Evolution and Modularity: The limits of mechanistic explanation

Jaakko Kuorikoski (jaakko.kuorikoski@helsinki.fi)
Social and Moral Philosophy, P.O. Box 24
University of Helsinki, 00014 Finland

Samuli Pöyhönen (samuli.poyhonen@helsinki.fi)
Social and Moral Philosophy, P.O. Box 24
University of Helsinki, 00014 Finland

Abstract[1]

Accounts of mechanistic explanation require that complex cognitive phenomena can be decomposed into simpler subtasks. We provide a theory of explanation that rationalizes this requirement, and then we use a simple genetic algorithm exercise to demonstrate that evolution can produce designs that violate this functional modularity requirement.

Keywords: mechanism; explanation; evolution; modularity; genetic algorithm

Introduction

Connectionism, dynamical systems theory, and new robotics have questioned whether the search for information-processing mechanisms provides a feasible approach to the study of biologically evolved cognitive systems such as the human mind. Whereas approaches that have their origins in classical AI tend to conceive of cognition as a set of computational operations to be mapped onto physiological parts according to functional decompositions inspired directly by the programmer's intuitions about possible efficient subroutines, the alternative research programs emphasize that biological evolution is likely to produce unintuitive designs of such complexity that renders heuristics based on decomposability and programming intuitions unusable.

In this paper we analyze the problems that evolved solutions raise to the mechanistic understanding of cognitive phenomena. The problem of understanding non-intuitive designs produced by natural selection is well-known in philosophy of psychology (e.g., Clark 1997, Ch. 5), philosophy of biology (Wimsatt 2007), and now even in popular psychology (Marcus 2008), but it has proved to be difficult to articulate without a clear idea of what exactly it is that evolutionary tinkering is supposed to hinder. The main challenge for scientific understanding is often framed and explained by pointing to the path-dependent nature and the resulting unfamiliarity of the evolved design (Jacob 1977). We argue that this is not the whole story. The aim of this paper is to provide an explicit theory of mechanistic explanation and understanding that will move us beyond intuitions towards a more systematic analysis of the nature of these challenges. We also combine our theory of explanation with a computational application of evolutionary design: problem-solutions generated by genetic algorithms. By analyzing the nature of solutions that genetic algorithms offer to computational problems, we suggest that evolutionary designs are often hard to understand because they can exhibit *non-modular functionality*, and that this creates problems for strategies of mechanistic explanation.

Mechanistic Explanation in the Cognitive Sciences

According to the proponents of the mechanistic approach to explanation (Bechtel 2008; Craver 2007; Piccinini & Craver 2011), a central goal of the cognitive sciences is to provide understanding of system-level properties of the cognitive system in terms of the properties of its physical component parts and their organization. The most developed philosophical account of strategies for reaching such mechanistic understanding is Bechtel and Richardson's (2010) study of the *heuristics of decomposition and localization* (DL). The DL procedure goes roughly as follows. First, the different phenomena that the system of interest exhibits are differentiated. Then the phenomenon of interest is *functionally decomposed*, i.e., analyzed into a set of possible component operations that would be sufficient to produce it. One can think of this step as the formulation of a preliminary set of simpler functions that, taken together, would constitute the more complex input-output relation (the system-level phenomenon). The system is also s*tructurally decomposed* into a set of component parts. The final step is to try to *localize* the component operations by mapping them onto appropriate structural component parts. If this cannot be done, the fault may lie with the functional and structural decompositions or with the very identification of the phenomenon, and these may then have to be rethought. The identification and decomposition procedures will in the beginning be guided by earlier theories and common sense, but empirical evidence can always suggest that a thorough reworking of the basic ontology and the form of the possible *explananda* may be in order.

What the schema of Bechtel and Richardson lacks is an explicit theory of explanation providing an account of what makes such decomposition and localization exercises explanatory. Whereas cognitive theories of explanation (Churchland 1989; Thagard 2012; Waskan 2006) focus on the internal models and processes of the individuals engaged in explanation-related tasks, such conceptualization is

[1] The authors are listed in alphabetical order. This paper is based

misleading when thinking about the *goal* of research: understanding. We use the term ‘understanding’ in order to shift our focus from single explanations to a broader collective epistemic goal. Scientific understanding proper is not what happens inside individual heads, but is constituted by the collective abilities of the scientific community to reason about and to manipulate the objects of investigation. To conceptualize scientific understanding directly based on models of individual explanatory cognition is to commit a fallacy of composition.

