Chapter 2: Advantages And Disadvantages Of Modeling

2.1 WHAT IS A MODEL?

In science, phenomena that are difficult to study or to understand in their own right are often approached through the use of models. The kinds of models that are used are as diverse as science itself. In biology, model organisms are used to study processes that cannot be easily measured in humans. In engineering, models of physical structures are tested in wind tunnels. Some might argue that physics is concerned with the development and testing of mathematical models of physical systems

Even within a single discipline, one can find a bewildering diversity of model types. For example, in psychology, computer simulation models have been created for many cognitive phenomena (Boden, 1977; Feigenbaum & Feldman, 1995; Grossberg, 1988; VanLehn, 1991). Mathematical models have been used to study human perception, learning, judgments and choice (Bock & Jones, 1968; Caelli, 1981; Restle, 1971). Statistical models have become the primary tool for expressing relationships between variables (Lunneborg, 1994; Pedhazur, 1982; Winer, 1971). Model organisms, such as the long-finned squid *Loligo pealei*, have been used to help understand the generation and transmission of nervous impulses (Hille, 1990; Levitan & Kaczmarek, 1991).

A famous philosophical passage highlights the perils of defining even the simplest of terms: “Consider for example the proceedings that we call ‘games’. I mean board games, card-games, ball games, Olympic games, and so on. What is common to them all? -- Don't say: ‘There must be something common, or they would not be called 'games' ‘-but look and see whether there is anything common to all” (Wittengstein, 1953, p. 31[e]). Given the diversity that we have briefly noted above, the term ‘model’ could just have easily been used to demonstrate this point! Wittgenstein went on to argue that there was only a family resemblance between members of a category. “For if you look at them you will not see something that is common to all, but similarities, relationships, and a whole series of them at that.” The features that constitute these similarities and relationships change as different members of the same class are compared to one another. What kind of family resemblance would we find amongst the members of the class ‘model’?

Intuitively, a model is an artifact that can be mapped on to a phenomenon that we are having difficulty understanding. By examining the model we can increase our understanding of what we are modeling. “A calculating machine, an anti-aircraft ‘predictor’, and Kelvin’s tidal predictor all show the same ability. In all these latter cases, the physical process which it is desired to predict is *imitated* by some mechanical device or model which is cheaper, or quicker, or more convenient in operation” (Craik, 1943, p. 51).

For it to be useful, the artifact must be easier to work with or easier to understand than is the phenomenon being modeled. This usually results because the model reflects some of the phenomena’s properties, and does not reflect them all. A model is useful because it simplifies the situation by omitting some characteristics. “Any kind of working model of a process is, in a sense, an analogy. Being different it is bound somewhere to break down by showing properties not found in the process it imitates or by not possessing properties possessed by the process it imitates” (Craik, 1943, p. 53). Similarly, “the word model may be used instead of theory to indicate that the theory is only expected to hold as an approximation, or that employing it depends upon various simplifying assumptions” (Braithwaite, 1970. p. 269).

While a model can imitate a phenomenon, it need not resemble it. “Kelvin’s tide-predictor, which consists of a number of pulleys on levers, does not resemble a tide in appearance, but it works in the same way in certain essential respects – it combines oscillations of various frequencies so as to produce an oscillation which closely resembles in amplitude at each moment the variation in tide level at any place” (Craik, 1943, p. 51). Similarly, Galileo revolutionized science by using geometry to represent physical quantities like velocity and acceleration that do not themselves resemble lines or angles (Haugeland, 1985).

Of course, consistent with Wittgenstein’s notion of family resemblance, none of the claims made in the preceding paragraphs apply equally well to every model. For instance, some models are less analogous than others. The properties of the ionic channels in one model, the giant axon of the squid, are expected to correspond perfectly to the properties of the same channels in the human nervous system (Kuffler, Nicholls, & Martin, 1984). Similarly, some models, such as the scale models of structures that are tested in wind tunnels, have much stronger resemblances to entities in the real world than do other kinds of models.

One property that does seem common to all models, though, is the notion of predictive utility. A model is used to generate predictions that can be used to test the validity of a theory. The model is used because in some sense it provides an easier or faster route to prediction. Later in this book we will see that the many different kinds of models available to psychology can be used in a variety of ways, and that in some sense it is not correct to describe the models of synthetic psychology as providing “predictive utility”. Prior to embarking on that much longer discussion in later chapters, let us first turn quickly to considering some of the advantages and disadvantages of using models in general.

