# Symbol Grounding and its Implications for Artificial Intelligence

**Michael J. Mayo**
School of Information Technology
Bond University, Gold Coast, Qld 4229
Australia

mmayo@bond.edu.au

## Abstract

In response to Searle's well-known Chinese room argument against Strong AI (and more generally, computationalism), Harnad proposed that if the symbols manipulated by a robot were sufficiently grounded in the real world, then the robot could be said to literally understand. In this article, I expand on the notion of symbol groundedness in three ways. Firstly, I show how a robot might select the best set of categories describing the world, given that fundamentally continuous sensory data can be categorised in an almost infinite number of ways. Secondly, I discuss the notion of grounded abstract (as opposed to concrete) concepts. Thirdly, I give an objective criterion for deciding when a robot's symbols become sufficiently grounded for "understanding" to be attributed to it. This deeper analysis of what symbol groundedness actually is weakens Searle's position in significant ways; in particular, whilst Searle may be able to refute Strong AI in the specific context of present-day digital computers, he cannot refute computationalism in general.1

*Keywords*: Chinese room argument, symbol grounding, artificial intelligence.

## 1 Introduction

Will a machine ever be able think and understand in the same way that a human mind thinks and understands? Aside from debates over definitions, this is a question that has dogged AI scientists and philosophers for a number of years. With the advent of digital computer technology, the question has tended to focus specifically on digital technology: could a computer, solely by virtue of running the correct program, think and understand? The affirmative answer to this question is the stance known as Strong AI (Searle, 1980), which has its roots in assertions by famous AI researchers such as Newell & Simon (1976) that intelligence resides in physical symbol systems.

It is no wonder, then, that John Searle created a storm of controversy when he published his now (in)famous Chinese room argument (CRA) against Strong AI (Searle 1980, 1990). More recently, Harnad (1993, 2001) has shown how the CRA can be reformulated as an attack on computationalism, a more general position than Strong AI, which holds that mental states are equivalent to implementation-independent computational states (i.e. states independent of the underlying hardware or substrate). Searle (1993) agrees that the CRA also holds for computationalism.

The CRA argument is essentially this. Programs are syntactic. That is, they consist solely of symbol strings. The shape of a symbol is arbitrary (not related to its meaning or content), and the rules for reasoning (combining and recombining the symbols) are themselves arbitrary symbol strings. The human mind, on the other hand, is both syntactic and semantic: the human mind has symbols, but it also attaches a meaning (semantics) to its symbols. Thus, a human mind can be said to *understand*. Searle states that because no amount of syntax will ever produce semantics, there is no way that a computer running a program (which is purely syntactic) will ever be able to understand. It follows that Strong AI is false.

To "set the reader up" to accept this conclusion, Searle makes use of a particularly effective thought experiment (or "intuition pump" as Dennett, 1987, calls it). Imagine a man (or a flea, or a machine) inside a room. The man is passed, through a slit in the wall, pieces of paper upon which are written sequences of Chinese symbols. The man understands no Chinese at all, and therefore has no comprehension of what the symbols on the paper actually mean. Yet, with the help of a rulebook specifying the correct matching response for every possible string of Chinese input characters, he is able to write a response in Chinese and pass it back through the slit to the outside world. To the native Chinese speakers who are outside the room, it is as if they are conversing (via written Chinese words) with another native Chinese speaker. The question is, does the man (or the flea, or the machine) understand anything? Searle says no, since the man is merely following the mechanical instructions in the rulebook that allow the man to simulate a native Chinese speaker. And by following the rulebook, the man is doing essentially the same thing as digital computer.

It is my intention to focus on a single significant reply to the CRA, namely Harnad's notion of *symbol grounding* (Harnad, 1990, 1993). By elaborating on the notion of symbol groundedness in three ways, I will show that Searle's CRA is considerably weakened. In particular, I claim that Searle's argument relies significantly on features of present-day digital computers that a sufficiently grounded computer (or

[1] Copyright © 2003, Australian Computer Society, Inc. This paper appeared at the *Twenty-Sixth Australasian Computer Science Conference (ACSC2003)* in Adelaide, Australia. Conferences in Research and Practice In Information Technology, Vol. 16. Michael Oudshoorn, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

robot) would not necessarily possess; thus Searle's argument works against present-day computers, but cannot refute computationalism in general.