We therefore approach understanding as a public, behavioral concept. Understanding is a regulative label, which is attributed with regard to manifest abilities in action and correctness of reasoning. Suitable cognitive processes (comprehension), and possibly the possession of right mental models, taking place in the privacy of individual minds, are a *causal* prerequisite for possible fulfillment of these criteria, but these processes themselves are not the facts *in virtue of which* something is understood or not. They are not the criteria of understanding in the sense that we would have to know them in order to say whether somebody *really* understands something. The *correctness* of internal mental models is judged according to manifest cognitive performance, not the other way round (Ylikoski & Kuorikoski 2010).

We take the principal criterion of understanding to be inferential performance: whether someone understands a phenomenon is assessed based on whether he or she can make correct inferences related to it. Thus our view of understanding can be linked to Woodward’s (2003) widely accepted account of scientific explanation, which tells us more specifically *what kinds of inferences* are constitutive of specifically explanatory understanding (see also Craver 2007). Explanation consists in exhibiting functional dependency relations between variables. This is the connection between explanation and understanding: knowledge of explanatory relationships grounds understanding by implying answers to what-if-things-had-been-different questions concerning the consequences of counterfactual or hypothetical changes in the values of the *explanans* variable. This is the important difference between explanatory information and purely descriptive information. Whether someone understands a phenomenon is evaluated according to whether he or she can make inferences not only about its actual state, but also about possible states of the phenomenon or system in question.

Modularity and Understanding

According to Bechtel and Richardson, decomposability is a regulative ideal in mechanistic model construction because complex systems are psychologically unmanageable for humans. Decomposition allows the explanatory task to be divided into parts that are manageable for cognitively limited beings, thereby rendering the system intelligible (Bechtel & Richardson 2010). The idea comes originally from Simon (1962), who claimed that complex systems have to be *nearly-decomposable* in order to be understandable for finite cognitive agents. Near-decomposability means that the system can be decomposed into parts in such a way that the intrinsic causal properties of the parts are more important for the behavior of the system than their relational causal properties, which are constituted also by their environment and interaction. Near-decomposable systems are thus hierarchical in the sense that the complex whole can be seen as made from a limited set of simpler parts and interactions. Hierarchical systems are more manageable for cognitively limited beings because their ‘complete description’ includes recurring or irrelevant elements describing similar recurring parts and non-important interactions. The removal of such descriptions does not hamper our understanding of the system and thus eases cognitive load.

Although there are a number of arguments that conclusively show that such informational economy by itself is not *constitutive* of understanding,[2] we agree with Simon (and Bechtel and Richardson) in that a property closely related to near-decomposability, namely *modularity,* is a necessary condition for mechanistic explanations. In the case of causal-mechanistic explanations, the explanatory dependencies track the consequences of *interventions* (Woodward 2003; see fig. 1) and causal knowledge thus enables the goal-directed manipulation of the object of explanation. These answers are the basis of the inferential performance constitutive of causal understanding.

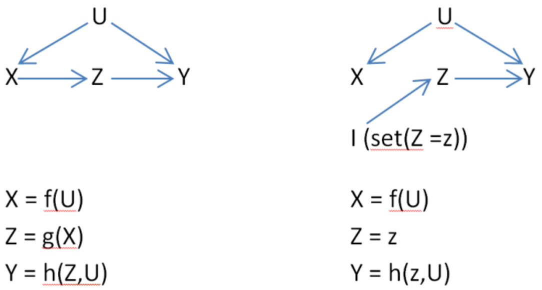


Figure 1. Invariance under exogenous interventions distinguishes “deep”, causal, dependencies from mere correlations. P(Y|Z = z) is not the same as P(Y|set(Z = z)).

