Three

COMPUTERS AS MODELS OF THE MIND: ON SIMULATIONS, BRAINS, AND THE DESIGN OF COMPUTERS

Peter Asaro

After the war, together with a small group of selected engineers and mathematicians, Johnny built, at the Institute for Advanced Study, an experimental electronic calculator, popularly known as Joniac*, which eventually became the pilot model for similar machines all over the country. Some of the basic principles developed in the* Joniac *are used even today in the fastest and most modern calculators. To design the machine, Johnny and his co-workers tried to imitate some of the known operations of the live brain. This is the aspect which led him to study neurology, to seek out men in the fields of neurology and psychiatry, to attend meetings on these subjects, and, eventually, to give lectures to groups on the possibilities of copying an extremely simplified model of the living brain for man-made machines.*

Klara von Neumann, wife of John von Neumann (John von Neumann, 1958, p. viii)

Turing knew perfectly well what the job he had to do, which was to manufacture or design a machine that would do the complicated sort of mathematics that had to be done in the Mathematical Division of NPL*. But he had all sorts of interesting things that he liked to do: for example, he was really quite obsessed with knowing how the human brain worked and the possible correspondence with what he was doing on computers …. Turing thought that the machine should be made quite simple, and at the same time should make everything possible that could be done. His particular purpose was to permit the writing of programs that modify programs, not in the simple way now common but rather in the way that people think.*

Ted Newman, colleague of Alan Turing (1994, p. 12)

1. Introduction

The purpose of this chapter is to clarify some of the important senses in which we might consider the relationship between the brain and the computer as one of "modeling." I will also consider the meaning of "simulation" in the relationships between models, computers, and brains. While a fairly broad literature on models and simulations in science has emerged, it has primarily focused on the physical sciences, not the mind and brain. And while the cognitive sciences have often invoked concepts of modeling and simulation, they have been frustratingly inconsistent in their use of these terms, and the implicit relations to their scientific roles. My approach is to consider the early convolution of brain models and computational models in cybernetics, with the aim of clarifying their significance for more current debates in the cognitive sciences. I believe that clarifying the historical senses in which the brain and the computer worked as models of each other in the historical period prior to the birth of AI and cognitive science is a crucial task for an archeology of AI and the history of cognitive science. "Model" is a challenging concept, in part because it is a noun and a verb, an object and a practice, and because it takes many prepositional phrases. Let *X* be the model, *Y* the thing modeled, and *Z* the agent doing the modeling. We may distinguish the following different meanings of "model": *X* models *Y*, *X* is a model of *Y*, *X* is a model for *Y*, *Y* is modeled on *X*, *Y* is modeled after *X,* and even *Y* models for *Z* (it would be more proper to include "for *Z*" in each of the prepositional phrases, but I have omitted it when the context is unambiguous). We could say that the brain was a *model for the structure* of computer (or the computer was *modeled on* the brain) in the sense that the designers of the early computers, such as John von Neumann, treated the biological brain like an artist's model, and crafted the computer in its image. Once completed, we might think of the artist's sculpture as a sort of model of its subject—and so we might think of the computer as a *model of* the brain.

This would seem to be the case for some electronic devices that were *models of the performance and behavior* of the brain, such as W. Ross Ashby's Homeostats, which intended to be models of the adaptive properties of the brain, or W. Grey Walter's Tortoises, which intended to be models of the dynamic drives of the living brain. In yet another sense, researchers such as Alan Turing, saw the first computers as *models of the fundamental structure* of the brain in the sense that the digital computer was an engineered device that worked on the same principles as the brain and as such it could be used to test theories about how the higher level functions of the brain might operate. For each of these cases, the modeling—the construction of each device—was unique and complex and involved further concrete and abstract models of the brain and its behavior. I have considered the nature of working models that the Homeostats and Tortoises represent elsewhere (see Asaro, 2006). In this chapter I will add to theoretical and working models another class—the "simulation"—which is a term we use in nearly as many ways as "model." Many authors understand "simulation" as a synonymous of "model," while others see simulations as "working models," because their behavior can become the subject of empirical inquiry. More precisely, "simulation" generally refers to a special class of models whose use defines them—specifically computational models that we use to approximate the behavior of a system of equations that are too difficult to solve by analytic techniques.[1] Because of the power of the computer, simulations have come to be one of the most significant kinds of models in scientific practice. Yet they occupy a strange place *vis à vis* working models like the Homeostat and the Tortoise. In fact, we might see computational simulations as challenging the distinction between theoretical and working models. On the one hand, simulations appear to be something like automated theories. On the other hand, simulations are the most abundant form of models that *do* something—working models. This duality of character leads to the power of simulations and to the difficulty of properly understanding their role in science.

I begin with a review of some crucial aspects of the work of John von Neumann and Alan Turing, who were involved in the theorization of mathematical computation and the design of the first general-purpose stored-program computers. They also theorized the ways in which the computer could simulate the brain and mind. This examination reveals that the design of the first general-purpose computers drew heavily upon the McCulloch and Pitts (1943) neuron model and other aspects of neurophysiology. The brain, or more precisely the theoretical models of neurons, *were* the models of the first computers. We also have a sense of how the human practices of performing mathematical computations shaped the mathematical theory of computation. We do not often discuss these basic forms of modeling, yet I believe they have had a significant impact on our theoretical intuitions regarding the more complex forms of modeling of the computer after the brain. This was especially true early on, when we used to refer to computers as "giant brains" (Berkeley, 1949).

The computer is a unique artifact among the tools of modeling for several reasons that deserve special attention. The very notion of a symbolic simulation is tied up with ideas about the nature of computation and the technological performance of working computers. These ideas, and the conceptualization and design of the first computers, are intimately related to models of the brain and mind in multiple and complex ways. Part of my purpose in this chapter is to make these multiple and complex relations visible as instances of the different types of scientific modeling and models. In considering the computer and the brain, we can find instances of each type of model we have just considered—theoretical and working models—as well as the simulations that I will discuss shortly.

Before embarking on our historical journey to the electronic brains of the 1940s, I would like to consider the view of the mind that has dominated cognitive psychology since then. Scholars have dubbed "computationalism" the philosophical view that the mind *is* a computer. The preface to a recent book on Computationalism summarizes the view:

> Are minds computers? … Computationalism—the view that mental states are computational states—is based on the conviction that there are program

> descriptions of mental processes and that, at least in principle, it is possible for computers, that is, machines of a particular kind, to possess mentality. (Scheutz, 2002, p. ix)

Despite many different formulations, Computationalism is rooted in the basic analogy between mental states and computational states—the different formulations are the result of different definitions and conceptions of these two types of states. Most often, we define computational states in terms of the stored-program digital serial computer, or the mathematical theory of computation. Two of the mathematicians most responsible for the development of the mathematical theory of computation, and who had the greatest influence on the design of the first such computers—Alan Turing and John von Neumann—were also building synthetic minds. As I will show in this chapter, while they shared a common understanding of computation, they differed in their views of how a computer might simulate a brain or the mind. Still different from theirs was W. Ross Ashby's attempt to model mental phenomena. Ashby saw "information processing" as central to the adaptive mind, but not strictly as a form of computation. Hence, he built analog simulations (working models) of the brain, whereas Turing and von Neumann were seeking different kinds of symbolic simulations (theoretical models).