Symbol grounding, stated simply, is an attempt to show that computer programs can have semantics. "Grounding" a symbol means taking the symbol, and associating it with a pattern of sensory data that is perceived when the entity that the symbol denotes is seen, or heard, or tasted etc. For example, I understand what the symbol "pizza" means because I know what a pizza looks like, smells like, tastes like and so on. This sensory information is called its *iconic* representation. Icons are analog rather than digital, because their form reflects their content. If a computer acquired the icon for a pizza, then according to Harnad, it too could understand the concept "pizza". Of course, the computer ideally should be able to learn about pizzas all by itself: it therefore would be more like a robot than a standalone computer, having a host of sensorimotor apparatus connected to it such as video cameras to see the pizza, artificial taste and smell receptors for tasting and smelling the pizza, artificial hands to pick up and touch the pizza etc. In this way, the symbol "pizza" would become grounded in the real world, and the robot would understand. Note that Harnad uses the term *category* to refer to the name of the thing that a symbol denotes, but a category is not a symbol; thus, a grounded symbol has both categorical and iconic representations.

A feature of the groundedness of symbols is that they can be easily combined. For example, if I have never seen a zebra before, but I do have the grounded symbols "horse" and "stripes", then I can construct the iconic representation for "zebra" from the icons for "horse" and "stripes". In this way, I would be able to recognise a zebra when (and if) I ever see one.

## 2 An Elaboration of Symbol Grounding

There are three issues that Harnad does not address in relation to his symbol grounding scheme, which I intend to discuss here.

### 2.1 Where do grounded symbols come from?

This first issue concerns how grounded symbolic representations could arise in a mechanical or biological mind initially. Bear in mind that there is a basic dichotomy between sensory data, which is continuous and constantly fluid, and symbols, which are essentially discrete and static. It has often been asked how the mind (be it human or robot) could ever be able to categorise the massive in-flow of sensory data that it receives in order to sort out the icons from the noise, and furthermore, how it can do this in real-time. I am going to ask a much more fundamental question: how does the mind induce grounded symbols (complete with categories and iconic representations) in the first place? Consider, for example, a child (or a newly built, “intelligent” robot). A child starts out in the world with no grounded symbols because groundedness can only arise as a consequence of experiences, and a child initially has no experience of the world. Admittedly, the child may possess a handful of innate responses (such as the ability to react in a certain way to a visual pattern resembling a face), but such autonomic reflexes do not constitute any form of understanding. Piaget, in his theory of cognitive development, states that a child is initially unable to distinguish even between self and environment (Wadsworth, 1989). In other words, the child must not only learn to associate icons with categories to form grounded symbols, but it must also come up with a grounded symbol that denotes the child’s own self as distinct from everything else.

Now, sensory data begins flowing into the child's brain via the eyes, ears, etc. There are no clean divisions in the sensory data that could be used to learn the icons. If there are divisions (e.g. discontinuities in intensity gradients), which ones are significant? To illustrate the problem, consider how difficult it must be to identify the leaves and the fruits of an apple tree in the absence of any a priori iconic information about what a leaf or an apple should look like.

Now consider this: without some sort of bias, it is computationally intractable to come up with the best set of categories describing the world. What do I mean by this? Given that sensory data is continuous, there is an effectively infinite (subject to the limits of the sensory apparatus) number of possible categorisations of the data. It is simply impossible to mechanically search this space of categorisations to find the one that best fits the perceived reality. To give a simple example, suppose I have a camera that can detect shades of grey to an arbitrary degree of accuracy, from pure white to pure black and everything in between. What is the best way to divide this continuum of stimuli up into discrete categories? I could halve the spectrum, for example, by assigning the lightest half to the WHITE category and the darkest half to the BLACK category. Alternatively, I could divide it into three categories, namely WHITE, GREY and BLACK. Or I might decide to divide it into four or more categories. Note that the categories need not all have the same width; I might assign a wider region of the range to the category GREY, for example. The key point is that the number of different categorisations is effectively infinite. So how can a robot or child ever find the best one, and what is the criterion for selecting a particular categorisation anyway? Scale the problem up to realistic proportions and you have to deal with sensory data that is not continuous over a single dimension like shades of grey, but is continuous over multiple dimensions. Furthermore, some of those dimensions will be visual, others auditory, and so on. A brute force search over all the possible different categorisations of this data (in order to find the one that best benefits the robot or child) is simply impossible due to the sheer size of the search space.

