A Praxical Solution of the Symbol Grounding Problem

Mariarosaria Taddeo · Luciano Floridi

Abstract This article is the second step in our research into the *Symbol Grounding Problem* (SGP). In a previous work, we defined the main condition that must be satisfied by any strategy in order to provide a valid solution to the SGP, namely the *zero semantic commitment condition* (Z condition). We then showed that all the main strategies proposed so far fail to satisfy the Z condition, although they provide several important lessons to be followed by any new proposal. Here, we develop a new solution of the SGP. It is called *praxical* in order to stress the key role played by the interactions between the agents and their environment. It is based on a new theory of meaning—*Action-based Semantics* (AbS)—and on a new kind of artificial agents, called *two-machine artificial agents* (AM^2). Thanks to their architecture, AM^2s implement AbS, and this allows them to ground their symbols semantically and to develop some fairly advanced semantic abilities, including the development of semantically grounded communication and the elaboration of representations, while still respecting the Z condition.

M. Taddeo
Dipartimento di Filosofia, Facoltà di Lettere e Filosofia, Università degli Studi di Padova, Padova, Italy
e-mail: mariarosaria.taddeo@unipd.it

M. Taddeo · L. Floridi (✉)
IEG, University of Oxford, Oxford, Great Britain
e-mail: luciano.floridi@philosophy.ox.ac.uk
e-mail: mariarosaria.taddeo@unipd.it

L. Floridi
Department of Philosophy, University of Hertfordshire, Hatfield, Great Britain

L. Floridi
St Cross College, University of Oxford, Oxford, Great Britain

Keywords Action-based semantics · Artificial evolution · Communication · Hebb's rule · Local selection · Symbol Grounding Problem · Two-machine artificial agents · Zero semantic commitment condition

Introduction: the Symbol Grounding Problem, its Nature and Constraints

The *Symbol Grounding Problem* (SGP) (Harnad 1990) is one of the crucial questions in the philosophy of information (Floridi 2004). It concerns the difficulty of specifying how an artificial agent (AA) can *autonomously* elaborate its own semantics for the symbols that it manipulates, by interacting with its environment and other agents. As Harnad puts it: 'How can the semantic interpretation of a formal symbol system be made intrinsic to the system, rather than just parasitic on the meanings in our heads? How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?' (Harnad 1990, p. 335).

In Taddeo et al. (2005), we analysed the SGP and assessed the main strategies developed in the last fifteen years in order to solve it. That research was meant to lay the foundation for this article, which introduces a new solution to the SGP. We shall therefore begin by briefly summarising the main conclusions reached in our previous work, to which the interested reader is referred for further details.

The requirement to be satisfied by any strategy seeking to solve the SGP is the *zero semantic commitment condition* (henceforth Z condition). According to the Z condition, no valid solution of the SGP can rely on forms of

(a) *innatism*, since no semantic resources (some *virtus semantica*) should be presupposed as already pre-installed in the AA; and
(b) *externalism*, since no semantic resources should be uploaded from the "outside" by some *deus ex machina* that is already semantically-proficient.

Of course, a valid solution draws on an AA's own capacities and resources (e.g. computational, syntactical, procedural, perceptual, educational resources, exploited through algorithms, sensors, actuators etc.) to ground its symbols. However, these should not already be semantic, as that would be begging the question.

Points (a) and (b) also clarify the sense in which a valid solution of the SGP must be fully *naturalised*, despite the fact that we are talking about *artificial* agents: no supernatural or extra-natural strategies are allowed.

Any approach that breaches the Z condition is semantically committed and fails to provide a *valid* solution to the SGP.

All the eight strategies analysed in Taddeo et al. (2005) seek to ground the symbols through the sensorimotor capacities of the AAs involved. They differ in the methods used to elaborate the data obtained from the sensorimotor experiences, and in the role (if any) assigned to the elaboration of the data representations in the process of generating the semantics for the symbols. However, they all turn out to be semantically committed, so none of them can be said to provide a valid solution to the SGP in the sense specified above. The most difficult issue is represented by the

semantic capacity to generate representations. For it cannot be presupposed without begging the question, but abandoning any reference to representations means accepting a dramatic limit to what any AA may be able to achieve semantically, since the development of even the simplest abstract categories becomes impossible.

From the analysis of the eight strategies, we concluded that a valid solution of the SGP will need to combine at least the following features:

1. a bottom-up, sensorimotor approach to the SGP;
2. a top-down feedback approach that allows the harmonization of top-level grounded symbols and bottom-level, sensorimotor interactions with the environment;
3. the availability of some sort of representational capacities in the AA;
4. the availability of some sort of categorical/abstracting capacities in the AA;
5. the availability of some sort of communication capacities among AAs in order to ground the symbols diachronically and avoid the Wittgensteinian problem of a "private language";
6. an evolutionary approach in the development of (1)–(5);
7. the satisfaction of the Z condition in the development of (1)–(6).

This was the recommendation made at the end of our previous research (Taddeo et al. 2005). In this paper, we propose a solution for the SGP that respects the Z condition and also satisfies the requirements listed above. For reasons that will soon become clear, we shall refer to it as the *praxical*[1] solution. We shall introduce it in two steps.

The first step, taken in the Section "Action-based Semantics", consists in outlining the appropriate approach involved in the process of generating new meanings. This is defined as *Action-based Semantics* (AbS). AbS requires an explanation of the specific process that allows the coupling of symbols to meanings. Such coupling is more intuitively introduced by referring to an actual agent implementing AbS, so we shall postpone its theoretical description until the Section "Two-Machine Artificial Agents and their AbS".

The second step, taken in the Section "Two-Machine Artificial Agents and their AbS", consists in describing a *two-machine* AA (AM^2) that implements the AbS. An AM^2 assigns meanings to symbols without elaborating any kind of categorical representation yet. We shall see that it does not presuppose *semantic* resources or capacities in order to generate its semantics, and hence that it satisfies the Z condition. We shall also describe the second stage of the semantic process, namely how an AM^2 generates representations. These are neither categorical nor conceptual, unlike Harnad's, and yet it will be shown that they allow the development of a semantics in which symbols may be names of classes of meanings. Such semantics avoids both the constraints highlighted in (Taddeo et al.

[1] In the same sense in which "praxis" is used to refer to "theory in practice", we use "praxical" to qualify interactions that are information—or knowledge-oriented. An embodied and embedded agent has a *praxical* relation with its surroundings when it learns about, and operates on, its environment in ways that are conducive to the acquisition of implicit information or knowledge about it. In human agents, *practical* experience is non-theoretical, whereas *praxical* experience is pre—but also pro—theoretical, as it conduces to theory.

2005) for the semantics generated by the non-representationalist strategy (Brooks 1990; Varshavskaya 2002), and the criticism levelled at the representationalist solutions (Harnad 1990). In the Section "Three Controversial Aspects of AM^2", we shall consider three objections to the process performed by an AM^2. In the Section "Learning the Performing Rule Through Hebb's Rule and Local Selection", we shall refer to a specific learning rule and to an evolutionary scenario in order to show how a population of AM^2s could develop its semantic abilities autonomously.