Such answers to what-if questions are derived from internal or external representations of the object of understanding. In order for these answers to be well defined, the dependencies grounding the answers have to possess some degree of independence such that *a local change in an aspect of the phenomenon under study cannot ramify uncontrollably or intractably*. If local modifications in a part of a system disrupt other parts (dependencies) in a way that is not explicitly specified (endogenized) in the (internal or external) representation of the system according to which the what-if inferences are made, the consequences of these changes are impossible to predict and counterfactual assertions impossible to evaluate (Woodward 2003, 333).

[2] See, e.g.,Woodward 2003, 362–364.

Therefore, a necessary condition for a representation to provide explanations, and thus understanding, of a phenomenon is that the modularity in the representation matches the modularity in the phenomenon.

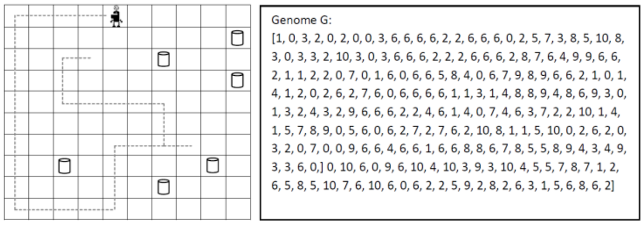
If we intervene on a causal input corresponding to variable X_i in a model of the studied system, and the intervention, no matter how surgical, also changes the dependencies within the system, or values of other variables themselves affecting variables causally downstream of X_i, the model does not give correct predictions about the consequences of the intervention. Hence, the model does not provide correct causal understanding of the system and the causal role of the variable in it. If the system cannot be correctly modeled on any level of description or decomposition so that it is modular in the way described above – if the system itself is not causally modular – no what-if-things-had-been-different questions concerning interventions in the system can be answered. This would mean that every local change brings about intractable changes elsewhere in the system to such an extent that there can be no representation that would enable a cognitively finite being to track these changes and make correct inferences about their consequences.

The problem of understanding causally non-modular systems has received some attention in the philosophy of science literature (e.g., Bechtel and Richardson 2010, Ch. 9). However, according to the DL schema, before we can even start thinking about searching for the causal-mechanistic implementation of the complex system behavior, we need to formulate hypotheses about the possible functional decompositions of the behavior (see also Cummins 1983). For example, what kind of simpler subtasks could possibly produce complex cognitive capacities such as language production and comprehension, long-term memory, and visual object-recognition? Importantly, this task is separate, though not independent, from hypotheses concerning the implementation of the capacity. Although the understanding offered by the functional decomposition is not, strictly speaking, causal – component operations do not *cause* the whole behavior because they are constitutive parts of it – the modularity constraint on understandability still applies in the following way. We can only understand the complex behavior by having knowledge of its component operations, if we can make reliable what-if inferences concerning the possible consequences of changes in the component operations for the properties of the more complex *explanandum* capacity. For example, we provisionally understand working memory if we can infer from possible changes in its hypothesized component operations (such as differences in the properties of the postulated phonological loop or episodic buffer) to changes in the properties of the capacity. These inferences are only possible if the functional decomposition itself is suitably modular, i.e., the consequences of "local" changes in component operations do not ramify in an intractable way, making the behavior of the whole completely holistic. We now argue that genetic algorithms demonstrate that design-by-selection can lead to such non-modular complex behavior.

Genetic Algorithms

From the point of view of AI, genetic algorithms (henceforth GAs) are a form of non-exhaustive but massively parallel search in the search space of a problem (Holland 1975; Mitchell 1996). Although GAs are not the only strand of evolutionary programming, they serve our purpose well because their basic principles are easy to understand and they are the most well-known kind of evolutionary programming outside computer science (Clark 1997, 2001; Mitchell 2009). GAs are useful for a number of different purposes, but here we use a simple example originally from Mitchell (2009, Ch. 9), where a GA is used to evolve a behavioral strategy for a simulated agent.

Mitchell's model shows how an algorithm mimicking biological evolution can be used to develop a controlling program for a robot picking up soda cans on a 10x10 grid. Robby the robot can only see the squares adjacent to its location (center, North, South, East, West), and each turn it can either move one step to a particular direction, move at random, try to pick up a can, or do nothing. Each simulation run lasts for a predetermined amount of time steps (originally 200), and Robby's task is to pick up as many randomly situated cans as possible.