It should be noted that this is a difficult problem to solve. AI and Artificial Life researchers often skirt over the issue by building systems and simulations in which the input data is assumed to be “automatically” symbolic.

So how can a child learn symbols and categories, and what can this tell us about how a grounded robot might learn its symbols? My argument is that initially symbols and their categories are grouped into *task-specific* sets. Task-specific means that the symbols are formed in order to solve specific problems in particular domains. By having a specific task to perform, a bias is provided for the problem of searching for the best categorisation of sensory data. The bias is simply that the symbols learned are those that make the solving of the task as simple as possible, and all other categorisations are ignored. For example, consider an entity that is learning to hunt. It will clearly need symbols denoting at least its prey and any obstacles; these symbols would belong in the task-specific set for hunting. It can then reason symbolically about what actions it needs to take in order to catch the prey. The number of categories relevant to each task is likely to be quite small as well, which may explain how minds and robots can identify and categorise sensory data so quickly: they are only searching for symbol icons relevant to the task at hand.

An implication of task-specificity is that the same sensory data will be categorised in different ways depending on the context. For example, in one task context some visual data may match the prey category (that being a single category), but in a different task context, the same data may be further subdivided into categories such as head, torso, arms, legs etc, because these additional symbols are required to effectively carry out the different task. So categories are grouped into sets where each set is relevant to a particular task. A mind or intelligent, grounded robot learns its symbols set-by-set, as it successively masters task after task.

## 2.2 Can symbols denoting abstract concepts be grounded?

The second issue is about abstract concepts. What does it mean to say that an abstract concept (such as "love", "politics", or "victory") is grounded? Or does Harnad's solution apply solely to concrete concepts (such as "cat", "pizza", and "house")? Harnad himself recognises this as a problem (Harnad, 1993). I want to show that the notion of grounded symbols denoting abstract concepts is meaningful, and furthermore, it arises naturally if one accepts that initially symbols are grouped into task-specific sets.

The first thing to note is that task-specific sets of symbols may overlap. That is, pairs of symbols from disparate sets may actually denote the same or a partially similar concept or thing, because their icons contain similar information. For example, consider the games of chess and tennis. The notion of "victory" in each game is a concrete concept, and so I would expect a "victory" symbol in both the task-specific set defined for chess, as well as the task-specific set defined for tennis. The iconic representation of the former would include (perhaps) an icon of the chessboard in a victory configuration, while the iconic representation of the latter would include information for determining if a game of tennis has been won. But both icons would contain a great deal of similar information as well. For example, the sensation of victory one feels when one is victorious in a game of chess or tennis, whether it be happiness, satisfaction or exhilaration, is similar.

So how does the abstract notion of "victory" arise, one that is not connected with any particular instance of victory? It is by a process of generalisation or decontextualisation, in which a new symbol, outside of the task-specific sets, is learned. The new symbol's iconic representation is precisely the iconic information that is shared by all the specific instances from which it derives. In other words, the abstract "victory" symbol has an icon that is the intersection of the icons of victory in each specific context, such as chess, tennis, and so on. Another example is the formation of the abstract notion of "love". A child learns that *Mummy loves Daddy* and *Uncle loves Aunty*, and from the similar parts of the iconic representations of these different, concrete categories, it can learn the abstract symbol "love".

This idea makes sense from a computational viewpoint. Intersecting icons is a way of minimising the amount of data to be stored. It is analogous to the idea of the compression algorithm from Computer Science (e.g. Witten et al, 1999), in which redundancy is eliminated by replacing repeating patterns with references to their first occurrence. It just so happens that as a by-product of this intersection process, new, useful, and abstract concepts can be formed.

So abstract symbols can be grounded; the primary difference between grounded abstract symbols and grounded concrete symbols is that the former are associated with only a partial icon that arises through a process of decontextualisation.