2.2 ADVANTAGES AND DISADVANTAGES OF MODELS

Modeling in psychology or cognitive science is associated with both advantages and disadvantages (e.g., Lewandowsky, 1993). In this section of the chapter, we will consider three general advantages of modeling. However, after each of these three advantages, we will follow with a discussion of associated disadvantages. Models are like fine knives with which you can create gourmet meals, but with which you can also cut off your fingers.

2.2.1 Rigorous Specification Of Theory

“Theory in a field as immature as psychology cannot be expected to amount to much -- and it doesn’t” (Royce, 1970. p. 17). There are many reasons for skepticism about the quality of psychological theory. Some researchers have argued that psychologists, envious of physics, attempted to develop quantitative theories without first laying a proper qualitative foundation (Kohler, 1975). Others would argue that whenever psychological theories are expressed verbally, they are necessarily vague and imprecise. As well, there is a long tradition in experimental psychology of being extremely wary of verbal data (Ericsson & Simon, 1984). It would not be surprising if there were an accompanying wariness of verbally or informally stated theories.

How do you make theories better? Many researchers would argue that this is accomplished by translating an informal verbal theory into a formal mathematical expression or into a working computer simulation. "Even deceptively simple models can benefit from the rigor of simulations" (Lewandowsky, 1993, p. 236).

2.2.1.1 Precision Of Terms

There are several reasons that the process of formalization is useful. First, it adds precision in specifying theoretical terms. An informal theory can be full of references to terms with vague definitions like “memory" or "attention". Many academic debates emerge because different researchers use the same terms in different ways. In a formal model, conceptual terms have to be carefully operationalized in order for the model to work. This forced precision enables

the theorist to communicate his ideas to others less ambiguously then would be the case if the theory were communicated as an informal statement.

One interesting historical example of this can be found in experimental aesthetics. One of the main goals of this discipline was to measure subjects' responses or preferences, and to relate these measurements to properties of the works of art or other objects that were presented (Berlyne, 1971). In this field, it has proven difficult to specify both properties of stimuli and properties of preferences. For example, the Gestalt psychologists introduced the notion of "goodness of configuration" with their Law of Prägnanz (Kohler, 1975). According to this law, we perceive organized patterns instead of isolated elements, and we actively organize these patterns to make them "good".

Unfortunately, the definition of good in the Law of Prägnanz was particularly vague: "psychological organization will always be as 'good' as the prevailing conditions allow. In this definition the term 'good' is undefined. It embraces such properties as regularity, symmetry, simplicity and others" (Kohler, 1975, p. 110). Berlyne (1971) revolutionized the field with formalization, in particular by characterizing stimulus properties numerically using definitions of complexity and redundancy that were taken from mathematical information theory. Berlyne took the same approach to the notion of preference, formalizing emotion in terms of arousal. Berlyne's approach led to extremely vibrant study of aesthetics by experimental psychologists in the 1960s and '70s. The renaissance of the field was largely driven by the fact that Berlyne's formalization permitted researchers in different labs to have more precise understanding of the stimulus and response properties that were being studied in diverse experiments.

2.2.1.2 New Tools For Studying Concepts

A second advantage of formalization comes from recognizing that the language in which a theory is expressed determines the kinds of ways in which the theory can be tested or explored. For instance, after a verbal theory has been formalized mathematically, one can use mathematical operations to investigate its implications (Coombs, Dawes, & Tversky, 1970; Lunneborg, 1994; Wickens, 1982). In other words, formalization not only results in a more precise specification of the concepts in the theory, but also results in a more precise set of tools for studying these concepts.

One example of this can be found in my own research on how the human visual system tracks the identity of moving targets (Dawson, 1991). In one approach, I converted a general theory of this tracking into a particular type of computer simulation. I was then able to use the simulation to generate hypotheses about what human subjects would see when presented apparent motion displays that had never been studied before (Dawson & Pylyshyn, 1988). In a second approach, I formalized the theory using some of the elementary operations of linear algebra. With this formalization, I was able to prove that the computer simulation would generate unique solutions to tracking problems. I was also able to prove that there was a strong relationship between my model and a more general model that was unrelated to motion processing (Hopfield, 1982). The algebra showed that both models could be described as minimizing identical energy functions. Both of these proofs were examinations of crucial characteristics of my theory, but would have been impossible to conduct had the model not been expressed algebraically.