In the Section "From Grounded Symbols to Grounded Communication and Abstractions", we shall describe how a population of AM^2s can develop more complex semantics abilities, such as semantically grounded communication and a shared semantics.

In the conclusion, we shall briefly summarise the work done and discuss an interesting consequence of the praxical solution of the SGP, namely the possibility of developing a theory of meaning based on it. Its development, however, lies beyond the scope of this paper and will be left to a third and last stage in our research.

Action-based Semantics

The basic idea of an action-based semantics is simple: in the beginning, the meanings of the symbols generated by an AA are the internal states of that AA, which in turn are directly correlated to the actions performed by the same AA.

Consider a common AA, such as a robot able to move in a laboratory. Let us call it FOTOC.[2] We shall describe and discuss FOTOC in the next section in more depth, but here suffice to say that any time FOTOC executes a movement, such as 'turning left', it enters into a specific internal state and should be able to take advantage of this internal state as a meaning to be associated to a symbol. So, by saying that the performed actions are the meanings of the symbols, we mean that the AA relates its symbols to the states in which it is placed by the actions that it performs, and that symbols are considered the names of the actions via the corresponding internal states.

The advantage of this approach is that the very first step in the generation of meanings is not in itself a *semantic process*, but rather an *immediate consequence* of an AA's performance. Through AbS, an AA can generate meanings without its perceptual data (e.g. FOTOC's detection of its location in the lab office) causing some kind of representations, a process that is always based on semantic criteria and therefore cannot but breach the Z condition. The internal states of the AA are excellent candidates for the role of non-semantic yet semantic-inducing resources.

By following the AbS, one avoids the use of any kind of external assistance (e.g. a programmer or a trainer) while also avoiding extrinsic biases: the initial generation of meanings is *teleologically free*, i.e. it is *neutral* with respect to any

[2] For a robot with similar skills to FOTOC's see Lego Wall Follower. It is equipped with a turret, enabling the rotation of its sensor (in the right direction) when a wall is detected, see http://www.techeblog.com/index.php/tech-gadget/lego-roverbot for a more detailed description.

purpose. Admittedly, most of the time, an AA performs an action in order to achieve some goal, but this form of teleological behaviour is not what is involved in the AbS. AbS assumes that the action performed—not the goal to be achieved—by an AA is going to ground its symbols semantically. In our example, FOTOC is supposed to ground a symbol to its internal state, induced by its action of turning left, and not by its command or goal 'avoid this obstacle' or 'catch that object' or 'turn left'. This is both plausible and easily achievable. The development of an AA's goal-oriented behaviour may be the result of the evolution of bio-chemical mechanisms that require no semantic resources at all. The heliotropic behaviour of plants, such as snow buttercups or sunflowers, is a canonical example.[3] Note that, even if an AA performs some action randomly—without any function or goal—or incorrectly, AbS still identifies that action as the source of the state that then provides the meaning of the related symbol.

To summarise, at this stage, the *purpose* of the action has no direct influence in the generation of the meaning. Hence, in AbS there are no extrinsic semantic criteria driving the process of meaning generation. This initial stage of the process is free of any semantic commitment, and thus satisfies the Z condition.

In the next section, we shall see how the general idea of an AbS may be implemented by an AA. We shall then consider the importance of evolutionary processes in the development of semantic capacities. Here, in order to clarify the AbS further, it is worth disposing of a potential misunderstanding. It concerns the similarities between AbS and the "meaning as use" semantics associated with the later Wittgenstein.

According to that semantic theory, a language is a form of social interaction. The meanings of the symbols follow from the uses of the language in given contexts, and from negotiations, stipulations and agreements among the speakers. Meanings are therefore partly conventionally defined in a community of speakers, partly identified with the speakers' intentions to perform some actions, given some symbols. All this qualifies Wittgenstein's linguistic games, pragmatically speaking, as teleological. Recall that, according to Wittgenstein, the meaning of the word 'slab' must be referred to its function within the linguistic game in which the word is used. A bricklayer says 'slab' in order to interact with his co-worker and cause him to have a specific reaction: the one which involves giving him the slab. Then, it seems that the meaning of 'slab' is the action that the co-worker executes in association with the word 'slab'.

All this may seem to be very similar, or perhaps outright identical, to a version of the AbS theory. The problem highlighted by this criticism is that, if AbS is indeed a semantics of use *à la* Wittgenstein, it follows that meanings really arise from social interactions among speakers, i.e. agents already belonging to a community that shares means of communication, and from a kind of practical finalism. However, these are all features that represent external criteria, and hence presuppose some pre-established semantic abilities on behalf of the agents involved. If such a family

[3] The diurnal motion (being these of flowers or of leaves) is a response to the direction of the sun, performed by motor cells in flexible segments of the plant specialized in pumping potassium ions into nearby tissues (thus changing the turgor pressure) reversibly.

resemblance between AbS and Wittgensteinian linguistic games were correct, it would be very hard to see how one could deny that AbS breaches the Z condition.

The criticism can be answered by explicating three main differences between AbS and the "meaning as use" semantics, which significantly differentiate the former from the latter and hence defuse the objection.

First, in the semantics of linguistic games, meaning is not the performed action. The meaning of 'slab' is defined through the linguistic game shared by the bricklayer and his co-worker, and the meaning is the way in which a symbol is to be *used* in order to trigger a particular reaction by the other player within the linguistic game. But in AbS the meaning of 'slab' is the internal state of the agent, a state triggered by the corresponding action. At this stage, no semantic interaction with other agents is yet in view.

Second, in a semantics based on "meaning as use", the association between meanings and symbols is conventional and contextual. It is based on negotiation and agreement among the speakers, requires training and is regulated by degrees of success. By contrast, according to AbS, the initial association of symbols and meanings is a direct input-output relation that follows only from the performance of actions. As we shall see in the following section, an individual agent associates a meaning with a symbol through the performance of an action, without considering the frame in which it has performed that action and, crucially, without taking into account the association performed by other AAs. The social component arises only *after* the association has taken place. To put it differently: according to AbS, semantics has its initial roots in the individual agent's behaviours, not in the community, and this is an advantage since, speaking in terms of logical order, the virtuous dialectic of interactions between a community of semantically-proficient agents and its members begins with the availability of individual agents capable of grounding their symbols, at least in principle and no matter how minimally and in some overridable way.

Third, to define meaning as a function of the use of the corresponding symbol entails a kind of finalism, which we have seen is not part of the AbS theory. AbS is therefore not a convention-based theory of meaning and does not entail, as a starting point, any kind of teleological theory of goal-oriented behaviour. This is what allows one to consider AbS free of any semantic commitment, unlike Wittgensteinian linguistic games, which clearly do not satisfy the Z condition.