First, a clarification of the distinction between simulations as working models and simulations as automated theoretical models is in order. An examination of the early history of the computer and of the attempts to simulate the brain with these early machines will demonstrate the usefulness of distinguishing the different types of models and modeling. In so doing, I will consider the automata theory that von Neumann developed as he was designing the EDVAC, arguably the first stored-program computer. I will also consider his use of the McCulloch and Pitts neuron model in this design, and his later thoughts on simulating the brain on the computer. I will then consider Turing's preoccupation with the universal (Turing) machine as a model of the mind and brain, which will lead me to Turing's suggestion to Ashby to simulate the Homeostat on his ACE computer, and to a direct comparison of these two types of simulations. I will then conclude with the implications of the view of models and simulations for Computationalism that we have developed over time.

2. Analog and Symbolic Simulations

Making a careful distinction between theoretical models and working models has an advantage. While the notion of theoretical models has its roots in the normative philosophy of science, theoretical and working models are useful to a descriptive naturalistic study of scientific practice. We have to make the crucial distinction between, on the one hand, working models—which do something in the world and have a material agency that is independent of humans—and, on the other hand, theoretical models, which—while sometimes endowed with agency of a normative and disciplinary sort (Pickering, 1995, p. 120)—are not completely independent of humans, but require an agent willing to conform to their normative rules and disciplined constraints. Let us consider more carefully the distinction between theoretical and working models, and how they relate to the current philosophical discussion of simulations.

In his recent work on computational simulations, Winsberg (2001; 2003) considers three traditional attempts to account for computational simulations in physics: as metaphors, as experiments, and a third middle mode. The view of simulations as metaphors, while perhaps never expressed as such, nonetheless holds that simulations are just brute-force number crunching procedures, that we use when analytic techniques are impossible—in other words, a degenerate form of theorizing. The view that simulations are experiments, and the computer is an experimental target, holds that there is some mimetic relation between the simulation and the simulated, such that the simulation can mimic the real and act as a stand-in. The third mode holds that simulations are an entirely different kind of thing, lying somewhere between theorizing and experiment.

Winsberg is careful to distinguish between simulations in which analytic solutions produce closed form expressions, and simulations that use numerical methods producing a "big pile of numbers" that require the usual tools of experimental practice to analyze: visualization, statistics, data mining, and so forth (2003, p. 111). The difference is not merely one of the mimetic qualities of the mathematics, but of the practices that scientists use to engage and work with the models. The numerical methods are more like experiments than theories, because the same practices used to study them are used in experimental investigations.

As I have mentioned above when discussing working models, we employ these synthetic numerical simulations to generate, or synthesize, phenomena that we want to investigate, explore, and experiment on. Unlike the theoretical models or the simulations based on tractable equations that we deduce from theory—which lend themselves to straightforward mathematical analysis—these synthetic numerical simulations are themselves objects of empirical study. In these analytic simulations we do not need to employ the data analysis techniques of experimental practice to discern the structures and patterns in the phenomena—these models only produce data as an instantiation of the theory for a given case, and we can easily derive the desired results from the equations. This distinction also corresponds nicely to an analytic/synthetic distinction. Here the analytic simulations engage in deriving local models from general theories in a formal way, whereas the synthetic simulations seek to fill in the gaps of missing theory and data by generating something new that we can manipulate, experiment on, and use to generate data on which to devise and test theories.

Much of the philosophical literature on simulations appears to Winsberg to be hung up on visualization as a key aspect of what makes simulations interesting. The notion of the mimetic qualities in the simulation's representations (Hughes,

1997) captures this. Like isomorphism, mimetic relations aim to provide an objective way of expressing the precise relations between the real and the simulated systems, but mimetic relations have the added requirement of preserving the graphical aspects of the original:

> The extensive use of realistic images in simulation is a stepping stone that simulationists use in order to make inferences from their data. It is also a tool they use in order to draw comparisons between simulation results and real systems; a move that is part of the process of sanctioning of their results. It is the drawing of inferences and sanctioning of results that give rise to the interesting philosophical connections between simulation and experimental practice. (Winsberg, 2003, p. 113)

The graphical aspects of such models can provide a means to the employment of visual laboratory practices, but it is not the only such aspect, and it is not essential. As Winsberg notes, we can derive visualizations from many mathematical models, and this appears tangential to what makes them good models. Further, if we put too much faith in the power of the mimetic features, we see simulations as purely and truly experimental, taking literally the notion of the "numerical experiment" and interpreting the computer simulation as a stand-in for the real phenomena. This begs the question of just how well a simulation mimics the real phenomena (2003, p. 115). It matters not just that it "stands in" in some respects, but in which respects, to what extent, and to what precision and accuracy. If we are interested in the epistemic status of models and simulations, these questions are of great importance, and we cannot take for granted that because we can perform experiments on simulations, they must have the same epistemic, much less metaphysical, status as real experiments. Still, they must not completely lack epistemic status either, or be metaphysically weak or unreal.

Viewing simulation as a unique new mode of scientific practice between theorizing and experimenting, Winsberg asserts that this is merely a good place to start thinking about simulation, not an explanation of it:

> What is of interest philosophically is to understand (a) how it is that what is at root a theoretical enterprise, takes on characteristics of experimentation, (b) what those characteristics are—at the abstract, reconstructed level, (c) what consequences there are of such a hybrid for our understanding of the nature of modeling, theorizing, and experimenting, and (d) how simulation produces knowledge and what kind of knowledge that is. (2003, p. 118)

In staking out this middle ground for simulation, Winsberg is careful to note that the techniques and practices of modeling can carry their epistemic credentials, independently of the theory that produces them:

> [T]he credibility of [a] model comes not only from the credentials supplied to it by the governing theory, but also from the antecedently established credentials of the model building techniques developed over an extended

> tradition of employment. That is what I mean when I say that these techniques have their own life; they carry with them their own history of prior successes and accomplishments, and, when properly used, they can bring to the table independent warrant for belief in the models that they are used to build. (2003, p. 122)

It appears that for Winsberg the epistemic basis for simulations comes from theory and from the practices of modeling. More importantly, he also holds that those simulations are autonomous (or semi-autonomous, as he says) from theory only to the extent that a tradition of practice roots these independent epistemic foundations. I agree with this approach, and only wish to add that working models are also produced in and support the development of a tradition of modeling practices.

Before discussing the early attempts at using the computer to simulate the mind and brain, I want to consider one more position on simulation that considers more explicitly how a model represents the system it models. During the Connectionist debates of the early 1990s, a recurring argumentative theme occurred, which the distinction between analog and digital computation motivated. Perhaps we made too much of this distinction, or rather, we failed to fully recognize its real nature. The work by Russell Trenholme (1994) is an exception. Developing a line of thought connecting the ideas of Kenneth J. W. Craik, Norbert Wiener, Philip Johnson-Laird, and Rodney Brooks, Trenholme presents a view of simulation that I believe is compatible with the idea of a working model. I hope that applying his view to the work of von Neumann and Turing on the early computers will provide a new perspective on how we variously conceived the computer as a simulation of the brain and mind.

