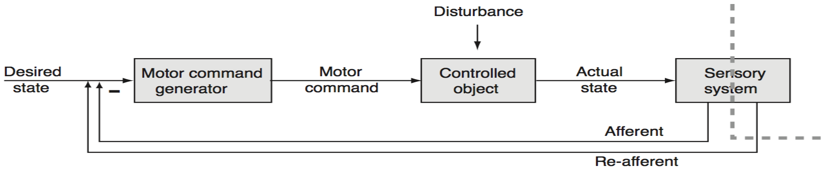


Figure 3: Trappenber's representation of the motor system (2009, 271)

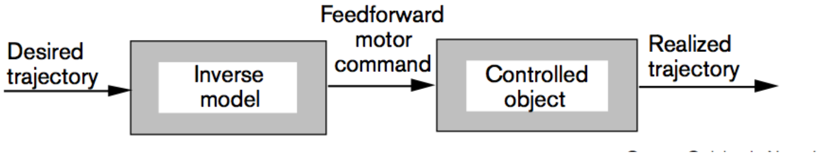


Figure 4: Kawato's representation of the motor system (1999, 719)

In these computational models, then, the role of motor commands can 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 the 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*.[2] 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, we do not need to lean on the assumption that motor commands must be linguistic. We may leave it open that motor commands are more akin to *pictorial instructions* (or *imperatival pictures*) such as architectural plans or road-sidewarning signs than they are to linguistic representations.[3]

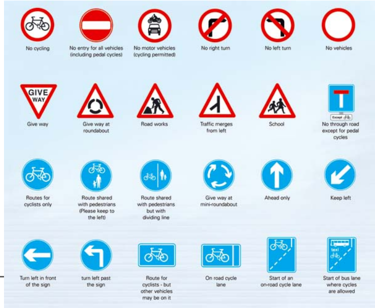

GIVE WAY
No cycling
No entry for all vehicles (including pedal cycles)
No motor vehicles (cycling permitted)
No right turn
No left turn
No vehicles
Give way
Give way at roundabout
Road works
Traffic merges from left
School
Except
No through road except for pedal cycles
Routes for cyclists only
Route shared with pedestrians (Please keep to the left)
Route shared with pedestrians but with dividing line
Give way at mini-roundabout
Ahead only
Keep left
Turn left in front of the sign
turn left past the sign
Route for cyclists - but other vehicles may be on it
On road cycle lane
Start of an on-road cycle lane
taxi
Start of bus lane where cycles are allowed

Figure 5: Example of imperatival pictures

[2] 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.

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

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 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.[4] 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 the 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

[4] 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.

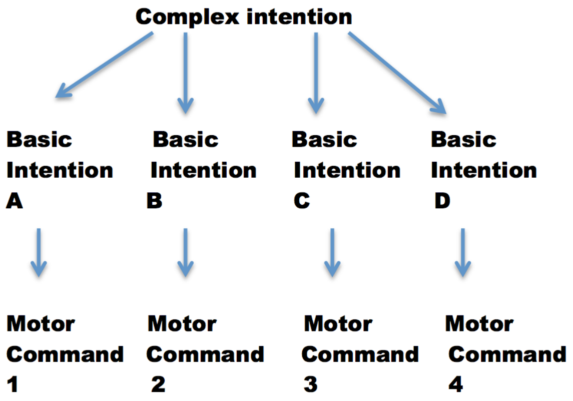


Figure 6: One-One Model

input to the translation by the motor planning must be some content available at the personal level (the *personal level input hypothesis*).[5] Finally, these intentions that 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 that is basic for the agent — in the sense that that agent can perform it intentionally without performing intentionally any part of it (Danto 1965). An intention is complex if it is not basic. 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

[5] 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.

basic intentions.[6]

In addition to these three hypotheses that are shared by most, if not all, descriptions of the motor system, to facilitate my presentation, I will make one further assumption: I will assume that bits of intentions procedurally translate into *a single* motor command (*one-one model*). Figure 6 illustrates the *one-one model*: a complex intention to pour wine on a glass divides into parts — i.e. basic intentions — and each of these basic intentions is mapped into a single motor command. This is certainly not the only way we can think of the relation between basic intentions and motor commands. We could envisage a different model, on which basic intentions themselves are translated into a series (or some other arrangement) of motor commands, not just into a single one. In other words, basic intentions could stand into a one-to-many relation to motor commands (*One-many model*, Figure 7). As I explain at the end, whether we assume the one-one model or the one-many model is immaterial to my main argument. For expository purposes, however, it will be convenient to start by assuming the one-one model. I will make sure to discharge this assumption later in the essay.