Genome G:
[1, 0, 3, 2, 0, 2, 0, 0, 3, 6, 6, 6, 6, 2, 2, 6, 6, 6, 0, 2, 5, 7, 3, 8, 5, 10, 8,
3, 0, 3, 3, 2, 10, 3, 0, 3, 6, 6, 6, 2, 2, 2, 6, 6, 6, 2, 8, 7, 6, 4, 9, 9, 6, 6,
2, 1, 1, 2, 2, 0, 7, 0, 1, 6, 0, 6, 6, 5, 8, 4, 0, 6, 7, 9, 8, 9, 6, 6, 2, 1, 0, 1,
4, 1, 2, 0, 2, 6, 2, 7, 6, 0, 6, 6, 6, 6, 1, 1, 3, 1, 4, 8, 8, 9, 4, 8, 6, 9, 3, 0,
1, 3, 2, 4, 3, 2, 9, 6, 6, 6, 2, 2, 4, 6, 1, 4, 0, 7, 4, 6, 3, 7, 2, 2, 10, 1, 4,
1, 5, 7, 8, 9, 0, 5, 6, 0, 6, 2, 7, 2, 7, 6, 2, 10, 8, 1, 1, 5, 10, 0, 2, 6, 2, 0,
3, 2, 0, 7, 0, 0, 9, 6, 6, 4, 6, 6, 1, 6, 6, 8, 8, 6, 7, 8, 5, 5, 8, 9, 4, 3, 4, 9,
3, 3, 6, 0,] 0, 10, 6, 0, 9, 6, 10, 4, 10, 3, 9, 3, 10, 4, 5, 5, 7, 8, 7, 1, 2,
6, 5, 8, 5, 10, 7, 6, 10, 6, 0, 6, 2, 2, 5, 9, 2, 8, 2, 6, 3, 1, 5, 6, 8, 6, 2]

Figure 2. Each "locus" in the genome G corresponds to one of the possible immediate environmental states of Robby, and each digit (the allele) to a move in that situation (e.g., '0' → 'move north', '5' → 'pick up') (see Mitchell 2009, 137).

Initially a random population of software individuals is generated, each with a "genome" consisting of 243 random numbers. Each locus in the genome guides Robby's behavior in a particular situation (Fig 2). The fitness score of each candidate in the population is calculated by running several simulation trials: crudely, the more cans the robot is able to pick up on average, the higher its fitness. Programs with the highest fitness are then used to form the next generation: they are paired randomly, and the genomes of the two parents are crossed over at a randomly chosen point to create the genomes of new individuals. Finally, for each locus of a descendant's genome, there is a small probability (.005) that a mutation occurs in it. As a result, the new generation is based on the most successful variants among the previous generation, and the process loops back to the fitness-calculation phase. Thus the GA continues searching for efficient solutions by charting new regions of the search space.

After a few hundred generations, the evolved strategies start to achieve impressive results. As we replicated Mitchell's simulation, we observed that after the 800th generation, the best strategies among evolved Robbys started to have higher fitness scores than a Robby programmed by a human designer (ultimately 480 vs. 440 points).[3] However, although solutions found with GAs are efficient, their behavior is often hard to understand. The ingenious behavioral strategies that the programs employ cannot be deciphered by simply looking at individual genes or sets of genes. Instead, it is necessary to look holistically at the broad phenotypic behavior of the robot. A nice illustration of this impenetrability of such evolved solutions is the fact that in some cases when a high-fitness Robby is in the same square with a can, it decides not to pick it up, but rather moves away from the square. While this behavior seems *prima facie* irrational, looking at the total behavioral profile of the robot uncovers a clever strategy: Robby uses cans as markers to remember that there are other cans on its side, and it explores the adjacent squares for extra cans before picking up the marker can. Thus by not treating cans only as targets but also as navigational tools, Robby uses its environment to extend its severely limited visual capacities and to compensate for its total lack of memory.