The time has come to consider the AbS in more detail.

Two–Machine Artificial Agents and their AbS

In this section, we shall describe a kind of AA capable of implementing AbS. We have already referred to such an AA as a two-machine artificial agent, or simply AM^2. We shall argue that AM^2s can solve the SGP while satisfying the Z condition.

There are two main difficulties that must be overcome in order to show that an AM^2 solves the SGP correctly:

(i) it must be able to associate symbols to the actions that it performs; without
(ii) helping itself to any semantic resource in associating actions and symbols.

The architecture of an AM^2 explains how it can achieve (i) while avoiding (ii). This is based on features of the so-called *reflective architecture*, in particular on the availability of *upward-reflection* processes. Such an architecture is well-documented in (Barklund et al. 2000; Brazier et al. 1999; Cointe 1999) and the interested reader may wish to consult those works for a more in-depth description.

Essentially, the upward reflection is part of the *metaprogramming architecture*. A system capable of metaprogramming operates at two levels, which interact with each other. It organizes actions at an *object level* (OL), where it interacts with the external environment. But it can also take actions on its internal states and on its own elaborations. In this case, it operates at a *meta-level* (ML), which takes as data the actions at the OL. The relevant metaprograms are the *reflection processes*, where these function as *upward reflection*. In these metaprograms, the OL computation enables the ML computation. The modifications performed at the ML are effective and have a corresponding impact on the OL computation. The utility of reflection shows that the whole system [OL + ML] not only interacts with itself but is also properly affected by the results of such interactions.

The kind of AA we are discussing here is constituted by two machines—M1 and M2—which interact with each other and perform actions on two levels. M1 operates at OL, interacting directly with the external environment (e.g. by navigating, detecting obstacles, avoiding them etc.), thus outputting and inputting actions. M2 operates at ML and the target of its elaborations is the internal states of M1. Any action that M1 outputs to, or inputs from the environment defines a particular internal state (S_n) of M1. So actions and internal states are causally coupled: for any different action in M1 there is a different internal state S_n and for all similar actions in M1 there is the same S_n. Two points need to be clarified before proceeding further: *continuity* and *similarity*.

Clearly, the agent's actions/states are not necessarily organised into a discrete flow, but may be subject to analogue/continuous variations. For instance, FOTOC may seamlessly move from action *a* to action *b*, and hence from the corresponding internal state S_n to another internal state S_{n+1}. All the same, here we shall disregard details about how this flow may be broken into a set of discrete elements. What is crucial is that, as in a continuous tape, cutting the flow means cutting both sides, as it were, with the action on the one hand and the corresponding internal state on the other; and that the same types of agents may be reasonably assumed to have similar types of internal states triggered by similar types of actions and to "cut" their tapes in equally similar ways. This assumption of "physiological or hardware-related similarity" does not breach the Z condition, since it refers to hard (structural and/or physical) similarities among agents, not to similarities assumed by the agents at a soft (semantic) level. Again, one may compare it to the similarity occurring in the behaviour and environmental interactions showed by a field of sun-flowers.

To highlight the connection between M1's actions and states, we represent (see Fig. 1) the internal states of M1 as the results of a function (*f*) of interactions (*e*) between the machine (Machine 1) and the environment (E), so that $S_n = f(e)$.

Let us see now how the actions performed by an AM^2 may ground its symbols.

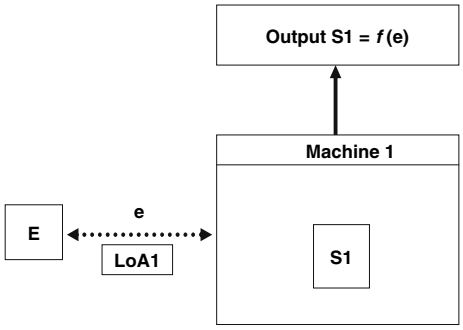

Fig. 1 The structure of Machine 1: E is the environment, S_1 is the internal state of Machine 1, LoA_1 is the level of abstraction at which Machine 1 interacts with E. f (e) is the function which identifies S_1, where (e) is a given interaction between the agent and the environment

Imagine an AM^2 positioned in an environment such a laboratory. In the previous section, we called this agent FOTOC. FOTOC is able to interact with the environment, it performs some actions, e.g. it moves around the laboratory office changing direction—and it has some perceptions. In particular, it is provided with a light-sensor on each of its sides, thus enabling it to detect the dark and light zones in the laboratory. When FOTOC detects a dark place, its M1 is in a specific internal state, say S_{dark}. Likewise, when FOTOC detects a light place, its M1 internal state is in S_{light}. For any dark place (for present purposes, the intensity of the darkness is irrelevant), FOTOC's M1 has the same (i.e. indistinguishable) internal state S_{dark}. That is why it does not need congruent perceptions of the environment to elaborate an internal state.

We use the theory of levels of abstraction[4] (LoA) (Floridi et al. 2004) to describe the degree of refinement of M1's perceptions. M1 accesses the environment at a LoA that allows only a specific granularity of detection of its features. Thus, through M1's perception, FOTOC can only obtain *approximate* (to whatever degree of granularity is implemented) *data* about its external environment. Note that such description makes full sense only from an external perspective, namely ours, where the LoAs are much more informative. For FOTOC, given its LoA, the world is just a sequence of dark and light *loci* with a hardwired LoA, i.e. with a specific granularity of details. The same holds true for the actions performed by an AM^2 embedded within an environment. Suppose FOTOC is able to move around the laboratory in such a way that it can turn 30° or 15° to the left. For both these actions, the M1 of FOTOC may have the same internal state, S_{left} if its LoA does not allow any discrimination between angles, but only the detection of a left turn. This feature follows from an AM^2's structure. LoAs are related to the interactions between AM^2s and the environment and to the features of the two machines M1 and M2 in the sense that

[4] In the Theory of Level of Abstraction (LoA) discrete mathematics is used to specify and analyse the behaviour of information systems. The definition of a LoA is: given a well-defined set X of values, an observable of type X is a variable whose value ranges over X. A LoA consists of a collection of observables of given types. The LoA is determined by the way in which one chooses to describe, analyse and discuss a system and its context. A LoA consists of a collection of observables, each with a well-defined possible set of values or outcomes. Each LoA makes possible an analysis of the system, the result of which is called a model of the system. Evidently, a system may be described at a range of LoAs and so can have a range of models. More intuitively, a LoA is comparable to an 'interface', which consists of a set of features, the observables.