[6] 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.

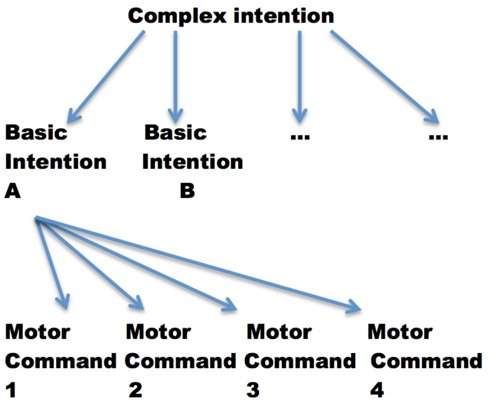


Figure 7: One-many model

Combining together the *procedural hypothesis*, the *personal level input hypothesis*, the *discrete hypothesis*, and the *one-one model,* 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 8: blue arrow = input; orange arrow = output):[7]

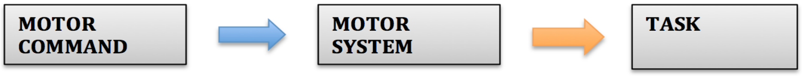


Figure 8: Input-output

Returning to our main question — what is the meaning of a motor command? — our discussion thus far suggests that we describe the function of a motor command as issuing a

[7] 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. This simplification will not affect the main claim of this section for it is undeniable that at least one of the functions of the motor system is to output the execution of a motor task.

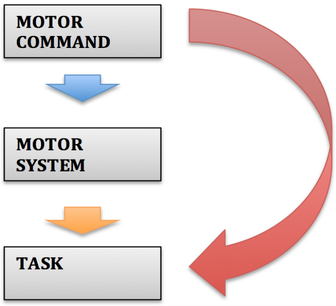


Figure 9: The Denotational Model

prescription whose content is a task (Figure 9: red arrow = denote). 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:[8]

(1) Dance!

What is an action? As Barker (2012, 1) puts it:[9]

> 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.

[8] 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).

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

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. 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* and other ones in which 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 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 $<w_{\mathrm{i}}, c>$ — 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): on this semantics, (1) denotes a set of ordered pairs of the form $<<w, c>, <w_{\mathrm{i}}, c>>$.

Extending Barker's proposal to the semantics of motor commands, we may then think of the meaning of a motor command *C*! as a task, 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 *C*-ing.

All in all, the denotational model sketched thus far is a plausible semantic analysis of motor commands. In what follows, I will argue, however, that although it is partially correct, the denotational model of motor commands is incomplete: the denotation of motor commands — i.e. the 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. 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 an infinite 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 10, as before orange = output; blue = input).[10]

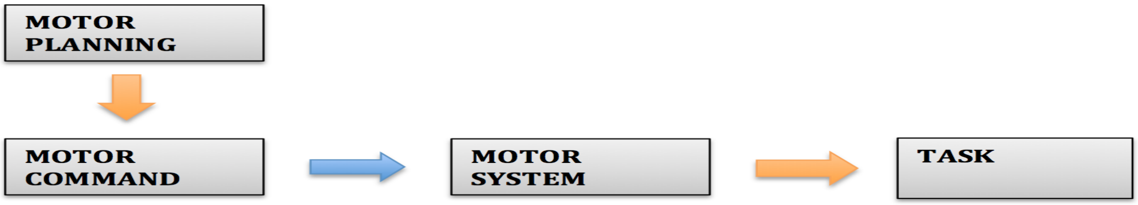


Figure 10:The role of motor planning

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

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 11 (from Wolpert 1997, 3) shows the motor hierarchy. In it, the same task goal — e.g. 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 by a certain method*. 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 is not ambiguous but is instead, univocal, for its instruction can now be executed only in one way.

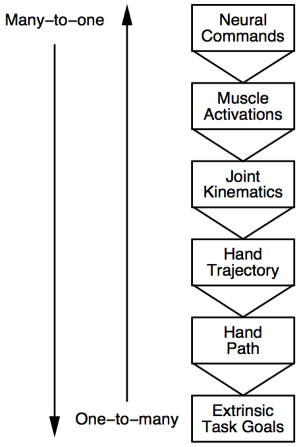


Figure 11: The Motor Hierarchy

So, the denotational model is incomplete. Motor commands do not just denote, or just represent, tasks. A specification of their contents must include mention of the *methods* in accordance with which they prescribe 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.