## 2.3 When is a robot sufficiently grounded for it to be considered intelligent?

The third and final discussion in this elaboration of symbol grounding concerns the criterion for intelligence. How grounded should a robot become before it can be said that it has an intelligent mind? The traditional test of intelligence is the Turing Test (Turing, 1950) or one of its variants (e.g. the Total Turing Test discussed in Harnad, 2001). But the Turing Test is purely behaviouristic, asserting that intelligence and understanding is an attribute assigned subjectively by an observer to an entity, rather than being an inherent property of the entity itself. The CRA is designed to show that the Turing Test is insufficient - a robot could simulate intelligence and pass the test, but still not really understand anything. Harnad's reply is to make assertions about how an intelligent robot's mind would be organised cognitively and how it could operate - specifically, its symbols would be grounded. So Harnad has effectively brought the debate into the arena of cognitivism, which in turn means that the Turing Test is no longer completely sufficient for attributing understanding, because hypothetically, a machine might meet Harnad's requirements but fail the Turing Test.

I propose a new test for determining when a robot's brain would be sufficiently grounded in order to constitute it being called a mind. The basis of the test rests on the observation that physical symbol systems (Allen & Newell, 1976) consist of both symbols denoting things in the world, and separate rules for meaningfully combining them. The rules may themselves be symbols, but the fact is that there is a clear distinction.

I claim that in a sufficiently grounded robot there would be no need for explicit, separate rules for combining symbols. The reason for this is as follows: the robot has a rich, iconic representation of each of its symbols. It should therefore be able to infer from the icon and the icon alone when it is meaningful to combine those symbols and when it is not. In other words, the symbols, by virtue of their groundedness, can be manipulated intrinsically without any distinct and artificial rules of composition being defined. This could serve as the starting point for a definition of understanding.

To illustrate the point, a robot that understands "horse" and "stripes" should, by analysing the icons of these symbols, admit the possibility of a meaningful new category being produced by their combination (irrespective of whether such a thing really exists in the world). It should likewise admit the impossibility of combining "pizza" with, say, "victory". And the robot should be able to do this solely by reasoning from the iconic representations of its symbols. So, effectively, thanks to groundedness, we no longer need distinct rules of composition, taxonomies or class hierarchies of symbols, and so on. A mind that understands will consist of meaningful, grounded symbols - and that is all.

Thus we have a new test for intelligence: examine the robot's program and determine if the robot can reason symbolically solely by virtue of the iconic information associated with its symbols. Is the iconic information sufficiently rich to support such "intrinsic" reasoning? If it is, then it passes the test. On the other hand, if the program uses rules external to the symbols themselves, rules that prescribe the way in which symbols should be combined, then the robot is not utilising the iconic information associated with the symbols. Instead, it is following mechanical rules, so therefore it fails the test.

Why are mechanical, prescriptive rules so bad? Researchers in the expert systems community would immediately be able to offer an answer to this question: mechanical rules inevitably lead to brittleness. Brittleness refers to the inability of a supposedly intelligent system to adapt to a slightly different task for which it was designed, without a major reprogramming effort. Humans are quite the opposite. To illustrate, the famous chess-playing computer Deep Blue (Campbell et al, 2002) is unlikely to be able to win a game of draughts. Similarly, a medical system designed to diagnose one type of cancer is unlikely to be able to diagnose cancer of another type, even though much of the domain knowledge may overlap. And the reason for this is simply that there is no real "understanding" in the system: all that exists are mechanical rules that are followed. The symbols are not grounded, and cannot therefore be used for purposes other than that for which they were designed. (It should be noted that if a gap or error occurs in these mechanical rules, then the system may exhibit some profoundly silly behaviour that betrays its lack of understanding. Some authors have referred to this as "artificial stupidity".)

A possible counter-argument to this stance may be to say that the rules could be generated on-the-fly using machine learning algorithms. The system would thus be able to adapt to new tasks. I think that the major flaw with this argument is that the no matter what machine learning algorithm is used, the number of possible rules that could be generated by any given algorithm will always be limited. And this must in turn lead to the brittleness problem. The only solution is avoid external rules altogether, and this is the test for intelligence that I am advocating. External rules are not needed if the icons are sufficiently rich.