2.2.1.3 Revelation Of Hidden Assumptions

A third advantage of formalization is that it can reveal hidden assumptions in an informal theory which themselves need to be fleshed out in greater detail in order for the theory to be complete. For example, many theories in cognitive psychology are expressed as flowcharts of black boxes. Ideally, each black box in such a flowchart is supposed to be a primitive operation that needs no further explanation (Cummins, 1983; Dawson, 1998). Bringing the flowchart to life

in, for instance, a computer simulation can reveal that some of these alleged primitives are themselves very complicated processes that require further analysis and explanation.

As a case in point, consider the study of vision. For most people, visual perception is extremely easy: we just look at something and see it. Because of this, artificial intelligence researchers believed in the 1960s that it would be very straightforward to build computer vision programs. "In the 1960s almost no one realized that machine vision was difficult" (Marr, 1982, p. 16). Indeed, Marvin Minsky has admitted that he assigned computer vision to a student as a summer programming project (Horgan, 1993). However, when serious attempts were directed towards programming a machine to see, astonishing difficulties arose. It became painfully obvious that underlying the process of seeing was a set of enormously complicated information processing problems that the human visual system was solving effortlessly in real time. Identifying the nature of these problems, let alone solving them, became a staggering challenge for vision researchers – and the core of a new discipline. Vision research has obviously benefited from attempts to formalize our intuitions about perceptual processing.

2.2.2 Problems With Formalization

We have seen in the preceding section that one general property of the model is that it can result in the conversion of an informal theory into a theory that is stated more rigorously or more precisely. We've also seen that there are several advantages to doing this. However, it is important to realize that the formalization of the theory can also be hazardous. Let's briefly consider some potential disadvantages of formalization.

2.2.2.1 The Irrelevant Specification Problem

One potential problem with formalization is that this process requires a researcher to make design decisions. For instance, in a computer simulation one might have many possible ways for representing information. To build a model, one of these representational formats must be selected. The hope is that the specific choice is *theory-neutral*. If the choice is theory-neutral, this means that the simulation will behave in the same manner whatever representational format is chosen. However, this is often not the case. Many design decisions are *theory-laden*. In other words, the behavior of the model is affected by the design decisions. With one representational code a computer simulation might behave one way, but it will behave differently with another representational code. Lewandowsky (1993) calls this the *irrelevant specification problem*.

To illustrate the irrelevant specification problem, let us consider a model of how human subjects perform in a particular memory task. One of the earliest techniques for studying memory was the paired-associate learning task (Ashcraft, 1989). In this task, subjects were presented pairs of consonant-vowel-consonant nonsense syllables (CVCs), such as XOP-LUD. When presented the first member of the pair, subjects' task was to remember the second member of the pair. So, when presented XOP a subject would respond with LUD. The dependent measure for this task was usually the number of trials that were required before a short list of these pairs was remembered perfectly. The paired-associate learning task was central to the study of interference theories of forgetting.

In 1961, a computer simulation of this type of memory task, called EPAM for Elementary Perceiver and Memorizer, was first described (Feigenbaum, 1995). This model used a discrimination learning process to create a discrimination net to represent remembered CVCs. This discrimination net was very similar to modern decision trees used by computer scientists for pattern recognition (e.g., Quinlan, 1986). Each branch of Feigenbaum's discrimination net was a test that would distinguish one CVC from another. Each terminal leaf of the discrimination net was one of the component letters of a CVC. During learning, EPAM would grow its discrimination net using the minimum amount of information required. As more items were added to the net, the early discrimination tasks might start to fail, which allowed EPAM to model interference effects in paired-associate learning.

One of the key design decisions in EPAM was the assumption that the primitive symbols in the discrimination net were individual letters. Feigenbaum and Feldman (1995) made this design decision for the very plausible reason "letters are familiar and are well-learning units for the adult subject" (p. 301). However, it turns out that this design decision is theory-laden. In one of my first experiences with computer simulation in a Minds and Machines course taught by Zenon Pylyshyn at the University of Western Ontario, we started with an EPAM model that used Feigenbaum's coding format. We then revised the model by making a different design decision about the internal symbols. In the revised model, we described each letter as a set of visual features. As a result, the discrimination net terminated in featural subcomponents of a CVC's component letters. The revised model had a great deal of difficulty learning any paired associates, indicating that the choice of internal representation strongly affects the model's performance.