In short, while all synthetic simulations are working models in the generic sense of being automatic, just as all models are "representational" in a generic sense, the nature of the scientifically relevant aspect of their being models can differ in significant ways. An analog simulation directly models a natural system, while a symbolic simulation employs an intermediate symbolic system and is thus an indirect model. This level of indirection is highly significant to the extent that it shifts the relevant aspect of a simulation from the realm of working models, epistemic artifacts, and material agency, to the realm of theoretical models and normative disciplinary agency. Trenholme calls a naturalistic analog simulation what I have been calling a working model, and he calls a symbolic simulation a special kind of theoretical model, one whose disciplinary agency has been transferred from human hands and minds to automatic computations.

Trenholme's argument begins with a clarification. While much of the debate at the time was couched in terms of "analog *vs*. digital" computation, Trenholme is careful to point out that the real issue is "analog *vs.* symbolic" representation. As I will show in the discussion of von Neumann's automata theory below, "analog *vs.*

digital" is a matter of how computers represent *numbers*. It says nothing, however, about what or how those numbers come to represent or simulate anything else beyond themselves. The real issue is whether numbers are an essential part of the simulation at all. The "analog *vs.* symbolic" dichotomy intends to capture the notion that analog simulations do not require or depend upon symbolic representations. Instead, they serve as simulations, primarily or completely, in virtue of their causal structure. Trenholme calls these "naturalistic analog simulations" in order to distinguish them from the more careless definitions of analog computation.

Trenholme goes on to argue that naturalistic analog simulations are not representational in the same sense that symbolic simulations are representational.[2] The involvement of an additional "mapping," in the sense of multiple levels of representation, motivates the difference. In short, analog simulations relate the causal structure of a natural phenomenon to the causal structure of the simulation by isomorphism or by the looser "similarity" relation (along with a probabilistic causal theory). We derive, or map, a symbolic simulation, on the other hand, from a formal theory of a natural phenomena, and we then map the result into a computational simulation of the formal theory. Missing from the symbolic simulation is the obvious sense of an isomorphism between causal structures. Symbolic simulations are instead representational in the sense of semantic relations like denotation. Analog simulations lack this purely representational layer.

Symbolic simulation is a two-stage affair: first comes the mapping of inference structure of the theory onto hardware states, which defines symbolic computation; second comes the mapping of inference structure of the theory onto hardware states, which (under appropriate conditions) qualifies the processing as a symbolic simulation. In contrast, a single mapping from causal relations among elements of the simulation to causal relations among elements of the simulated phenomenon defines analog simulation (Trenholme, 1994, p. 119).

While a symbolic simulation depends on causal structures at *some* level, the symbolic level is crucial to its successful performance as a simulation. Similarly, while an analog simulation depends on representation at *some* level, it is the causal structure, or material agency, that is operative in its performing as a simulation. Both types of simulation entail a generic sense of representation: the simulation is in some sense a representation of the system that is simulated, but the kind of representation at stake is not necessarily symbolic representation.

For instance, the mercury barometer "represents" air pressure without employing symbolic systems *in its performance*—the symbols only come at the end, when we read the result off from the markings on the device. Instead of depending upon the symbolic relations to achieve a well-behaved simulation, analog simulations depend upon causal structures. These causal structures are the material agency of working models. On the one hand we need to constrain these causal structures in specific ways so as to constitute a valid simulation (they are disciplined material agencies), while on the other they, to the extent that causal structures engage the potentially infinite causal relations of the world and can always exhibit new emergent properties, are always unconstrained and open-ended. In analog simulations, an implicit respect is in order for this dual aspect of material agency, which is largely missing in symbolic representations that attempt to completely constrain the behavior of the physical system to produce only the precisely intended symbolic system. When a symbolic simulation behaves irregularly, its output is meaningless.

Symbolic representation is a quite specific form of representation in which symbols represent via denotation and reference. Symbols alone do not have a causal structure, and in order to base an automatic simulation on them, we ought to have a computer, which Newell and Simon call a "physical symbol-system":

> *The Physical Symbol System Hypothesis.* A physical symbol system has the necessary and sufficient means for general intelligent action. By "necessary" we mean that any system that exhibits general intelligence will prove upon analysis to be a physical symbol system. By "sufficient" we mean that any physical symbol system of sufficient size can be organized further to exhibit general intelligence (Newell and Simon, 1981[1976], p. 41)

Simulations based upon symbolic representations are so closely tied to the development of the modern computer that we often just call them "computer simulations." Other expressions evoking notions of modeling and simulation are in use, but they confuse the issue:

> Certain ways of speaking—for example, saying that a theory is "modeled on a computer"—risk conflating the two mappings, thus blurring the distinction between analog and symbolic simulation, as does the notion of representation when applied to analog simulations .... Thus we may say that a user (or an observer) recognizes an analog simulation as such when properties in the analog device are held to *represent* the corresponding properties (those that play the corresponding causal role under the causal-structural isomorphism). In the case of symbolic simulation, once the user identifies the phenomenon to be simulated, the term of the theory coded into the computer may be held to *represent* the relevant features of the phenomenon under its standard (intended) interpretation; here the notion of *representation* stands for language-world semantical relations such as *reference* and *denotation*. An obvious distinction can be made between these semantical relations and the notion of isomorphism of causal structure used in characterizing naturalistic analog simulation. (Trenholme, 1994, p. 119)

Once the distinction between analog and symbolic representation is clear, we can begin to see how this argument relates to the larger questions concerning Computationalism.[3]

There are at least two significantly different ways to elaborate the isomorphism between a simulation and the system or process it simulates.[4] One is the simple input/output correspondence, or what Ashby first called the "black box" simulation. Under this view, one system is a black box simulation of another if it produces similar outputs under similar inputs. In the case of analog simulations, the inputs and outputs are causal, in symbolic simulations they are symbolic. The matter is more complicated in the second type of isomorphism in which the *internal processes* of the simulation and the system or process that are being simulated have to correspond. In this form of simulation not only do the input/output relations matter, but also the internal states and processes of the simulation. Such cases make the difference between analog and symbolic simulations clearer.[5] While symbolic simulations must realize symbolic processes, these are independent from their physical realization. We generally refer to this feature as "multiple realizability," meaning that different causal systems can realize the same symbolic process. For analog simulations, which lack an independent symbolic level, the causal processes that realize the simulation must be isomorphic to the system being simulated for it to count as a simulation.

Let us turn from this abstract discussion of simulation to some more concrete historical examples. We will see in the ideas of von Neumann and Turing two different kinds of symbolic simulation. For von Neumann, we ought to use the computer to simulate a *physical model* of the brain, while for Turing the computer ought to simulate the same essential algorithm, or program, that the brain itself simulates.