The test proposed here is certainly not designed to displace the Turing Test. Rather, it enhances the Turing Test. The Turing Test is behaviouristic, but Harnad's reply to the CRA makes cognitive assertions about intelligent robots and minds. We therefore need to account for this cognitive dimension, and this test does just that.

## 3 Implications for the CRA

Now that Harnad's original concept of symbol grounding has been extended and analysed, what are its implications for Searle's CRA? Is the CRA still a convincing argument when confronted with a robot whose symbols are richly grounded, and meaningfully combined without resort to separate rules of manipulation?

Unfortunately (for Searle), the plausibility of the CRA rests quite strongly on the distinction between the symbols and the external rules that manipulate them:

the "man" in the Chinese room uses rules to shuffle the symbols in and out through the slits. But I am suggesting that in a properly grounded robot, there would be no such rules. There would be no rulebook that the man could look up with which to manipulate the symbols. Instead, there would just be the symbols and their rich iconic representations. The man would be able to manipulate the symbols because he has enough information within the icons to "understand" them. So the concept of a pure syntactic manipulation of symbols does not apply to a grounded robot; rather, in such a robot, the semantics (the icons) will play a large part in how the symbols are combined. This is one argument undermining the CRA.

The CRA is weaker still when one considers that Searle uses a neat trick to lend plausibility to his argument. The trick relies on generating confusion in the minds of the readers between *implementation-level* symbols and *referential-level* symbols. Implementation-level symbols are determined by the hardware that the program will execute on. For example, implementation-level symbols in a digital computer include bits such as 1 and 0, fundamental instructions like SHIFT and COMPARE, and so on. They are precisely the symbols that the man in the Chinese Room is meant to process. Referential symbols, on the other hand, denote things in the world such as "pizza", "zebra", "cat" etc, which have been the focus of this discussion. Such symbols are not determined by the hardware at all, and so they are "implementation-independent" in the computationalist sense. Searle's argument, in fact, is that a digital computer manipulating symbols at the implementation-level cannot understand. That, I believe, is a valid assertion. However, Searle makes an error when he claims that his conclusion holds true of any type of symbol manipulation - which implicitly includes referential symbols as well. He gets away with it because he never distinguishes between the implementation and referential levels in the first place, and consequently many readers are "persuaded" by his conclusion that *all* types of symbol manipulation must be mechanical, syntactic, and therefore non-intelligent.

I find this conclusion somewhat ironic because symbolic AI scientists do actually try to develop programs at the referential level; models of cognition, especially classic AI programs, focus a great deal on defining and manipulating referential symbols, and do not in theory depend on the implementation level. Of course, the flaw in classical symbolic AI is that the referential symbols are not grounded, but are defined in terms of other referential symbols (the Chinese/Chinese dictionary-go-round, as Harnad, 1990, calls it), which leaves the door wide open for a CRA-type argument. But in general, the symbolic AI scientist would not give consideration to whatever implementation symbols their particular platform compiles their program into. Yet Searle attacks implementation-level symbol manipulation quite specifically.

What the notion of symbol grounding does suggest, in respect of the implementation/referential distinction, is that a computer could be developed that operates solely at the referential level. Unlike a present-day digital computer, which can manipulate only implementation-level symbols, the referential computer would have as its most primitive element symbols that are grounded in the world. Its sensory apparatus, and even its iconic memory, would need to be primarily analog. Rather than converting sensory data first into a stream of bits, it would operate directly on the sensory data itself; matching data to category (which, as I have argued, will depend on the current task-specific context), and pulling the corresponding iconic representations into its analog working memory. Thus the memory of the grounded robot would be an iconic representation of the world itself, at a particular level of detail determined by the task it is trying to solve. Once the symbols and their icons are drawn into memory, and after the sensory data has been categorised, reasoning then becomes the process of manipulating the symbols by virtue of their icons. In other words, a sufficiently rich iconic representation of a situation in a robot's memory should suffice for understanding and problem solving.

At this stage it is premature to speculate about how such a system might be implemented. Harnad has argued for many years that neural networks offer the best means of achieving grounded representations (e.g. see Harnad, 1993, and many of his other papers). However, I believe that the main flaw of his proposal is that he only suggests using neural networks for storing the iconic information associated with the symbols. The symbols themselves are still manipulated by external rules, which as I have argued, should not be necessary if the icons are sufficiently rich. Whether neural networks are sufficiently flexible to allow the type of processing I have described here is an open question for future research. It may be the case that they are not, but perhaps some new emerging technologies such as bio-computers (e.g. Garcia et al, 2002) or analog/associative hardware does offer some hope.