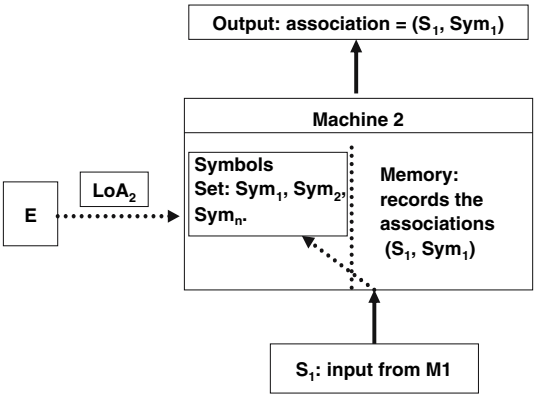

Fig. 2 The structure of Machine 2 (M2). E is the environment, M2 does not act on the environment but on M1; the environment acts on M2 indirectly, through the evolutionary process. Sym_1 is the symbol elaborated by Machine 2. LoA_2 is the level of abstraction at which Machine 2 interacts with E. (S_1, Sym_1) is the ensuing association between a symbol and an internal state of Machine 1, the output of M2's elaboration

they are hardwired in AM^2s, that is, they are *structurally dependent* on the physical implementation (embodiment) of the AAs and of their interactions with their environment.[5]

Following the metaprogramming architecture, M1 communicates with the other machine, M2. M1 sends its (uninterpreted) internal state to M2 (see Fig. 2 below). M2 is a symbol maker and retainer. It is constituted by a symbol source, a memory space, and a symbol set. The two machines communicate their data at their respective LoAs. M2 reads the states from M1 according to its LoA (LoA_2), which is less refined then M1's LoA. Because of LoA_2's granularity, M2 does not read S_n as it has been sent by M1; instead, S_n is modified by the LoA_2 in such a way that the new state is more generic. In other words, M1's internal state is *transduced* into a new state at LoA_2. For example, suppose the state sent to M2 is related to the action 'turn left by 32°', the state read by M2 according to its LoA is a more generic 'turn left'. The new state can be considerate the result of a function as $LoA_{2\ (Sn)} = S_{n2}$, where S_{n2} is a less specified state then S_n.

The transduction process is affected by M2's LoA. It is not defined by extrinsic criteria and it is not learned by the AM^2. Rather it follows directly from the AM^2's structure and its specific embodiment. Like bacteria, an AM^2 may be assumed to have developed the transduction processes by evolution.[6] Bacteria, cells and unicellular organisms perform transduction processes in order to interact with the

[5] Even if LoAs are not yet directly involved in the emergence of the elementary abilities required to overcome the SGP, a clear analysis of an agent's LoAs is crucial in order to understand the development of advanced semantic abilities. Hence, it is important to introduce an explicit reference to them at this early stage in the description of the architecture of an AM^2.

[6] Bacteria interact with the external environment, sending and receiving signals. The transmission of the signal is possible thanks to some receptors—*glicoproteins*—on the membrane. Such receptors interact with the signal's molecules, *ligandi*. The interaction determines a change that determines a new behaviour of the bacteria.

external environment and exchange information with it. During such processes the molecular structure of the signal is converted in such a way that it can be perceived by the receptor of the signal, so that the receptor can read the signal and modify its behaviour.

Once the new state is obtained, M2 associates the transduced state with a symbol removed from the symbol set. The process of removing a symbol from the set and coupling it with a state is discrete, non-recursive, arbitrary but not random, in the following sense. M2 makes explicit just one symbol for each input it receives; and cannot remove the same symbol more than once. The choice of the symbol is arbitrary, since it is semantically unrelated to the transduced states, but it is not random, because similar types of agents will associate similar symbols with similar transduced states. Still, symbols and transduced states are different kinds of data: they are associated—coupled together—but not transduced one into another.

Once a symbol has been chosen, M2 applies a *storing rule* and a *performing rule*. The storing rule records the symbol and the related state in the memory space. The performing rule regulates the communications between M1 and M2 and concerns the association between a symbol and a state. Following the performing rule, each time M2 receives an input from M1 it initially verifies whether the input received, or any another similar (i.e. indistinguishable by M2 at its LoA) input has already been elaborated. If M2 does not locate an input similar to the input stored in its memory, then it continues the process described above. Otherwise (if M2 finds the input, or an indistinguishable one, in its memory) it does not produce a new symbol, but reproduces the association already founded in its memory.

The association process is coherent: by following the performing rule, M2 obtains the same association any time it receives the same kind of input from M1, thus nomically associating different symbols to different internal states of M1. Any symbol elaborated by M2 is related through the internal state of M1 to a cluster of actions, i.e., all those actions not distinguished as different by the hardwired LoA. M2's symbols are now grounded in the actions through the corresponding internal states of M1. The resulting symbol is the outcome of a function, namely $\mathrm{Sym}_1 = g(\mathrm{S}_n)$. And since S_n is also the result of a function $f(e)$, the symbols selected by M2 are actually the result of a function of a function, $\mathrm{Sym}_1 = g\,(f(\mathrm{e}))$, see Fig. 3.

As we have shown above, M2's performances are also characterized by a specific LoA. In particular, the LoA of M2 is less refined than the LoA of M1. In our example, FOTOC's M2 may distinguish between M1's state $\mathrm{S}_{\mathrm{left}}$, related to the action 'turn left', and M1's state $\mathrm{S}_{\mathrm{maintain}}$, related to the action 'maintain this direction', but it may not draw any distinction between M1's state $\mathrm{S}_{\mathrm{left}}$ and M1's state $\mathrm{S}_{\mathrm{right}}$ related to the action 'turn right'. In short, by "abstracting in hardwired fashion", an AA ends up associating a single symbol with a cluster of similar actions. In the vocabulary of data compression, one may say that the process of transduction is lossy or never perfectly efficient. At this point, it is also important to stress that the whole process is formulated in such a way as to make it intuitive to us, external observers, but that, in order to satisfy the Z condition, no assumption should be made in terms of a "proper" way of abstracting that might result in some magic overlap between an AA's abstractions and ours. To use a previous example, heliotropism is a response to blue light, so if the plant is covered with a red

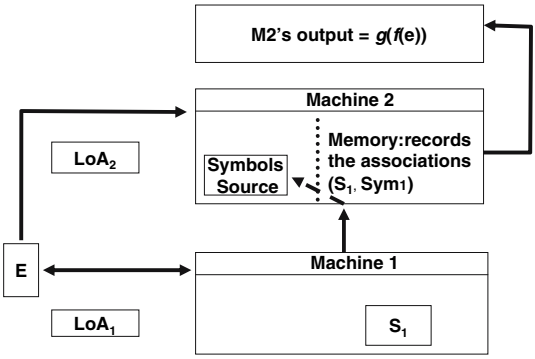

Fig. 3 Two-machine artificial agents' architecture. A two-machine artificial agent inputs/outputs some action/perception (e) from/on the environment E. E interacts with Machine 1 (M1) and acts on Machine 2 (M2) modifying it according to the evolutionary process. Any action is related to a corresponding internal state (S_1) of M1 at a specific level of abstraction, LoA_1. M 1 communicates its internal states to M2. M1's internal state is transduced into an input for M2, which associates the input with a symbol (Sym_1). M2 stores the state and the relate symbol in its memory. For any other input, M2 follows the procedure defined by the performing rule. Each symbol selected by M2 is a function (g) of the internal state, S_1. Since also S_n is the result of a function—*f*(e)—a M2's output is a function of a function, *g*(*f*(e))

transparent filter at night, blue light is blocked and the plant does not turn towards the sun, whereas a blue filter does not affect its behaviour. Now the filters are the physical implementations of the LoAs at which the plant interacts with its environment. So an external observer may simplify by saying that the plant abstracts the colour blue from light in hardwired fashion, in order to operate successfully in its environment. This is fine as long as it is not taken literally. In our example, FOTOC abstracts in ways that we shall see are merely determined by its evolution and survival as an agent.