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 12: black = expresses; green = fixes).

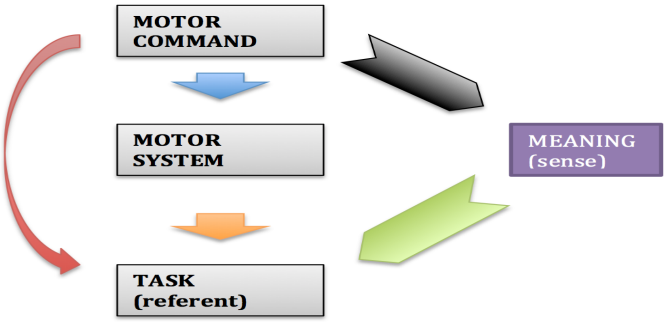


Figure 12: The Two-Dimensional Model

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).

2.3 The need for distinctively practical meanings

The picture is incomplete unless we clarify what a method is. As I have argued in Pavese (2015, 2-3), different methods can be thought of *as ways of breaking down a task into sub-tasks or sub-operations*.[11] 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 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.

Consider two different sets of directions to reach a same destination X, as shown by

[11] In my jargon, *tasks* are operations that can be performed intentionally and their execution can be attributed to the person/agent. Many operations are beyond the reach of intentionality: their execution can be attributed to the motor system but not to the person/agent.

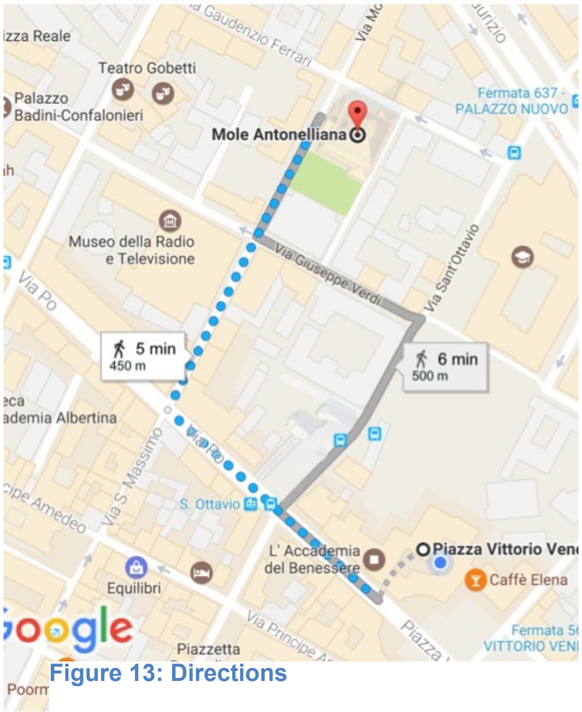


Figure 13: Directions

Figure 13. The first set of direction (in blue) breaks down the task of reaching Mole Antonelliana from Piazza Vittorio Veneto in Turin in a sequence of tasks that consists in taking via Po to the crossing with via San Massimo and then in turning right until almost reaching via Ferrari. The second set of directions (in grey) breaks down the same task into different sets: after taking via Po, it says to turn right on via Sant'Ottavio, to turn left on via Verdi, and then again right on via S. Massimo. These two sets of directions are two methods to reach the same destination, for they break the same task – reaching the Mole Antonelliana – 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 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. Otherwise, the motor system could never complete the execution of the task. If ways of breaking the task into parts cannot divide into subtasks indefinitely, they 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 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.[12] 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: the system comes to execute it directly, without executing any of its parts. 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).

If chunking is possible, then the set of elementary operations of a system must change over time, for the new chunks get to be included in the list of newly acquired elementary

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

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 through 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 depends on the system one considers and on its stock of elementary operations. Accordingly, because an assignment of meaning or sense to a motor commands 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: what we assign to a motor command as its meaning will depend on the relevant system's stock of elementary abilities.