3. Analogies, Digits, and Numbers

The distinction between analog and digital is due to John von Neumann (Aspray, 1990). His work on the first stored-program computers and views on the relationship between the computer and the brain presents a complex history involving numerous layers of analogy, modeling, and simulation. A complete account of the development of his thought is well beyond the scope of the present chapter, but it is worthwhile to consider how the view of working models and simulations would apply to the different positions he stated. Clearly, his design of the first computer memory drew heavily upon his review of research in neurophysiology, and quite explicitly upon the work of Warren McCulloch and Walter Pitts (1943) and their conception of logical neural networks. This much is clear from his 1945 EDVAC design proposal (1993). The story becomes more complex when we consider his 1948 theory of automata (1951), his 1946 letter to Norbert Wiener about simulating the brain (1997[1946]), and his later reflections on the relationship between the brain and the computer for his posthumously published Silliman lectures (1958). In this section I will briefly review some of von Neumann's thoughts on the issues involved in thinking about the computer as being modeled after the brain, and his approach to simulating the brain with a computer.

The key to understanding the development of von Neumann's ideas on the relationship between the brain and the computer is to keep in mind that for him the computer is a representational system of a specific type. In his automata theory, the computer is fundamentally an automatic system for representing and manipulating *numbers*. Numbers are not the same thing as quantities or numerals—they are abstract mathematical entities. Quantities and numerals are concrete ways of representing numbers for practical purposes. To automate mathematics we need to develop a physical system that can represent numbers; before we consider the technological possibilities, the choice between quantities and numerals is an open question.

On 20 September 1948, von Neumann presented a theory of automata at the *Hixon Symposium on Cerebral Mechanisms in Behavior* (1951). In his presentation, "The General and Logical Theory of Automata," von Neumann spelled out the need he saw for a rigorous theory of computation, and outlined a formal theory of *automata* (axiomatically idealized computational mechanisms). He began by distinguishing two general classes of automata by their mode of representing numbers. The class of automata built on the "analogy principle" represents numbers by analogy. They use some physical quantities in the same way a thermometer represents temperature by the height of its mercury column. If, for example, we are representing two numbers by the electrical currents in two circuits, we can add the numbers by combining the circuits appropriately and we will register the result as the total current output from the combined circuit. We can do all the basic arithmetic operations ($+, -, *, \div$) in roughly this way by using the currents in a circuit and providing the appropriate configurations of the circuit's relays and switches. Automata built on the "digital principle," instead, do not represent numbers as physical quantities, but as aggregates of numerical digits in the manner human beings typically do when they write them down on paper or count on their fingers (the etymological origin of "digit" itself). Such an automata might have a dial with ten positions representing the range from 0 to 9, or a series of such dials for the ones column, tens column, hundreds column, and so on, of the decimal numbering system. The digital representation that nearly all modern computers use is a binary system where wires, which carry electrical currents of two sufficiently distinct magnitudes, employ a set of canonical circuits to perform mathematical and logical operations on the binary representations.

The two types of automata that von Neumann described correspond to Trenholme's (1994) "analog *vs.* digital" distinction, though Trenholme, distinguishing between types of simulation, criticizes it as spurious. In each case, we seek to represent numbers and, unless we want to "simulate" pure mathematics, both are forms of symbolic simulations and both are caught up in a double-mapping. Even if we use an analog computer, such as a differential analyzer, we still need a theory about how the causal structure of the analog computer realizes the desired mathematical calculations, as well as a theory about how those calculations relate to the natural phenomena in question. A differential analyzer is not a simulation, for instance, of some hydrodynamic system, but an automated device for solving a set of equations derived from a theory of that hydrodynamic system.

The main consequence of the different kinds of numerical representation in automata is the practical realization of them when building electronic computers. Though analog computers have the advantage of arbitrarily high precision, the constant introduction of noise into their circuits make them deeply susceptible to errors. When dividing, as when dividing "5" by "7," we cannot express the result in a finite number of digits and so the number of digits that a computer can represent at one time will limit a digital computer's answer, while an analog computer can represent the answer to an unlimited precision.[6] The down side is that this is only an ideal; in actuality to get analog circuits to perform calculations with much accuracy was quite difficult, because slight fluctuations in the physical quantities, due to sources outside the computation, build up as noise in the system. Consequently, while our analog division of "5" by "7" might be perfectly precise in theory, even low probabilities of very small disturbances in the current of the circuit will limit the actual precision enormously. Von Neumann insisted that to maintain a reliable signal-to-noise ratio that prevents the analog computer from becoming more precise than a few decimal places is a difficult engineering problem, and that it just so happens that the same problem does not arise in digital machines—which have the further advantage that we can increase precision indefinitely and economically by increasing the number of digits represented by duplicating components.

While we often overlook this point, von Neumann claims that, in spite of their great practical differences, he sees no essential mathematical difference between the two kinds of automata. Any kind of calculation that we can, in principle, do on an analog automata we can also do on a digital automata, and *vice versa*. Von Neumann argues that organisms, natural automata, are *mixed* automata utilizing both principles for different particular functions, and that high speed computers are strictly digital machines. He is careful to point out that trying to base a theory of mind on the kinds of computations achievable in digital computers should not affect its applicability to the living brain since research has shown that neuronal activity is digital and, even if it were not, an appropriately constructed analog machine could realize the theory anyway. In other words, a computational simulation of the brain will be a symbolic simulation, as long as that simulation depends on the computation of theoretical models, regardless of whether we use an analog or digital computer.

4. Modeling the Computer on the Brain

> *These repeated excursions into biological information processing and the interdisciplinary study of cybernetics have been ignored in previous accounts of von Neumann's computing, yet they clearly shaped his ideas.* (Aspray, 1990, p. 189)

This section will digress from my discussion of simulations, but it will still be relevant, I hope, to the extent that it will consider the practice of "modeling" in the development of the computer as a machine. In particular, I will tell the story of how the computer was modeled on the brain, or at least upon some specific theories of how the brain and its neurons work. The stored-program computer required something more than numerical representation to become a technological reality. Namely, the stored-program computer would need to represent its *instructions* as numbers, which it could then store in memory, retrieve, execute, alter, and so on. In other words, some numbers are *representations to the computer* of instructions that the computer must interpret and perform. This great leap in design turned mere calculators into universal computers and also sowed a deep confusion regarding the representational character of computations. The connections between computational memory and neural structures in the brain soon fertilized this early kernel.Von Neumann's description of the EDVAC design made clear that in the design of the first computer memory he looked to the brain and to McCulloch and Pitts neuron for inspiration. This led to a peculiar kind of analogy between the computer and the mind/brain that the proponents of Computationalism would soon embrace, but which von Neumann himself would ultimately reject.

While researchers had constructed different electromechanical calculators before, during, and after World War II, the principal technological impediment to creating a general purpose computer (a working Universal Turing Machine) was with how to give it a memory.[7] The problem was one of both organization and technological realization. We generally hold that the first machine to solve the memory problem was von Neumann's Institute for Advanced Study machine. Von Neumann solved the problem by introducing a memory unit, or "organ" as he called it, as a central part of the computer architecture. The representation, or storage, of logical instructions or "codes," as well as numbers, distinguished the memory organ from numerical counters. The computer could, by following the instructions in memory, calculate partial solutions, store the intermediate results, reconfigure itself to perform a new function, and resume the calculation with the instructions that the memory stored as numbers (Goldstine and von Neumann, 1987[1946]). The advantages over having to reconfigure the machine by hand for each portion of a calculation, or having to build a computer so complicated that all the necessary steps in a computationally demanding problem had to be preprogrammed were obvious.