Moreover, by examining the behavior of a highly efficient 1500th generation Robby, it can be seen that this marker strategy manifests in slightly different ways in different environmental situations. It is not a discrete adaptation, but rather a collection of independently evolved sub-strategies. Furthermore, the marker strategy is tightly intertwined with another environment-employing "hack" that the sophisticated Robby uses: when there is already a lot of empty space on the grid, Robby employs a "vacuum-cleaner" movement strategy. It follows the walls of the board, departing toward the center when it detects a can, employs the marker strategy if possible, and immediately after cleaning up its local environment, returns directly to the south wall to continue its round around the board. This strategy also includes an ingenious "bounce" feature: when Robby arrives to the corner preceding the wall that is parallel to its default navigation direction in an empty field, it bounces off the wall to increase the range of this search pattern.

Such "kluges" are common to designs created by GAs. Like biological evolution, GAs can come up with solutions that a human designer would not think of. These solutions often offload parts of problem solving to the environment, and thus rely on a tight coupling between the system and its environment. And, as pointed out by Clark (1997, 2001), recurrent circuitry and complex feedback loops between different levels of processing often feature in systems designed by GAs. Such designs are often difficult to understand.

We suggest that these difficulties in understanding are often created by the lack of modularity in the functional decomposition of the behavior. The high-fitness Robby (genome G in Fig. 1) mentioned in the paragraph above only leaves cans as markers in some specific situations, and only the totality of this selective marking strategy – together with navigational strategies utilizing cans and walls – constitutes the effectiveness of the can-search procedure. Looking at isolated genes in Robby's genome only reveals trivially modular elements corresponding to elementary subtasks in its behavior: one gene corresponds to an elementary move in a specific environmental situation. But we cannot make inferences from local hypothetical changes in these elemental behaviors to consequent effects on fitness. The connection between any single elementary behavioral rule and the strategy is simply too complex and context dependent. A change in a single rule (in situation B; a can present; whether to pick up or not to pick up the can) has consequences for the effectiveness of the other elementary behavioral rules. Explanatorily relevant inferences would require an extra "level" of modular sub-operations between the individual movements and the strategy as a whole.

The marker and vacuum-cleaner strategies mentioned above appear to be examples of such middle-level sub-operations, but by themselves they are insufficient to yield understanding of the whole behavior of our most successful Robby. This is because the effectiveness of leaving a can is a result of the evolved coupling between the specific situations in which Robby leaves a can and the rest of the navigation behavior. Therefore, there is no way of independently altering these middle-level strategies. Also "the bounce" is intertwined with the rest of the vacuuming navigation and cannot be independently altered. In general, genetic algorithms do not often produce easily discernible designs. Rather, the interesting heuristics in the system's behavior can only be revealed by simultaneously looking at constellations of different genes, and eventually, the whole genome.

To recapitulate, our example exhibits several distinct (yet related) challenges to understanding:

(1) The discernible middle-level strategies (marker, vacuum-cleaner) do not have dedicated structural bases. Instead, the nature of the design process leaves all atomic structural elements (the 243 DNA elements) open for exploitation by all capacities serving the main goal. Consequentially, the system is neither structurally nor behaviorally nearly-decomposable, but instead has a "flat hierarchy," and strategies are implemented in highly distributed structures.

(2) Challenge 1 above means that the interactions between subtasks tend to be strong: a change in one subtask constituting a part of the marker-behavior also affects the functioning of the vacuum-cleaner navigation. In general, the middle-level strategies can only be discerned and defined in a very abstract way, and the interaction-effect on their contribution to the overall fitness is so large as to make

[3] Code obtainable on request.

any inferences about the consequences of partial changes in one strategy next to impossible.

(3) The way in which the strategies contribute to the fitness of the individual is highly context-dependent and depends on the properties of the environment as well as the DNA of the agent. Even small modifications to the environment can lead to drastic changes in the performance of a strategy. For instance, we observed that adding only a few randomly placed extra walls on the grid radically collapses the average score of the successful Robby described above.

Extrapolating from this very simple case, we contend that GAs may design behavior that cannot be tidily decomposed into hierarchical and modular subtasks, whose individual contributions would be easy to understand (i.e., we could infer how a change in a sub-routine would affect the behavior of the mother-task). Instead, feedback, many tasks using the same subtasks as resources, and tight system-environment coupling lead to holistic design where almost "everything is relevant for everything." The evolved functional architecture is flat in that there are few discernible levels of order between the elementary operations and the complex behavior. The counter-intuitiveness of such flat architectures is apparent in the deep mistrust faced by connectionist suggestions for non-hierarchical design of cognitive capacities (see e.g., Rumelhart and McClelland 1986 vs. Pinker and Prince 1988).