This completes my argument for thinking that the denotational model is incomplete. The meaning of motor commands cannot be exhausted by their denotations. There is a further meaning dimension to the meaning of motor commands — i.e. the method through which they prescribe that a task is to be performed. I have argued that these methods stand to tasks as meanings (or senses) stand to their denotation (or referent). Moreover, I pointed out that this further dimension of meaning is distinctively *practical*, for it is relative to a system's set of elementary abilities. Hence, I called this further dimension "practical meaning."

2.4 More Support for the Two-Dimensional Model

Before moving on, let me pause to consider an alternative to the two-dimensional model that might seem to work just as well. Suppose that, instead of adding a further dimension to the meaning of motor commands (a sense in addition to their referents), we make their referents (the relevant tasks) more fine-grained. One way of doing so is to take motor commands to refer to *highly specific tasks* — i.e. tasks that cannot be executed in more than one way. We could model such fine-grained tasks by thinking of them 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 include 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 methods directly, rather than tasks, and banish altogether the level of meanings (or senses). In this way, by making the referent more fine-grained, we could obviate to the need of appealing to senses in addition to referents. Or so one might argue.

This picture has 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*, extrinsically 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 task, even though in different ways. Consequently, it is extremely natural to follow **Action Semantics** in taking these coarsely individuated tasks to be what commands prescribe and to identify these tasks that motor commands prescribe with the motor commands' denotation. In the last section, I showed that the required fineness of grain can be reached by adding a further layer of meaning — a sense in addition to their 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 grasping 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 the role of motor commands' senses.

In conclusion, the two-dimensional model provides a more natural answer to the question "What is the meaning of motor commands?" than its obvious alternative. 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. According to the *one-one model*, a single motor command translates procedurally a basic intention. On this model, the denotational model has it that motor commands have tasks as their denotation — modeled as ordered sets of input-output. But this model is incomplete. Such tasks can be executed in accordance with different methods. So if the input to the motor system is to be unambiguous, the motor command must prescribe what method is to be used by the motor system for the execution of the task. Therefore, I suggested that a motor command represents a task 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 the notion of an elementary operation is intrinsically 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 one-one model and on the *Explanatory Constraint*. It is time to discharge or to defend these assumptions.

As I have observed at the outset, we could envisage the relation between the basic intentions and the motor commands to be one-many, as illustrated by Figure 7. On this model, a basic intention is translated into a complex representation, made out of an arrangement of motor commands, each corresponding to the elementary operations of the system. One might think that on this model, practical meanings are dispensable, for this complex representation can be simply 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.

This way of looking at things is, however, misleading. On this one-many model, the question arises: In virtue of what does the motor planning map a certain basic intention into that particular complex representation? Part of the answer to this question must be that that particular complex representation represents the relevant task in terms of operations that the system can elementary perform. Hence, although this complex representation's content can be thought of as a structured task, structured out of those elementary operations, what determines that it have as its content 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, as we have seen, a way of breaking down a task into operations that are elementary for the system is nothing else but a practical meaning. Hence, also on the one-many model, we get practical meanings back into the picture, playing on this model the meta-semantic role of fixing the denotation of the complex motor representation. Therefore, on the one-many model too, practical meanings play a role in the explanation of motor behavior, although a role that is more similar to that of a Kaplanian character than to that of a Fregean sense (Figure 14).

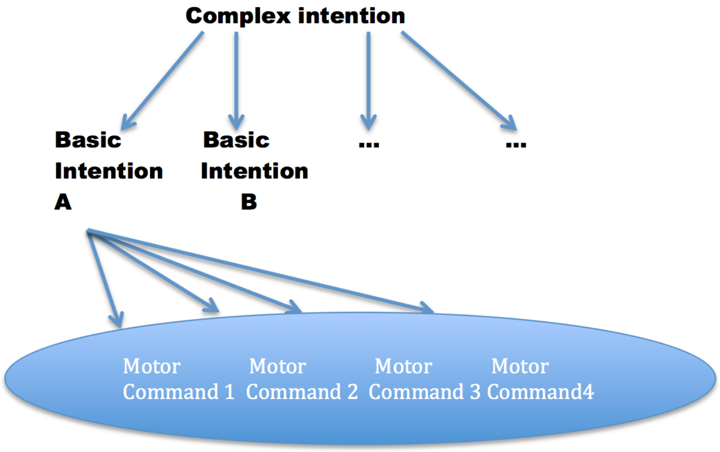


Figure 14: Sequence

The last assumption left to discuss is what I called the *Explanatory Constraint*. According to it, motor commands are representations and 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 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 only compute representations (Newell and Simon 1976, Fodor and Pylyshyn 1988, Peacocke 1995, Peacocke 1999), 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 appeal to 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, as we have seen for example in the discussion of Gallese and Metzinger (2003) and Butterfill and Sinigaglia (2011), 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 content, 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 they need motor representation and the best account of the content of motor representation is in terms of practical meanings.