From the fact that M2's LoA is less refined than M1's, it follows that M2 does not have a finely-grained perception of M1's internal states and may be unable to distinguish between M1's similar internal states. M2 will generate the same symbol to name all the actions which allow e.g. FOTOC to change the direction of movement. We call these symbols *general symbols*. To M2, the meaning of such a symbol is a *general meaning*, which arises from a generalization of similar meanings; in our example, for FOTOC's M2, the general meaning would be 'turning'.

An AM^2 does not have to rely on some semantic criterion in order to collect similar meanings in the first place and then elaborate the general one. Rather, we have seen above that a general meaning arises from a class of similar meanings elaborated by M2 according to its LoA. In its elaboration, M2 considers only the syntactical features of M1's internal states, not their meanings, i.e., the actions they refer to. So here too there is no semantic commitment in defining the class of meanings, which is elaborated whilst respecting the Z condition and can be used as a representation. In our example, FOTOC's M2 would not notice the difference between M1's internal states related to turning actions, but would simply consider all the states as if they were the same in elaborating a class of meanings.

From the elaboration of the abstraction follows an impoverishment of AM^2's semantics. In elaborating a general meaning, an AM^2 loses the specific meanings related to the symbols. Thus, it appears that the evolution of the praxical process would generate a semantics composed of generic meanings and lacking specific ones. For an evolved AM^2, there would be only the meaning 'turning' and there would (or indeed could) be no distinction between meanings such as 'turning left' and 'turning right'. To show how AM^2's semantics overcomes this shortcoming more details about the praxical process are required. This is a fair requirement but we will delay the explanation until the Section "From Grounded Symbols to Grounded Communication and Abstractions" because we need to concentrate our attention on a more basic issue first. The reader will recall that we have outlined two main problems that must be solved to overcome the SGP. The first one—the ability to elaborate meanings and associate meanings with symbols—has been solved in this section. In Sections "Three Controversial Aspects of AM^2" and "Learning the Performing Rule Through Hebb's Rule and Local Selection", we offer a solution to the second problem, the one posed by the fulfilment of the Z condition.

Three Controversial Aspects of AM^2

There are three main elements in the process performed by an AM^2 that might be criticized for not being semantically free: the *transduction process*, the *storing rule* and the *performing rule*. We shall now show that, in each case, the process described in the Section "Two-Machine Artificial and Their AbS" satisfies the Z condition.

One may suspect that the association between M1's internal states and M2's symbols is implemented by following some semantic criterion, yet the process described is purely mechanical, i.e. a simple input/output process in which, given an input, S_n, M2 transduces and associates it with a symbol, Sym_n. No semantic contents or interpretation rules occur at this stage. The symbols are chosen arbitrarily and the input S_n is elaborated by M2 only by virtue of its LoA. As we showed in the Section "Two-Machine Artificial and Their AbS", LoAs are hardwired in relation to AM^2, they define the kind of perceptions that the machines have of the environment and they do not imply any semantic content. What we have is a functional process that gives an output (symbol) for any received input (description of internal state). Input and output are then recorded together in M2's memory and only then do they become coupled together.

Against the availability of M2's capacity to apply the storing rule, one may object that recording capacities require in turn the ability to discriminate between useful (or relevant) and useless (or irrelevant) contents, but that this capability presupposes the existence of some semantic criteria that enable the agent to learn and apply some categorical order and to identify what should be stored and what should be discarded. However, M2 does not draw any distinction in applying the storing rule, as it records some/all of the received inputs and some/all of its outputs. Some numerical threshold might be implemented, but no categorical criterion is at work in defining how M2 applies the storing rule. The latter dictates that M2

registers the elaborations without any distinction. Thus, no semantic criteria are presupposed at this stage either.

The third aspect concerns the performing rule. One may argue that, given the sort of transduction, association and memorization described above, the AA must also be supposed to learn how to use the associated symbols and internal states (what we are treating as their meanings) successfully (that is, correctly, accurately, relevantly, efficiently etc.), and hence that it is at this stage that the AA must rely on some semantic resources, which would be extrinsic to the AM^2 and therefore beg the question. Perhaps not initially, but in the long run the elaborations of an AM^2 would not satisfy the Z condition and the SGP would remain unsolved. For one may object that an AM^2 cannot *acquire* any proficiency in using the grounded symbols without violating the Z condition. Once in place, the performing rule may satisfy the Z condition, but its development in the first place actually violates that condition.

Fortunately, the objection is mistaken since it is possible to show that AM^2s can learn how to use their symbols successfully through their interactions with the environment, without presupposing any semantic resource. This is the second step, which we are going to see in the next section.

Learning the Performing Rule Through Hebb's Rule and Local Selection

To show how a population of AM^2s can evolve to the point where its members can learn the performing rule while satisfying the Z condition we shall consider a typical learning rule, *Hebb's rule*, first formulated in Hebb (1949), and draw on the resources made available by the method of artificial evolution. More specifically, we shall consider *local selection* (LS) *algorithms*, and especially ELSA (*Evolutionary Local Selection Algorithm*) developed in Menczer et al. (2000, 2001). Note that the scenario described in the remainder of this section represents only a general framework, that is, only one of the possible ways in which AM^2s may be able to learn how to use the performing rule while respecting the Z condition. That there is such a possibility is all that is needed for our purposes; showing that this is the only way that is viable in terms of engineering, or that it is the actual way or even a biologically plausible way in which agents may be able to learn the performing rule falls outside the scope of this paper.

Hebb's learning rule may be summarised in the following statement: neurons that fire together wire together. The rule follows from a principle formulated by Hebb: 'When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased' (Hebb 1949, p. 62). Hebb's rule is considered a fundamental way in which experience changes behaviour in both vertebrates and invertebrates (Donahoe et al. 1997; Real, 1991).

Hebb's rule has been studied in biology and ethology and it is used to simulate learning processes with artificial neural networks. It is a general learning rule according to which an AA learns to couple an input and an output. The algorithms based on Hebb's rule define a kind of reinforced learning. This is the most common

process by which organisms learn from their interactions with the environment to achieve a goal. In such algorithms, the correlation of activity between two cells is reinforced by increasing the weighting between them; so the network's weightings are set in such a way that its output reflects its "familiarity" with an input. The learning follows from a *scalar reinforcement signal*, which is defined according to the efficiency—established through the environment's feedback—of the performed associations.