We often overlook the extent to which von Neumann drew upon McCulloch and Pitts neuron model and the neurological language he used to describe the new design. Instead, we see a clear case of modeling on—the transfer of a theory in one domain to the design of a technology in another domain. As the language he uses to describe the computer unmistakably manifests, von Neumann modeled his design after neurophysiological theory and McCulloch and Pitts neuron:

> The three specific parts CA [Central Arithmetical organ], CC [Central Control organ], and M [Memory organ] correspond to the *associative* neurons in the human nervous system. It remains to discuss the equivalents of the *sensory* or *afferent* and the *motor* or *efferent* neurons. These are the *input* and the *output* organs of the device, and we shall now consider them briefly. (1993, p. 20, emphasis in the original)

Most notably, von Neumann does not refer to the parts of the computer's architecture as "units," "components," "modules," or any other such connotatively neutral engineering terms. As a mathematician, he chose the semantically loaded term, taken from the description of biological systems, "organs." These organs explicitly *correspond to specific elements in the human brain*—a theme that his description of the machine carries out:

> It is worth mentioning, that the neurons of the higher animals are definitely elements in the above sense. They have all-or-none character, that is two states: Quiescent and excited. They fulfill the requirements of [section] 4.1 with an interesting variant: An excited neuron emits the standard stimulus along many lines (axons). Such a line can, however, be connected in two different ways to the next neuron: First: In an *excitatory synapses*, so that the stimulus causes the excitation of that neuron. Second: In an *inhibitory synapses*, so that the stimulus absolutely prevents the excitation of the neuron by any other (excitatory) synapses. The neuron also has a definite reaction time, between the reception of a stimulus and the emission of the stimuli caused by it, the *synaptic delay*. Following W. Pitts and W. S. MacCulloch [*sic*] ("A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophysics*, vol. 5 [1943], pp. 115–133.) we ignore the more complicated aspects of neuron functioning: Thresholds, temporal summation, relative inhibition, changes of the threshold by after effects of stimulation beyond synaptic delay, *etc.* It is, however, convenient to consider occasionally neurons with fixed thresholds 2 and 3, that is neurons which can be excited only by (simultaneous) stimuli on 2 or 3 excitatory synapses (and none on an inhibitory synapses). It is easily seen, that these simplified neuron functions can be imitated by telegraph relays or by vacuum tubes. Although the nervous system is presumably asynchronous (for the synaptic

> delays), precise synaptic delays can be obtained by using synchronous setups. (*ibid.*, p. 24)

Clearly, von Neumann is making a serious consideration of the structure and function of biological neurons, albeit he idealizes them in the manner of McCulloch and Pitts neuron model. Here we see how he uses the functional identity of neuron activity and mathematical logic (a unification of neural mechanisms and Turing machines) as a modeling tool in the design of the electronic computer. He sees its usefulness as a model as metaphorical or analogical, to be sure, but this kind of model is powerful, because it influences design *decisions*, and not merely the construction of design *alternatives*. In that sense it is evaluative or normative:

> The analogs of human neurons, discussed in 4.2–4.3 and again referred to at the end of 5.1, seem to provide elements of just the kind postulated at the end of 6.1. We propose to use them accordingly for the purpose described there: as the constituent elements of the device, for the duration of the preliminary discussion. We must therefore give a precise account of the properties which we postulate for these elements. (*ibid.*, p. 30)

But he saw this model as not the only criteria that is operative in design decisions:

> At this point the following observation is necessary. In the human nervous system the conduction times along the lines (axons) can be longer than the synaptic delays, hence our above procedure of neglecting them aside of t would be unsound. In the actually intended vacuum tube interpretation, however, this procedure is justified: t is to be about a microsecond, an electro-magnetic impulse travels in this time 300 meters, and as the lines are likely to be short compared to this, the conduction times may indeed be neglected. (*ibid.*, p. 30)

McCulloch's and Pitts's neuron model evidently almost completely circumscribed von Neumann's conception of physical computation.

What conclusions can we draw from these observations about the significance of the McCulloch and Pitts neuron model on the design of the first computers? First, the computer was from its very conception a kind of model of the brain. Researchers in Artificial Intelligence or Cognitive Science did not "discover" any analogy between the mind and computer. It was always there. Instead, they reconfigured it in an effort to develop an analogy between computer programs and psychological theories. One way of approaching the question of Computationalism might be to ask whether the computer itself, regardless of the program or simulation it is performing, is a model of the mind, or a good model of the mind. It obviously could be, and has been, such a model. The question of whether the mind is a computer, or a computer is a mind, however, remains. But almost no one holds the view that every computer is a mind—even strong AI holds that we must properly program computers. Turing appears to have held the view that minds are Universal Computers, and perhaps also the converse, that Universal

Computers are at least capable of being minds. I will consider his views shortly. First I will consider whether the computer may somehow simulate the brain so efficiently that it becomes a mind.

5. Simulating the Brain on the Computer

I would like to consider at this point how von Neumann envisioned the computer as a simulation of the brain. Even though he had based his automata theory on a strict distinction between analog and digital numerical representation—and modeled the memory of the stored-program computer after McCulloch and Pitts's essentially digital model of neurons—in other writings he made clear that the brain itself was much more complicated than these theoretical idealizations let on. As a result, the simulation of the brain by a computer would be far more elusive.

On 29 November 1946, just after the second Macy Conference, von Neumann wrote a letter to the mathematician and cybernetician Norbert Wiener in which he assessed the situation regarding the theory of biological information processing, and the replication of the brain's abilities in a computer. The letter is interesting for several reasons, and warrants more thorough examination than our current purposes permit. Despite his early enthusiasm, von Neumann was perhaps the first to realize the limitations of computers for simulating the brain. The clearest articulation of the shortcomings of this approach comes in his letter to Wiener. Here von Neumann argued that directly modeling the physical structure of the brain will be so complicated as to be nearly hopeless, an idea he would express more explicitly in later work. He concluded from this situation that a much better approach would be to turn instead to detailed cytological work. Specifically, he proposed to understand simple organisms in complete atomic detail (literally), by starting with the study of bacteriophagic viruses.

In the letter, von Neumann intimated one of the primary reasons for the difficulty in making the analogy between brains and computers:

> Besides, the [brain] system is not even digital (i.e. neural): It is intimately connected to a very complex analogy; (i.e. humoral or hormonal) system, and almost every feedback loop goes through both sectors. If not through the "outside" world (i.e. the epidermis or within the digestive system) as well. (1997[1946], p. 507)

Two crucial points in this passage call for attention. First, von Neumann was keenly aware of the embedded and situated nature of the human brain, and that its information processing relies on feedback loops with its environment. Unfortunately, von Neumann did not direct his energies in pursuing this issue. The other point is that the supposedly neat distinction between analog and digital was not so clear in the living brain and von Neumann directed a great deal of his energy in this direction.