Furthermore, GAs underscore the path dependence of evolutionary problem solving. For sufficiently complex computational problems there are often several local maxima in the fitness landscape of the problem, and the population can converge to different maxima in different runs of the simulation. The functional decomposition that a human designer comes up with is just one possible solution among several others. Perhaps our biological evolution actually ended up with a radically different one.

Lessons for the Study of Mind

Genetic algorithms demonstrate that evolution can, in principle, lead to non-modular functionality. This imposes a limit on our ability to understand such behavior: if we cannot trace the consequences of changes in the sub-operations, we cannot answer *what-if* questions concerning the complex behavior. Such behavior constitutes a thorny problem for mechanistic understanding of the implementation of the said behavioral capacities, since the DL heuristic cannot get off the ground: we do not even know what we are supposed to localize. We can now ask two questions: should we expect to find such non-modular functionality in nature, especially in human cognition, and if so, what attitude should we adopt with respect to this problem. Should the aim of causal-mechanistic understanding of the brain be given up, and be replaced, for example, with non-mechanistic dynamical models often employing a limited set of instrumentally interpreted macro-variables?

There are important disanalogies between GAs and biological evolution. As is the case with Robby, there is often no genotype–phenotype distinction. In biological evolution, however, genes do not directly cause properties of the phenotype, but rather participate in guiding ontogenesis. It has been suggested that ontogenesis itself favors modular design. GAs may also seem a problematic platform for exploring the possibilities of DL heuristics, since the lowest level of functional organization and the level of implementation are identical (i.e., the genome). However, we see no reasons why this would affect our argument. Moreover, the argument developed here is not only about genetic selection, but about selection in general, and failures of functional modularity may in principle also arise in the course of development – at least if the idea of neuronal group selection or "neural Darwinism" is taken seriously.

The recent research on biological control networks (metabolic and gene regulatory networks) suggests that evolved modular organization is in fact the rule rather than the exception: control networks exhibit network modularity and the recurring modules (motifs) have easily discernible modular functions. Therefore, the question in the recent years has rather been to formulate an evolutionary explanation for this modular design. Genetic algorithms have been used to argue that modularity is not selected for, but that it is instead a byproduct of specialization of gene activity (Espinasa-Soto & Wagner 2010) or of selection against densely connected networks and long connections (Clune, Mouret & Lipson 2013).

Most interesting for our case, Kashtan and Alon (2005; see also Kashtan et al. 2007) have demonstrated that when the goals themselves are composed of modularly varying sub-goals, evolution tends to produce modular functionality. It seems easy to see why this is the case. If the tasks to which the system has to adapt remain the same, the selection environment does not change, and the peaks in the fitness landscape are stable, then selection favors strategies that offload problem solving to that particular environment as much as possible. But if the task itself is composed of changing subtasks, it makes sense to design the adaptive response in such a way that a particular sub-operation can locally adapt to a local change in a subtask without altering the totality of the otherwise well-functioning behavior.

In their research, Kashtan and Alon evolved several network models to compute complex Boolean functions, with fitness calculated according to how close the network output was to the target. They found that by modularly varying goals, it is often possible to considerably speed up the evolution. In our Robby simulation, we studied the effects of changing environment for the evolution of modularity by allowing the environment to change discretely from an initial no-walls (torus) condition to one with walls, and eventually to one with also random obstacles. Our results suggest that although "modularity in tasks" does speed up learning, it can often prematurely weed out diversity in the population in such a way that, in the end, the global maximum for the main target task cannot be reached.

It seems likely that our cognition has evolved in at least partly modularly changing selection environment, but the extent to which we should expect to find modular functionality in human cognition is hard to estimate. We suspect that the usefulness of many of the existing computational models investigating the evolution of biological modularity is constrained by the fact that the tasks (e.g. simple categorization, logic circuits) solved by the algorithms are straightforwardly computational and do not really involve any interesting behavioral aspects. This is why the Robby platform has certain advantages for exploring evolved functionality: The dynamic nature of the simulation allows the "emergence" of novel and irreducibly top-level strategies in a way that is lacking in the more static contexts.