Suppose we have a very first generation of AM^2s embedded in an environment. They are able to perform a few actions, such as moving around in the environment. Since it is the very first generation, it is plausible to assume that their architecture is simpler than the architecture of the AM^2 described in the Section "Two-Machine Artificial and Their AbS". Their M2s have a finite set of symbols, and they do not delete the symbols they have already associated with states/meanings. Hence a symbol used for an association could be associated more than once either with the same M1's state or with different ones. Same AM^2s execute, in the same arbitrary way, the associating process. As we know from the Section "Two-Machine Artificial and Their AbS", every time M1 sends an internal state S_n to the M2, it is transduced at a given LoA and M2 then selects a symbol, say Sym_n, from a symbol source, associates it to S_n and stores this association in its memory. Before learning the performing rule, a M2 does not distinguish whether—and with which symbol—an incoming S_n has been already associated. Suppose that, after a finite number of runs, it turns out that an association between the same symbol and the same internal state has been used more than the other ones. According to Hebb's rule, the associations that are most used will be further privileged until they become stable. So that, when in the future M2 receives as input S_n, it will more readily associate S_n with Sym_n. In this way, AM^2s learn to associate a symbol with a meaning in a stable fashion and hence to execute the performing rule. The evolution of even rudimentary ways of grounding their symbols and hence of managing some basic communication will then further privilege and reinforce the selection of such AM^2s able to obtain the "right" symbols-states associations. Gradually, generations of more evolved AM^2s will be able not only to perform some of the steps required to apply the performing rule, but also to impose a social pressure on future AM^2s that grows exponentially, until new agents will start being selected in relation to their capacities to respond to old agents' semantically-oriented behaviours. At that point, the hardwired nature of the initial stages in the process of symbol-grounding may even become redundant, atrophy and disappear. One may object that Hebb's rule, or one like it, provides an extrinsic bias towards identifying the most rewarding behaviour—in our case, this is the development of stable transductions and associations between behaviours, internal states and symbols—and that therefore it breaches the Z condition. In order to answer this final objection, we shall refer to an evolutionary scenario simulated by running ELSA. This algorithm is well-known, so the reader already acquainted with it may wish to skip the following summary.

ELSA is derived from a realistic scheme of the evolutionary processes. It follows from algorithms originally motivated by Alife models of adaptive agents placed in ecological environments. ELSA's main feature is that the selection is *locally* mediated by the environment in which the AAs are situated. That is to say that the

fitness of AAs does not follow from global interactions across the whole population and the environment. Rather, the fitness is defined through the interactions between a singular AA and the environmental niche that the AA happens to inhabit. The environment biases the selection by managing the energetic resources, for it associates an energy bonus—which constitutes the selecting parameter—to every feature that the AAs may develop. The energy bonus assigned to any individual solution is in proportion both to the degree of the fitness of the solution and to the level of energy available in any zone of the environment. In this way, the environment can be considered as a data structure, which contains all the values assigned to each skill optimized by the AAs and keeps track of the actions of AAs. Two main aspects of ELSA need to be highlighted here.

First, the evolutionary process is independent of any external intervention. This defuses the previous objection. Running ELSA, the selection process is not performed according to some central bottleneck or predefined parameter; rather, the population changes depending on its interactions with the environment. The population's features are an intrinsic consequence of the environment's characteristics. This way, ELSA may be used to explain the Z-compliant use of Hebb's rule by a population of AM^2s. Suppose that, in some niches, the energy resources are set according to some instantiation of Hebb's rule. In such niches, the environment promotes those AM^2s able to follow Hebb's rule and hence to elaborate stable couplings of inputs and outputs. In so doing, the AM^2s do not appeal to any supervision from the programmer or from any other AA that is already semantically proficient; they just adapt to whatever bias is present in their environment. It follows that, in learning and performing Hebb's rule, they do not violate the Z condition. Moreover, since some fundamental biases are shared by most types of agents (think in biology of the famous three fs), it is literally natural that some functionally-similar types of eco-tuned AbS will evolve among different populations of agents.

Second, according to ELSA, the energy bonus is shared by the AAs developing the same feature in the same niche. So, the competition between the AAs is about the finite environmental resources and it is never across the whole population, but rather among the AAs situated in the same environmental area. Hence, the AAs have "interest" not only in achieving the best features, but also in finding the least populated zone in the environment where more energy is available. Thus, the population quickly distributes itself across the ranges given by the environmental features. This way ELSA encourages *coverage* and multi-modal optimization (all good solutions are represented in the population) rather than standard *convergence* (all individuals converging on the best solution). ELSA guarantees the natural implementation of a *heterogeneous* population, a feature that is pivotal for the solution of the SGP in view of a realistic account of the variety of groundings, and hence of semantics, that might become available across sub-populations.

So far, we have provided a solution to the two problems posed at the beginning of the Section "Two-Machine Artificial and Their AbS". Recall: an agent must be able (i) to associate symbols with the actions that it performs, without (ii) helping itself to any semantic resource in associating actions and symbols. It follows that, through the praxical approach, an AM^2 is able to develop some elementary semantic skills

while respecting the Z condition. Let us now see how an AM^2 would evolve its semantic abilities from the very first stage, described here, to a more complex one.

From Grounded Symbols to Grounded Communication and Abstractions

In the Section "Introduction: The Symbol Grounding Problem, it's Nature and Constraints", we outlined seven requirements that a strategy must satisfy to provide a valid solution for the SGP. So far, we have shown that the praxical strategy satisfies six of the seven requirements. In short, it allows the AM^2s to ground the meanings of the symbols in the data following the sensori-motor interactions between the AM^2 and the environment; the development of some sort of representations and abstraction capacities; the use of evolution in the development of the semantic skills and respect of the Z condition. We have still to prove that the praxical strategy enables the AM^2s to develop some sort of communication capacities among AAs in order to ground the symbols diachronically and avoid the Wittgensteinian problem of a "private language". So, in the remainder of this section, we shall describe how a population of AM^2s can develop more complex semantic abilities, such as communication and the elaboration of a shared lexicon, and thus satisfy the last requirement. We shall then rely on AA's communication abilities to show how AM^2s can overcome the problem of an impoverished semantics, anticipated in the Section "Two-Machine Artificial Agents and Their AbS"

Communication represents an invaluable achievement of a population of AAs for which coordinated social activity and the exchange of information provides highly adaptive benefits and may be vital for survival. Given such advantages, one can explain the development of communication and of a shared lexicon in a population of AM^2s as a result of natural selection and of the interactions among a population of AM^2s and between AM^2s and environment. We shall specify an evolutionary scenario in which such abilities would evolve.