One significant consequence of McCulloch and Pitts neuron model was to establish the digital character of neuronal behavior in the brain as the character relevant for understanding its organization. While this move had great benefits in terms of formalizing neural networks, and thereby treating them using mathematical logic, the idealization was ultimately a gross simplification. Von Neumann stated the case most devastatingly in his Silliman manuscript. In one passage, he assailed the notion that it is safe to treat the mechanism of neuronal excitation and inhibition as a straightforward summation function:

> It may well be that certain nerve pulse combinations will stimulate a given neuron not simply by virtue of their number, but also by virtue of the spatial relations of the synapses on a single nerve cell, and the combinations of stimulations on these that are effective (that generate a response pulse in the last-mentioned neuron) are characterized not only by their number but also by their coverage of certain special regions on that neuron (on its body or its dendrite system, *cf.* above), by the spatial relations of such regions to each other, and by even more complicated quantitative and geometrical relationships that might be relevant. (1958, pp. 54–55)

While it might be tempting to treat things as if all inputs to a neuron were equal, in fact the complex three-dimensional geometry of the synaptic connections to the neuron's dendrites is relevant to the electro-chemical processes that trigger a pulse. Similarly, the ideal of the synchronous timing of neuronal activity, essential to McCulloch and Pitts's assertion that we can treat multi-layered networks as equivalent to logical propositions, was also untrue:

> On all these matters certain (more or less incomplete) bodies of observation exist, and they all indicate that the individual neuron may be—at least in suitable special situations—a much more complicated mechanism than the dogmatic description in terms of stimulus-response, following the simple patterns of elementary logical operations, can express. (1958, p. 56)

So while he found the suggestions of treating neurons as logical units performing summations useful for designing computer memory circuits, he was deeply disturbed by just how remote this idealization was from real brains when it came to building a simulation.

Not only were the neurons not performing the idealized functions that McCulloch and Pitts required, even von Neumann's own distinctions between analog and digital automata applied to the brain in complex, and by no means straightforward, ways:

> The observation I wish to make is this: processes which go through the nervous system may, as I pointed out before, change their character from digital to analog, and back to digital, etc., repeatedly. Nerve pulses, i.e. the digital part of the mechanism, may control a particular stage of such a process, e.g. the contraction of a specific muscle or the secretion of a

> specific chemical. This phenomenon is one belonging to the analog class, but it may be the origin of a train of nerve pulses which are due to its being sensed by suitable inner receptors. When such nerve pulses are being generated, we are back in the digital line of progression again. As mentioned above, such changes from a digital process to an analog one, and back again to a digital one, may alternate several times. Thus the nerve-pulse part of the system, which is digital, and the one involving chemical changes or mechanical distortions due to muscular contractions, which is of the analog type, may, by alternating with each other, give any particular process a mixed character. (1958, pp. 68–69)

Ultimately, these complexities led von Neumann to believe that the study of the brain would lead to a new mathematics.

Von Neumann begins the letter to Wiener with a devastating critique of his own work, as well as that of Wiener, Turing, McCulloch and Pitts, to formulate a substantive theory of information processing in the brain:

> Our thoughts—I mean yours and Pitts' and mine—were so mainly focused on the subject of neurology, and more specifically on the human nervous system and there primarily on the central nervous system. Thus, in trying to understand the function of automata and the general principles governing them, we selected for prompt action the most complicated object under the sun—literally .... The difficulties are almost too obvious to mention: They reside in the exceptional complexity of the human nervous system, and indeed of any nervous system. (John von Neumann, 1997[1946], pp. 506–507)

Evidently, von Neumann recognized that the lack of formalized scientific understanding of the operation of the neurons was a major hurdle to constructing synthetic brains with artificial automata. The consequence of his efforts in clarifying these ideas over the course of several years was one of exasperation:

> What seems worth emphasizing to me is, however, that after the great positive contribution of Turing-cum-Pitts-and-McCulloch is assimilated, the situation is rather worse than better than before. Indeed, these authors have demonstrated in absolute and hopeless generality, that anything and everything Brouwerian can be done by an appropriate mechanism and specifically by a neural mechanism—and that even one, definite mechanism can be "universal." (*ibid.*, p. 507)

Besides being perhaps the most devastating critique of Turing's project before the rise of AI, this passage clearly stated the paradoxical relationship between the universal computer and the synthetic brain, which I will consider in the next section. The critique was this: the universal computer, and its equation with the functioning of the brain is infinitely potent, yet flaccid. The "absolute and hopeless generality" of the theory is such that while it could replicate the processes of mind, if those are formally definable (Brouwerian), it gives absolutely no insight into the nature, structure, or organization of those processes. The universal computer can imitate any machine, and thus the brain-machine, but this tells us nothing about brains. Instead, a rigorous theory of the biological brain was in order:

> Inverting the argument: Nothing that we may know or learn about the functioning of the organism can give, without "microscopic," cytological work any clues regarding the further details of the neural mechanism. (*ibid.*, p. 507)

Ultimately, the study of neural networks in the abstract has left us with the fundamental problem of understanding the brain empirically. Our theoretical understanding of neuroscience limits the simulation of the brain on the computer—unsurprisingly so, I would add, when we recognize that von Neumann's approach to brain simulation was a form of symbolic simulation, and hence a theoretical simulation. His notion of a computer simulation of the brain completely depended on a theoretical model of the brain that a symbolic representation on a computer could map. We see a clear example of the alignment between the symbolic simulation and the theoretical model. I will now turn to a different conception of how the computer can simulate the brain, which is nonetheless a symbolic simulation based on a theoretical model.

6. Computer as Universal Modeling Machine

> *It is possible to invent a single machine which can be used to compute any computable sequence. (Turing, 1937, p. 127)*

After the examination of the complex relationship between the computer and the brain in von Neumann's work, I turn to the more straightforward, if more abstract, relationship between the computer and the mind in Alan Turing's work. Put simply, Turing was preoccupied with his idea of the Universal Machine (now called Universal Turing Machines or UTMs). The idea of the UTM that Turing described in (1965 [1936]) is of an abstract formal computer able to compute any computable function. Elsewhere he refers to the ability of the UTM to "model" any other machine. Our earlier distinction between analog and symbolic simulation circumscribes this way of talking. The UTM is a purely symbolic conception, and real working computers are only approximations of this mathematical formalism. Turing did, however, seek to design and build real computers that were based on this formalism, most notably the Advanced Computational Engine (ACE), and later the Mark I at Manchester (Carpenter et al., 1986).

In nearly every speech and paper related to computation that Turing produced from 1944 to 1950, he cited the similarity between his UTM and a programmable digital computer as being highly significant:

> Some years ago I was researching on what might now be described as an investigation of the theoretical possibilities and limitations of digital

> computing machines. I considered a type of machine which had a central mechanism, and an infinite memory which was contained on an infinite tape. This type of machine appeared to be sufficiently general. One of my conclusions was that the idea of a 'rule of thumb' process and a 'machine process' were synonymous. The expression 'machine process' of course means one which could be carried out by the type of machine I was considering. It was essential in these theoretical arguments that the memory should be infinite. It can easily be shown that otherwise the machine can only execute periodic operations. Machines such as the ACE may be regarded as practical versions of this same type of machine. There is at least a very close analogy. (1992, pp. 106–107)

Turing saw universality as the critical element of UTMs—their ability to model any other machine made them so remarkable. The mathematical theory of UTMs had to be the model for the design of computing machines —the only difference in Turing's mind between the mathematical formalism of UTMs and the physical computer was the finite size of the memory, and the additional physical limitations of the machine, such as time. Apart from these practical constraints, we might easily call Turing's vision of the computer a "universal modeling machine."