Because of the uncertainty related to the actual extent of non-modularity in human cognition, we stress the conditional nature of our argument. Our study of genetic algorithms and our analysis of the properties of the resulting designs only demonstrates that evolution *can* create designs, which are in principle beyond the understanding of unaided cognitive beings such as us.

Yet there is nothing mysterious in such designs. Simon pondered whether the apparent abundance of hierarchical, nearly decomposable complexity was due to our selective attention to precisely such systems, but we believe this to be a somewhat hasty conjecture. We have no trouble finding and delineating systems, such as Robby, or possibly ourselves, that manifest functionally non-decomposable behaviors sustained by a flat architecture. However, there certainly might be a psychological bias that makes us see hierarchical design also where there is none. One way of coping with this obstacle to understanding is to realize that there are no fundamental reasons to limit the relevant *epistemic* agent to be an *unaided* human. Although only a human agent can experience a sense of understanding, this feeling should not be confused with understanding itself. Therefore brute computational approaches can produce understanding as long as the epistemic subject, the cognitive unit whose inferential abilities are to be evaluated, is conceived as the human-computer pair.

References

Bechtel, W., & Richardson, R. (2010). *Discovering Complexity: Decomposition and Localization as Strategies in Scientific Research*. Cambridge MA: The MIT Press.

Churchland, P. M. (1989). *A Neurocomputational Perspective: The Nature of Mind and the Structure of Science.* Cambridge, MA: MIT Press.

Clune, J., Mouret, J-B., & Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society B,* 280**.**

Clark, A. (1997). *Being There: Putting Brain, Body and World Together Again*. Cambridge MA: The MIT Press.

Clark, A. (2001). *Mindware: An Introduction to the Philosophy of Cognitive Science*. Oxford: Oxford UP.

Cummins, R. (1983). *The Nature of Psychological Explanation*. Cambridge MA: MIT Press.

Espinosa-Soto C, Wagner A. (2010). Specialization Can Drive the Evolution of Modularity. *PLoS Computational Biology,* 6(3).

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Jacob, F. (1977). Evolution and Tinkering. *Science* 196 (4295): 1161–1166.

Kashtan, N., & Alon, U. (2005). Spontaneous evolution of modularity and network motifs. *PNAS* 102 (39): 13773–13778.

Kashtan, N., Noor, E. & Alon, U. (2007). Varying environments can speed up evolution. *PNAS* 104 (34): 13711–13716.

Levins, R. (1973). The Limits of Complexity. in Pattee, H. (Ed.), *Hierarchy Theory: The Challenge of Complex Systems*. London: Braziller.

Marcus, G. (2008). *Kluge: The Haphazard Construction of the Human Mind*. Boston and New York: Houghton Mifflin.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge MA: MIT Press.

Mitchell, M. (2009). *Complexity. A guided tour.* Oxford: Oxford University Press.

Piccinini, G., & Craver, C. (2011). Integrating Psychology and Neuroscience: Functional Analyses as Mechanism Sketches. *Synthese* 183 (3): 283–311.

Pinker, S., & Prince, A. (1988). On Language and Connectionism: Analysis of a Parallel Distributed Processing Model of Language Acquisition. *Cognition* 23: 73–193.

Rumelhart, D., & McClelland, J. (1986). On Learning the Past Tenses of English Verbs. In McClelland & Rumelhart *et al*. (Eds.), *Parallel Distributed Processing vol. I.* Cambridge MA: MIT Press.

Simon, H. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society* 106: 476–482.

Thagard, P. (2012). *The Cognitive Science of Science*. Cambridge MA: MIT Press.

Waskan, J. (2006). *Models and cognition: Prediction and explanation in everyday life and in science.* Cambridge, MA: The MIT Press.

Wimsatt, W. (2007). *Re-Engineering Philosophy for Limited Beings*. Cambridge MA: Harvard UP.

Woodward, J. 2003. *Making Things Happen*. Oxford UP.

Ylikoski, P., & Kuorikoski, J. (2010). Dissecting Explanatory Power. *Philosophical Studies* 148, 201–219.