Let us assume an environment in which the evolution is still a local evolution. Suppose we have a heterogenic population of AM^2s—made of both AM^2s able to elaborate only more specific meanings (SAM^2) and AM^2s able to elaborate only more general meanings (GAM^2). The AM^2s inhabiting a given niche interact with the environment in two ways: they feed and they can hide themselves to avoid the attacks of three kinds of predators—α, β and γ—which put them in three internal states (the reader will find more details about this scenario in Grim et al. (2001)). Suppose the AM^2s involved in this scenario engage in a kind of adaptive language game, such as the *guess game* (Steels 2005). A guess game is a technique used to study the development of common language in situated AAs. This kind of game involves two AAs situated in a common environment. Each AA involved in the game has a role: one is the *speaker*, and names the object that it perceives; the other one is the *hearer* and has to find out the objects named by the speaker by trial and error. The speaker communicates only to convey the name of a perceived referent, and the hearer communicates only to inform the speaker about its guessing concerning the referent named by the speaker. During the game, the AAs interact

and develop a common system of symbols. The game ends successfully if the two AAs develop a shared lexicon, grounded in the interaction among themselves and with the environment. In the case of the AM^2s described above, the communicated symbols are related to the speaker's internal states and, indirectly, to the action that it performs, for example 'open your mouth' or 'hide yourself' (using an observer-oriented description). Any time a symbol is communicated, the hearer performs one of these two actions. Since the actions are relevant to survival, the agents who perform the appropriate action—open their mouths and hide themselves when the communicated symbol indicates one of these actions—have a higher chance of surviving and hence reproducing than the ones which do not perform the right action. The agents that survive receive positive feedback from the environment, and they learn—through a Hebb-like rule—to associate that received symbol with the internal state related to the action that they perform. We can suppose that the hearer applies the storing rule to the received symbol. It performs a new association process, and the AM^2 stores in its memory the ensuing couple: the symbol received and its M1's internal state related to the action performed once the symbol has been heard (see Fig. 4). In the memory of the hearer's M1 internal states are associated both with the symbol it first used to name those states and with the new symbols communicated by the speaker.

In this way, the symbols communicated acquire a meaning also for the hearer and can be used by the AM^2s to develop a semantically grounded communication system. Since the same AM^2 interacts through different guess games with other AM^2s, the same symbol can become related to the internal states of different AM^2s—with different LoAs—among the population. Thus, the symbol communicated by the speaker ends up naming a set of similar states. Following this strategy, a shared lexicon can emerge through communications among a population of AM^2s. The shared symbols emerge according to use, and one can conclude that the most useful and hence recurrent symbols will be used as names of sets of similar states. For example, suppose a GAM^2 and a SAM^2 are involved in a guess game. We know the GAM^2 will use the same symbol to name all the states related to the attacks of

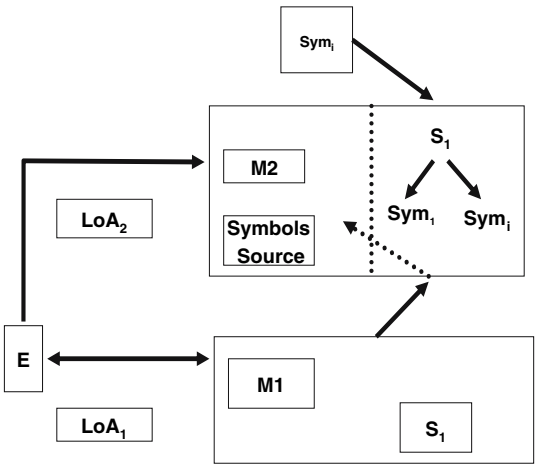

Fig. 4 Sym_i is the incoming symbol communicated by the speaker to the hearer. Once it has received the symbol the hearer will record it in its memory. Sym_i will be recorded together with the hearer's internal state and the symbol that the hearer firstly associated with that state

the predators α, β and γ. Suppose a GAM^2 communicates its symbol to a SAM^2. In order for the game to end successfully, the SAM^2 has to associate the generic symbol with one of his states related to the attacks of the predators, it does not matter which. Thus, a GAM^2's symbols acquire a meaning also for a SAM^2, since they are related to its internal states as well. In this way, the meanings elaborated by any AM^2 can be communicated among the population in the system. The semantics elaborated following the praxical strategy does not incur the problem of the private language. Generation by generation the AM^{2}s select recurrent symbols until they define a set of shared symbols that they all use as names of similar internal states.

Given how the AM^{2}s develop the shared lexicon it could be objected that the semantics generated by the praxical strategy reproduces the Wittgensteinian semantics of meaning-as-use and hence that it violate the Z condition (cf. section " Action-based Semantics"). But the AM^{2}s'communication abilities and the development of their shared lexicon, described in this section, follow from a different process and do not play any role in the process of semantically grounding of the symbols.

Semantically grounded communication develops when the AM^{2}s already use grounded symbols, and the SGP is solved before the AM^{2}s start to communicate with each other. The system needs to be jump-started somewhere within the virtuous circle, and the "where" is the relation between internal states and symbols in single members of the population, not their communication processes. In particular, one feature helps us to distinguish praxical semantics from the semantics of meaning-as-use. In the Wittgensteinian theory, meaning arises from the communications among the agents; the agents play in order to reach some agreement about the meaning. In praxical semantics, the meaning does not arise from the communication processes, but it is already defined, or at least well-sketched, when the AM^2 starts to communicate. What is shared in the communication process are the grounded symbols, not the meanings.

Consider now a last limitation of an AM^2's semantics. The elaboration of abstractions, described in the Section "Two-Machine Artificial Agents and Their AbS", causes an impoverishment of AM^2's semantics. In elaborating a general meaning, an AM^2 loses the specific meanings related to the symbols. Thus, it might be objected that the evolution of the praxical process generates a semantics composed of only very generic meanings, which tend to become even more generic. In our example, for an evolved AM^2, there would be only the meaning 'turning' and there would (or indeed could) be no distinction between meanings such as 'turning left' and 'turning right'. Moreover, the same agent also runs the risk of losing even the meaning 'turning' in favour of an even more generic 'moving'. The answer is that it is true that the semantics of a single AM^2 is bounded by the LoAs of that agent. And yet, this limit can be overcome when a whole population of AM^{2}s is taken into consideration. For we have seen that the ability to share semantically-grounded symbols to communicate among agents ensures that, through evolutionary pressure, the right (i.e. fit for survival) balance between generality and specificity of the semantics in question will be reached. In other words, it is the diachronic evolution of the population of agents that ensures the anchoring of otherwise possibly too-generically-grounded symbols to concrete usage in the real world.

Conclusion: From Grounding to Communicating

In this paper we have proposed a new solution for the SGP, analysing its possible developments and some of its limitations. The solution suggested has been called praxical to stress the interactions between agents and environment. The praxical approach is based on two main components: an Action-based Semantics (AbS) and the AM^2s, which implement the AbS thanks to their architecture. The two components allow the AAs to develop semantic skills sufficient to satisfy the seven requirements spelled out in the Section "Introduction: The Symbol Grounding Problem, it's Nature and Constraints" and hence to overcome the SGP without violating the Z condition. There are two points of view from which the praxical approach may be understood.