His focus on universal modeling was also central to his consideration of learning and intelligence, and how to model them on a computer. He cast the problem as being that of an unorganized system becoming organized. But instead of accepting a thermodynamic interpretation of organization, as the cyberneticians would, Turing's approach sought to show how unorganized systems could efficiently organize themselves into UTMs through reinforcement by pleasure and pain.

Turing applied the same conception of universal modeling machines to the human brain as he did to the design of computers:

> All of this suggests that the cortex of the infant is an unorganized machine, which can be organized by suitable interfering training. The organizing might result in the modification of the machine into a universal machine or something like it. This would mean that the adult will obey orders given in appropriate language, even if they were very complicated; he would have no common sense, and would obey the most ridiculous orders unflinchingly. When all his orders had been fulfilled he would sink into a comatose state or perhaps obey some standing order, such as eating. Creatures not unlike this can be found, but most people behave quite differently under many circumstances. However, the resemblance to the universal machine is still very great, and suggests to us that the step from the unorganized infant to a universal machine is one which should be understood. When this has been mastered we shall be in a much better position to consider how the

> organizing process might have been modified to produce a more normal type of mind. (1992, p. 16/120)

For Turing, it appears that the key to unlocking the secrets of the mind lay with the UTM. For him, the mind had the ability to model other machines, and was thereby a universal model of a particular sort, though perhaps not completely identical to the uncreative computer.

We can safely contrast this approach to intelligence and mind to that of W. Ross Ashby, who saw behavior not in terms of logical rule-following, but as trajectories in phase space. He sought to embody his ideas in machines that he could directly interact with, such as the Homeostat, and later the DAMS. Learning for Ashby was a never-ending dynamic process in which the goal was survival, and the environment was continually changing. Modeling was a consequence of this, not necessarily the cause. Turing saw learning as a search for a single stable goal—the organized universal computer. For Turing, it was the patterns of symbols in memory, the symbolic "instruction tables" which held the secrets of the mind, while for Ashby, it was the patterns of interactions and feedback loops between the system and its environment. In both cases, they could build a machine to embody their vision. However, for Turing this machine would epitomize the symbolic simulation of mind, while for Ashby the machine would epitomize the analog simulation of mental behavior. We can best understand this contrast by considering their letters to one another.

7. Simulating the Homeostat

When Ashby returned from his service with the Royal Medical Corps in India in the Spring of 1946, he had been concerned with the mechanisms of learning for over a decade and had begun thinking about constructing a machine to demonstrate the principles he had concluded to be essential to adaptation. During that year he filled his notebooks with mathematical formalisms for different stability-seeking systems. These eventually turn to diagrams of simple, and then more complex, electrical circuits to realize the behavior of a set of equations in a system that he could directly engage with manually (Ashby, 1948; Ashby, 1952b; Asaro, 2008).

In the fall of 1946, Ashby wrote to Turing about the adaptive machine Ashby was designing.[8] As with all other machines and processes, Turing believed that a programmable computer such as the ACE could model Ashby's Homeostat, as he explained in his reply to Ashby on 20 November 1946:

> The ACE will be used, as you suggest, in the first instance in an entirely disciplined manner, similar to the action of the lower centres, although the reflexes will be extremely complicated. The disciplined action carries with it the disagreeable feature, which you mentioned, that it will be entirely uncritical when anything goes wrong. It will also be necessarily devoid

> of anything that could be called originality. There is, however, no reason why the machine should always be used in such a manner: there is nothing in its construction which obliges us to do so. It would be quite possible for the machine to try out variations of behaviour and accept or reject them in the manner you describe and I have been hoping to make the machine do this. This is possible because, without altering the design of the machine itself, it can, in theory at any rate, be used as a model of any other machine, by making it remember a suitable set of instructions. The ACE is in fact, analogous to the 'universal machine' described in my paper on computable numbers. This theoretical possibility is attainable in practice, in all reasonable cases, at worst at the expense of operating slightly slower than a machine specially designed for the purpose in question. Thus, although the brain may in fact operate by changing its neuron circuits by the growth of axons and dendrites, we could nevertheless make a model, within the ACE, in which this possibility was allowed for, but in which the actual construction of the ACE did not alter, but only the remembered data, describing the mode of behaviour applicable at any time. I feel that you would be well advised to take advantage of this principle, and do your experiments on the ACE, instead of building a special machine. (1946)

This letter did not stop Ashby from pursuing his machine and, in fact, shortly after he received this letter he came up with one of the basic elements of the Homeostat circuit. The difference of perspective and approach between Ashby and Turing is now clearer. Turing's insistence, even while admitting the importance of learning to intelligent behavior in general, on the universal character of the ACE meant that there was no point in building a special machine of the type Ashby was proposing. A tension exists between the rigid determined behavior of the ACE and the adaptive behavior of the brain. Considering how the ACE could model the Homeostat will bring into focus the differences between analog and symbolic computation.

While Turing's letter seems to move casually from talking about the ACE modeling the brain to talking about the ACE modeling the Homeostat, we should pause to consider the difference. This difference is the essence of the analog and symbolic distinction: Turing did not believe that the unprogrammed computer, which in spite of some resemblance to the "lower centres" was too rigid and disciplined to be brain-like, was a model of the brain. Instead, the computer required the proper instruction tables, the program defining the machine of the brain. And while Turing had some idea that conditioned behavior might be the base of it, he had little idea how to program his machine to behave in this way. The ACE might be able to perform a symbolic simulation of the brain, but only if it receives the proper symbolic organization. Ashby's Homeostat instead, because of its causal and not its symbolic structure, was itself a model of the brain. Turing saw in the Homeostat the potential to model the brain indirectly. Because the Homeostat was a machine whose design we understood, we should be able to simulate it symbolically on the Ace. Because the Homeostat simulated the brain, the Ace's simulation of the Homeostat would be a simulation of the brain (once removed). The level of indirection in this case was the Homeostat itself as a theory of the brain's adaptive behavior. Any simulation that the Ace performed would only be as good as the theory upon which it relied. Simulations told us something about the Homeostat, and only indirectly something about the brain. But we may still wonder about the advantages to the directness we got from the analog simulation that the Homeostat provided. Several were the reasons why the analog Homeostat was a better model of the brain than the symbolic Homeostat that the Ace simulated. Ashby and Turing faced first practical issues, then methodological issues of scientific practice, and finally the issue of the effectiveness of the two simulations as a demonstration. I will now turn to these questions.

First is the question of the practicality of building each simulation. As it turned out, the Homeostat was far easier to build than the Ace. Pilot Ace would not become fully operational until May of 1950, two years *after* the Homeostat was demonstrated to the Eeg Society in Bristol (von Neumann's machine was not fully operational until 1952, though this was due more to the perpetual redesign of its parts). In addition, the Homeostat was a far more economical solution than the Ace, even if it had only one application. Ashby was an intellectual entrepreneur who built a significant machine on his kitchen table out of surplus parts from the war, in a self-financed project. While this was true then, the opposite is true now, when computers sit on the majority of desks and finding the analog circuit components that Ashby used would be extremely difficult. Therein lies the true advantage of the symbolic computational simulation—because the computer potentially suits everyone's modeling needs, we can produce them cheaply in quantity and thus, despite their vastly more complex structure, they turn out to be cheaper and easier to build than specialized models. By 1946, Turing had this much down. The quality of the symbolic simulation when compared to the analog was less clear, though.