2.2.2.2 The Relevant Formalization Problem

Hodges (1983) describes a problem that mathematician Alan Turing encountered when he formalized a method for playing chess. "Alan had all the rules written out on its paper, and found himself torn between executing the moves that his algorithm demanded, and doing what was obviously a better move. There were long silences while he totted up the scores and chose the best minimax ploy, hoots and growls when he could see it missing chances" (p. 440). This illustrates a disadvantage that I will call the *relevant formalization problem*. After you formalize a model, like Turing, you have to accept its bad properties along with the good. The relevant formalization problem occurs when this is not done, because there is a strong temptation to selectively focus on a formalization's successes, and ignore its failures.

My own experience with the relevant formalization problem came when I taught myself connectionism by programming the equations in a popular account of the generalized delta rule (Rumelhart, Hinton, and Williams, 1986b). After programming the equations, I tested my work by trying to train networks on the problems that Rumelhart, Hinton, and Williams described. To my dismay, I found that in several cases my program didn't converge to a trained connectionist network. Thinking that there must be a bug in my code, I spent a great deal of time poring over it, and was frustrated by failing to find any errors. It turned out that my code was correct, but that in many cases it was failing to converge because the network connection weights were driving the system into a local minimum.

I should have expected this, because the generalized delta rule is, in principle, subject to this kind of problem (Minsky & Papert, 1988). However, I had different expectations because, in my opinion, Rumelhart, Hinton, and Williams (1986b) had fallen into the relevant formalization problem. They reported that "we do not know the frequency of such local minima, but our experience with this and other problems is that they are quite a rare. We have found only one other situation in which a local minimum has occurred in many hundreds of problems of various sorts" (p. 332). My own experience with this kind of network is that problems like local minima are much more frequent.

Having to take the formalization seriously can be extremely productive. One excellent example of this is found in work that uses production systems to model human search of short-term memory (Newell, 1973), and is described in the paragraphs that follow.

Sternberg (1969) reported one famous study of short-term memory. In the Sternberg memory task, subjects were given a string of digits to hold in short-term memory. After a set delay, subjects were presented an additional probe digit. Their task was to say whether or not the probe was a member of the memorized list. The dependent measure in this experiment was reaction time. Sternberg found a linear increase in response time as a function of the number of digits in the memorized list. Sternberg also found that the slope of the reaction time function for lists that did not contain the probe was twice the slope of the reaction time function for lists that

did. Sternberg used these results to propose a self-terminating serial search model of short-term memory; this was one of the first experiments that demonstrated how reaction time data could be used to infer the properties of internal processes.

Newell (1973) described a series of production system models of the Sternberg memory task. Production systems are described in more detail later in Chapter 5. For the time being, a production system is essentially a set of condition-action pairs that scan a memory. When the contents of the memory match a production's condition, then it takes control of the memory and performs its action. Usually this action involves changing the contents of the memory, so that some other production's condition might be met.

Newell (1973) found that it was very easy to create fairly simple production system models of the Sternberg memory task. In fact, he describes seven different production system models written in a language called PSG. Each of these models was capable of making the correct response when given the probe. However, only one of these models generated response latency functions that resembled those of human subjects. Interestingly, this production system was not a model of search. Instead, it was a model of a general encoding and decoding scheme that could be used to perform the Sternberg task, as well as other basic tasks in cognitive psychology.

Given this result, it would have been quite reasonable for Newell (1973) to report only his last production system model. However, had he done so, he would have fallen victim to the relevant formalization problem. This is because one of his basic assumptions was that production systems described the functional architecture of human cognition. "In this view PSG represents the basic structure of the human information processing system. It follows that any program written in PSG should be a viable program for the human subject" (p. 494). As a result, in addition to coming up with one model that fits the human reaction time data, Newell must come up with a theory about why humans might use that production system, and not any of the other six, some of which are simpler. "Our example makes clear that multiple production systems are possible. Without a theory of which system is selected the total view remains essentially complete".