Technically, the praxical approach provides a solution to the SGP and describes a plausible and implementable model of AA. The architecture of AM^2 is based on the meta-programming paradigm, which is largely used to program AAs. And there are programming languages based on a framework that can already be interpreted in terms of AbS (Mosses 1992).

Philosophically, Sections "Learning the Performing Rule Through Hebb's Rule and Local Selection" and "From Grounded Symbols to Grounded Communication and Abstractions" show how AM^2s develop more complex semantic skills by combining the praxical approach and artificial evolution. We saw how a population of AM^2s could elaborate abstracted meanings and develop communications abilities and how they could grow a shared lexicon. This points towards a more ambitious and challenging perspective: the possibility of providing a theory of meaning based on praxical terms.

The distinction between symbols and meaning is a crucial difference between the praxical solution and the approaches reviewed in our previous work (Taddeo et al. 2005). Other attempts to solve the SGP consider meaning and symbol as two aspects of the same data. Thus, an AA is supposed to elaborate a set of perceptual data in order to obtain a representation which is both the meaning and the symbol that is then used to name that very representation. On the contrary, our praxical solution treats meaning and symbol as two kinds of independent data: the first one is given directly every time an AM^2 interacts with the environment, whereas the second is produced by M2. Only at the end of the process does an AM^2 couple them together. This allows the AM^2s to respect the Z condition: since there is no need for a process through which meaning must be elaborated, there is also no need for any extrinsic criteria required to guide the elaboration of meaning. Still, the semantics elaborated by the AM^2s has a certain lack of completeness and complexity. AbS provides a minimal and simple semantics. It is patently not truth-functional nor does it justify the elaboration of meanings through some explicit agreement among the agents. We have seen that it is also far from being Wittgensteinian. AbS and the praxical approach more generally define a semantics that is simple and elementary enough to be developed autonomously by AAs. This is a semantics that is compatible with AAs' features and hence, in this sense, it is *non-anthropocentric*. However, we have shown in the Section "From Grounded Symbols to Grounded Communication and Abstractions" that the complexity of AM^2s' semantics can be escalated through

evolutionary and social processes, to the point when it allows the AM^2s to develop communication skills and create a shared lexicon.

The possibility of the evolution of language skills in a population of agents through social pressure has been described in Smith et al. (1999). "When we meet a linguistic novelty we do not give up too easily: we try to guess the meaning by watching others, as well as trying it out of ourselves. [...] (the meaning) must be built on pre-existing neuronal structures", (Smith et al. 1999, p. 165). We agree completely. Going back to the AM^2 population, and considering a generation of AM^2s already provided with semantic skills, it turns out that further semantic elaborations are greatly facilitated and improved by social interactions among AM^2s. So we can suppose that AM^2s acquire the performing and storing rule through a 'genetic assimilation learning' (Pinker 1994). Through this process, a learnt behaviour is converted (replaced) into one that is genetically programmed. More specifically, in a generation of AM^2s, in which the performing and storing rules are genetically assimilated, the meaning no longer has to be directly related to the interactions between an AM^2 and the environment, but can be based on interactions with other AM^2s.

We believe that the solution of the SGP offered in this paper provides the seeds for an interesting explanation of how advanced semantic and linguistic skills develop among biological agents in natural environments. Yet these implications of the praxical approach have been only briefly sketched here, for they are left for a further and last stage in our research.

Acknowledgements A first draft of this paper was the topic of a seminar given by one of us (Mariarosaria) at the Department of Philosophy at University of Padua and we are grateful to the participants for their helpful discussions. We would also like to thank Massimiliano Carrara, Roberto Cordeschi and Jeff Sanders for their suggestions and comments on several versions of this paper. We are very grateful to the members of our research group on the philosophy of information, the IEG, at the University of Oxford for their useful comments; in particular, we would like to acknowledge the help and the very valuable feedback by Sebastian Sequoiah-Grayson and Matteo Turilli. Filippo Menczer very kindly provided many suggestions about the use of ELSA and Local Selection in solving the Symbol Grounding Problem. None of the people listed above is responsible for any remaining mistake.

References

Barklund, J., et al. (2000). Reflection principles in computational logic. *Journal of Logic and Computation, 10*, 743–786.

Brazier, F. M. T., et al. (1999). Compositional modelling of reflective agents. *International Journal of Human-Computer Studies, 50*, 407–431.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems, 6*, 3–15.

Cointe P. (Ed.) (1999). *Meta-Level architectures and reflection, second international conference on reflection*. Saint-Malo, France: Springer-Verlag.

Donahoe J. W., et al. (Eds.) (1997). *Neural network models of cognition: Biobehavioral foundations.* Amsterdam: Elsevier Science Press.

Floridi, L. (2004). Open problems in the philosophy of information. *Metaphilosophy, 35*, 554–582.

Floridi, L., et al. (2004). The method of abstraction. In M. Negrotti (Ed.), *Yearbook of the artificial, dedicated to "models in contemporary sciences"* (pp. 177–220). Berna: P. Lang.

Grim, P., et al. (2001). Evolution of communication with a spatialized genetic algorithm. *Evolution of Communication, 3*, 105–134.

Harnad, S. (1990). The symbol grounding problem. *Physica, D*, 335–346.

Hebb, D. O. (1949). *The organization of behavior: A neuropsychological theory*. New York: John Wiley & Sons.

Menczer, F., et al. (2000). Efficient and scalable pareto optimization by evolutionary local selection algorithms. *Evolutionary Computation, 8*, 223–247.

Menczer, F., et al. (2001). Evolving heterogeneous neural agents by local selection. In M. Patel, et al. (Eds.), *Advances in the evolutionary synthesis of intelligent agents* (pp. 337–366). Cambridge, MA: MIT Press.

Mosses, P. D. (1992). *Action semantics*. Cambridge University Press.

Pinker, S. (1994). *The language instinct*. New York: William Morrow.

Real, L. A. (1991). Animal choice behavior and the evolution of cognitive architecture. *Science, 30*, 980–985.

Smith, J. M., et al. (1999). *The origins of life*. Oxford University Press.

Steels, L. (2005). The emergence and evolution of linguistic structure: from lexical to grammatical communication systems. *Connection Science, 17*, 213–230.

Taddeo, M., et al. (2005). Solving the symbol grounding problem: A critical review of fifteen years of research. *Journal of Experimental and Theoretical Artificial Intelligence, 17*, 419–445.

Varshavskaya, P. (2002). Behavior-based early language development on a humanoid robot. In C. G. Prince, et al. (Eds.), *Second international workshop on epigenetic robotics: Modelling cognitive development in robotic systems* (pp. 149–158). Edinburgh, Scotland.