In his analysis of simulations (2003), Winsberg argues that often only the local outcomes of a specific situation, not the mathematical theory underlying the simulation, are in question. One of the great advantages of simulations for these cases is the ability to visualize the model, what R.I.G. Hughes (1997) calls their mimetic properties. This advantage is due to the ability of researchers to employ experimental practices (most notably observation) and laboratory techniques in understanding the local simulations, as opposed to depending upon mathematical practices and analytic techniques. Yet these mimetic properties are dependent upon the development of instrumentations within the simulation to make those properties visible. The symbolic processes themselves are rather opaque and not susceptible to direct observation, thus entailing more levels of indirection in the interpretation and visualization of data that simulations produce. From an epistemic perspective, we are getting further and further away from the phenomena in question. Consider the Homeostat and its ACE simulation. We can directly interact with and observe the Homeostat. But we must preprogram, any interactions with the ACE simulation, and we must interpret the results from the symbolic output of the machine. While today we could write a simple Java program with a colorful on-screen visualization of the Homeostat, this was hardly conceivable in 1946—which just proves the point that any such symbolic visualization introduces another layer of interpretation.

Other important practical issues of time, which the increasing sophistication of computing machines is similarly obscuring, are also present. First are the issues of mathematical complexity and computation time. To simulate the behavior of the Homeostat's circuits using the equations of electrical engineering, or physics, would probably have taken far too long for a machine like ACE to perform in real-time (something approximating the response time of a real Homeostat). The significance of having a real-time simulation relates to the mimetic property mentioned above. Namely, we can interact with a real-time system in ways that we must simulate in false-time systems. Science does not always need this, but it is extremely important for understanding certain phenomena. Real-time here applies to the observers' time frame and ability to act upon the system, receive feedback from that action, and evaluate the relationship of the two. This feedback loop must be real-time for the observer, even if the time-frame of the simulated phenomena is greatly sped-up (as in astronomical phenomena like the movement of the planets) or slowed-down (as in high-speed phenomena like protein-folding). The significance of this aspect of analog models becomes even more apparent when we consider the idea of a regulator as model.

8. Conclusions: Simulation and Computationalism

Von Neumann saw the power of mathematics to simulate detailed physical models, and ultimately to allow for the control of enormously complex systems involving many dynamically interdependent variables. One of the goals of the computer he built was to automate the mathematical calculations needed to simulate these systems so as to allow mathematicians to develop ever more complex and detailed simulations. The brain, too, was a natural phenomenon that could be so simulated, once we understood it scientifically. The difference between Turing and von Neumann was that von Neumann felt that Turing's universal machine concept added almost nothing to our understanding of the mind because it told us nothing about the brain. Without a clear understanding of the brain's operation, the computer would never embody an artificial mind.

Von Neumann ultimately came to see the task at hand to be one of modeling the *detailed microstructures of brain*. For him, the computer was not itself a brain, though his design for the EDVAC computer drew heavily on analogies to brain structures and neuronal functioning. The crucial aspects of the design of the computer were technical considerations of scalability, reliability, and numerical accuracy as computers were given more memory and faster processing speeds. He quickly realized that the scale of the simulations that would be needed for brains would require vast numbers of calculations, and even small or infrequent errors would result in critical failures. He also quickly became skeptical about the pace of progress in building simulations of the brain.

Turing's view of the computer was as the universal modeling machine. But rather than worry about modeling the brain in physical detail, he sought to simulate its behavior symbolically. Turing and von Neumann sought after symbolic simulations of the brain, but approached these in a remarkably different way. Ashby remained committed to analog simulations through most of the 1950s, but appeared to give up on them after his frustration with DAMS to accept that the increasing capabilities of digital computers better suited his needs.

Another way of thinking about the difference between analog and symbolic simulations is to note that the symbols of a simulation inside a computer are not connected to the world in the same way as the causal structures of an analog simulation are. For the Homeostat, these connections were highly relevant to Ashby's theoretical views of the mind.[9] Turing's claim that the distinctions are irrelevant thus bespeak a significant theoretical disparity between the two men. The best and simplest way of characterizing this difference is in terms of the relationship between a system and its environment, which for Turing is irrelevant, yet for Ashby is central. It is a distinction of theoretical and practical import. Theoretically, Ashby starts from the rich complexity of relationships between system and environment, while Turing considers the environment only in the simplest and most idealized terms possible. Practically, Ashby was interested in building a machine that an observer could directly interact with in real-time. Such a direct interaction was essential to his pedagogical and rhetorical aims, and to his experimentalist desires (see Asaro, 2006). For Turing, logical proof was the ultimate form of demonstration, experimentalism was not a priority, and the details of the physical machinery were inconsequential. These distinctions became even more pronounced over time. For even as Turing turned to deeper considerations of learning and adaptation, he became more frank about his reasons for avoiding issues of general intelligence, human behavior, and human-like bodies for computers (Turing, 1950; Turing, 1953; Teuscher, 2002).

NOTES

1. A classic example are the Monte Carlo simulations devised by John von Neumann, Stanislaw Ulam, and Enrico Fermi in the development of the atomic and hydrogen bombs.

2. I will call these "analog simulations," but will be careful to distinguish them from analog representations and analog computations, as is necessary.

3. For instance, the kind of argument made by Searle, 1980, in his famous Chinese-room thought experiment turns on just these semantic aspects of symbols. Searle aims to show that intentionality is an essential element of symbolic representation and that Computationalism, or at least that the strong program in AI fails to explain it.

4. I use isomorphism here not in its strict sense of correspondence, but in the looser sense of similarity proposed by Teller, 2001.

5. For black box simulations, we do not care about internal processes so it makes little difference whether these are achieved symbolically or causally, or even magically.

6. Computer programmers call this a "truncation" error—when the number of digits needed for an answer exceeds the number the machine can represent. In division, this generally results in a truncation or rounding-off of the decimal positions after the last available digit is used. Consider a machine multiplying two 10-digit numbers, and capable of only representing 10 digits at a time. The result of the multiplication will be at least 18 digits long, but the machine can only represent half of these, and hence can not represent a meaningful answer at all—or must convert it to scientific notation, truncate it, and thereby lose precision.

7. The early computing machines were highly specialized to perform a single class of functions or, like ENIAC, had to be programmed manually by arranging walls of dials and networks of patch cables in a fashion similar to hand-operated switchboards for telephones. For instance, Howard Aiken's Mark I Automatic Controlled Sequence Calculator, built for IBM in 1944, used 72 rotary counters for storing numbers mechanically, but had no means for storing the calculation instructions themselves—the "function" resided in the mechanical configuration of the device.

8. Unfortunately, this letter has been lost.

9. See Ashby, 1952b; Ashby, 1956a; Ashby, 1961; Ashby, 1962; Ashby, 1967; Ashby, 1968; Ashby, 1972; Asaro, 2008; Asaro, 2009.