So, could Searle's CRA apply to a grounded robot as described in this paper? The answer is no. Clearly, you can no longer draw an analogy between the man in the Chinese room and the CPU of such a computer. The CPU of a traditional digital computer operates at the implementation-level, where the form of the input is not related to what is denoted. The machine I am proposing, however, is quite different: it operates on the referential level where the form *is* related to the content at the lowest level (and that level can vary depending on the task at hand). It cannot thus be argued that the CPU or man in the Chinese room is merely shuffling meaningless symbols. Searle’s CRA is therefore inapplicable to such a machine.

## 4 Conclusion

To summarise, Harnad's symbol grounding idea has led to the following tenets about what constitutes intelligence.

Firstly, an intelligent robot should acquire its symbols as it learns new tasks. The task essentially provides a bias for distinguishing the icons of each symbol from everything else in the sensory input.

Secondly, although concrete symbols are initially compartmentalised according to the task, later they are generalised out of their initial task-specific contexts by a process of intersecting (or compressing) the icons, thus forming abstract symbols.

Thirdly, a consequence of this is that the icons for each symbol should be sufficiently rich to allow symbolic reasoning without the use of external, prescriptive rules. This provides the basis for a new test of intelligence.

Fourthly and finally, these new tenets have proved incompatible with the notion that the CRA can refute computationalism in general. The CRA only holds against machines like present-day digital computers that operate with implementation-level (as opposed to referential-level) symbols. Note that it does not actually matter whether or not one believes that the variety of robot discussed here really could understand; the point is that Searle's argument cannot deny the possibility, which is what it was originally designed to do. And because such a robot is still basically a machine running a program, computationalism is still valid.

## 5 References

CAMPBELL, MURRAY A., HOANE, JOSEPH and HSU, FENG-HSIUNG. (2002). Deep Blue. *Artificial Intelligence* **134**(1-2): 57-83.

DENNETT, D. (1987) Fast thinking. In Dennett D., *The Intentional Stance*, 323-337. MIT Press.

GARCIA, PAUL S., CALABRESE, RONALD L. DEWEERTH, STEPHEN P., and DITTO, WILLIAM. (2002) *Simple Arithmetic with Firing Rate Encoding in Leech Neurons: Simulation and Experiment* (Pre-print). Available on the WWW at http://www.neuroengineering.com/publications/111180.pdf

HARNAD, STEVAN. (1990) The symbol grounding problem. *Physica D* **42**: 335-346.

HARNAD, STEVAN. (1993) Grounding symbols in the analog world with neural nets. *Think* **2**: 12-78.

HARNAD, STEVAN. (2001) Minds, machines and Searle 2: What's right and wrong about the Chinese Room Argument. In Bishop, M. & Preston, J. (eds.) *Essays on Searle's Chinese Room Argument*. Oxford University Press.

NEWELL, A. and SIMON H. A. (1976) Computer Science as empirical enquiry: symbols and search. Communications of the ACM **19**. Reprinted in Boden M (Ed.) (1990) *The Philosophy of Artificial Intelligence*, 105-132. Oxford University Press.

SEARLE, JOHN R. (1980) Minds, brains and programs. *Behavioural and Brain Sciences* **3** (3): 417-457.

SEARLE, JOHN R. (1990) Is the brain's mind a computer program? *Scientific American* **262**(1): 20-25.

SEARLE, JOHN R. (1993) The failures of computationalism. *Think* **2**: 12-78.

TURING, ALAN M. (1950). Computing machinery and intelligence. *Mind* LIX no. **2236** 433-60. Reprinted in Boden M (Ed.) (1990) *The Philosophy of Artificial Intelligence*, 40-66. Oxford University Press.

WADSWORTH, BARRY J. (1989). *Piaget's theory of cognitive and affective development.* Longman, New York.

WITTEN, IAN H., MOFFAT, ALISTAIR, and BELL, TIMOTHY C. (1999) *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, San Francisco.