Practical meanings are ways of breaking down tasks into subtasks that a system can elementarily perform. So, assignments of practical meanings are relative to a system's stock of elementary abilities. 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^1), and sequencing (;). If so, then an operational semantics will assign PROGRAM TEXT a complex semantic value (**R-MULT**) that describes multiplication in terms of those operations:

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

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[13] 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

[13] Configurations are indicated by an ordered pair of an instruction and an assignment function f assigning values to variables in the instruction — e.g. ⟨MULT (n, m), f⟩.

result *res*, after performing a series of additions of *n* to itself $m - 1$ times (**R-MULT2**).[14]

Consider now a different system, whose set of elementary abilities only includes the operation **ADD$_n$** 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-ADD$_n$**:

R-ADDn = $\langle \mathrm{ADD}_n(m), f\rangle \longrightarrow \langle res, f^*\rangle$ 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 ADD$_n$ (m):

R-MULT*=
$$\left\{\begin{array}{l} \text{If } \langle \mathrm{ADD}_n(m), f\rangle \longrightarrow \langle res, f^*\rangle, \text{ then} \\ \\ \langle \mathrm{MULT}(n, m), f\rangle \longrightarrow \langle res, f^*\rangle \end{array}\right.$$

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.

[14] 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.

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 v^1, 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.

Second, operational semantics is fully compositional in that the operational semantic values of complex instructions is fully determined by the operational semantic values of simpler instructions together with the structure of those complex instructions.

By satisfying the conditions of structure and relativity, operational semantics provides a perfect example of an assignment of practical meanings to commands in a programming language.

3.2 Elementary Abilities

An assignment of practical meanings is relative to a system's elementary abilities — so that the same command may be assigned different practical meanings if it were fed into two systems that differ in their elementary abilities.

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

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 appealing 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).

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. It seems utterly 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 utterly 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. 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:

(i) <u>sequence</u>: execute A; B; by executing A; then executing B;

(ii) <u>loop</u>: execute A; B; by executing A; then executing B;

(iii) <u>if-command</u>: execute A if C obtains;

(iv) <u>while-command</u>: 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.

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 can 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 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 primitively perform and that can put together in accordance with a system's primitive modes of combination. In this sense, an assignment of practical meanings is relative to a system's set of primitive abilities. Practical meanings can be modeled formally 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.[15] In 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 the idea 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

[15] Cfr. Kaplan 1989.

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 they 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.[16] 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

[16] 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 (2011)'s notion of practical modes of presentation.

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 content of these prescriptive representations?

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. The denotational model, on which motor commands denote tasks, is only partly correct. In addition to denoting tasks, motor commands have a further dimension to their meaning, if they have to play the functional role that 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 put together through its primitive modes of combinations. Hence, by assigning a central role to motor commands and motor representations, computational explanations of motor behavior 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 (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 intention 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 & Signigaglia 2014, Levy 2016, Fridland 2017). An exhaustive review of this literature is beyond the scope of this essay. Let me just mention a further possible 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 cases of *goal-directed* motor behavior that are nonetheless involuntary were observable. 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. 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 the discussion of these empirical findings for future work (cfr. Pavese 2018). I also leave it to future work to discuss whether practical meanings could figure as building blocks of thoughts[17] 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*.[18]

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.

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.

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

[18]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 for comments. I am particularly grateful to Barbara Vetter, Guillermo Del Pinal, Gabe Greenberg, Ian Jakeway, Diego Marconi, Bryan Pickel, Brian Rabern, Anders Schoubye, Will Starr, and Wolfgang Schwarz for helpful discussion about some of the ideas in this article.

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). *No^us*, pages 24–33.

Church, A. (1974). Outline of a revised formulation of the logic of sense and denotation (part ii). *No^us*, 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.

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.

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

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

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

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.

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.

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, Tamminga, 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 h45ow and gradability. *The Philosophical Review*, 126(3). forthcoming.

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.

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. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society B: Biological Sciences* 358.1431 (2003): 593-602.