Newell (1973) went on to explore why an encoding model for performing the Sternberg memory task might be more adaptive than other possible production systems. He proposed that for the Sternberg task, short-term memory is unreliable, and an encoding model of memory processing is better at dealing with this unreliability. He also showed how an encoding strategy works well for a variety of other tasks, which is not the case for the simpler production system models that he was able to devise. However, Newell also identified plausible alternatives to the encoding model that are worthy of further exploration. In short, by avoiding the relevant formalization problem, Newell was able to develop a rich and detailed understanding of the Sternberg memory task that went far beyond what would be possible by only having a single, successful model that fit the data.

2.2.2.3 The Communication Problem

In formalizing a theory, a typical goal is to convert a set of informal verbal statements into a set of precise expressions that can be manipulated by some formal mechanism – mathematics, logic, or an algorithm. With this goal in mind, it is apparent that a theory will be more technical after formalization than it was before. This leads to another problem that must be faced: communicating the formalization to others, including those who might be interested in the domain, but not as interested in the technical details of the formalization.

Zeigler (1976) points out that the construction and testing phase of modeling can be quite exciting – often more exciting than recasting the model into a form for general distribution. As a result, "once the modeling challenge has been successfully overcome and the modeler's own curiosity satisfied, he may find it difficult to become enthusiastic about the task of clarifying it for

himself and communicating to others what he has accomplished" (p. 7). But clarification and communication are both required if the model is to have any impact.

Zeigler (1976) proposes that the effective communication of a model involves the following aspects. First, the researcher must generate an informal description of the model and its underlying goals and assumptions. Second, the researcher must provide a formal description of the model, including a presentation of the program used if the model is a simulation. Third, the researcher should present the tests of the model, including results and analysis. Fourth, the researcher should generate some conclusions about the model's range of application, validity, and cost. Finally, the researcher should relate his or her current model to both past and future models.

Zeigler (1976) notes that when the model is communicated, two different audiences must be kept in mind. One audience is the set of potential users of the model or its variations. The other audience is composed of "people who may not use the program or model directly but may make other uses of it in relation to their own research and development – call them the *colleagues*" (p. 8). With these two different audiences in mind, Zeigler suggests that the "informal description of the model is the most natural and effective way of establishing contact with the reader's intuition and of interfacing your world model with his world model" (p. 9). However, it is important to realize that with the audience of colleagues, this informal account might be the only way that contact is made. They may not be interested in paying the necessary attention to the more formal descriptions of the model, because they are an audience that isn't interested in using it.

2.2.3 Exploration Of Complex Domains

We have already seen that one advantage of modeling is the rigorous specification of theory. A second advantage is that models permit the exploration of complex ideas. "Simulations can be of value in this way either because a seemingly attractive idea might otherwise be too unconstrained to support predictions and tests or because a complex model may resist analytic exploration" (Lewandowsky, 1993, p. 237). Let us briefly explore each of these ideas.

2.2.3.1 The Economy Of Models

In mathematical psychology, as we will see in Chapter 4, one usually attempts to define a relationship between one set of variables and another. Within this framework, it sometimes is the case that there are a great many variables to be explored. Each of these variables can take on one of many different numerical values. The problem for a mathematical psychologist is to explore the set of possible settings for the variables in order to determine the best possible model. Mathematical psychologists have realized that the fastest, most economical approach to exploring the parameter space for a model is to use computer simulations (Estes, 1975; Luce, 1989, 1997, 1999).

The economy of modeling provides advantages for scientists who have little direct interest in mathematical psychology. Many are interested in studying systems that are highly complex, and that are also very difficult and expensive to examine experimentally. For example, neuroscientists who study the nervous systems of animals have to face the combined expenses of maintaining animals, of providing resources for drug or surgical treatments, and of histological examination of manipulated nervous systems – not to mention the ethical expenses of sacrificing animals for the advancement of knowledge. When a neuroscience experiment is performed, it would be very valuable to have a strong sense beforehand that the experiment is going to work, and is also going to provide important information. This kind of research is simply too expensive for "fishing" for interesting results.

One approach for increasing the likelihood that an experiment is going to be successful is to use computer simulation techniques to identify key issues, or predict the likely outcomes of

experiments. The simulation is itself much less expensive to run, and can be easily used to simulate a variety of experiments. One can use the simulation to "fish" for interesting results in a fashion that is far faster and cheaper than by actually performing the experiments on animals. Once an interesting set of predictions has been identified using the computer simulation, the result can be verified by actually performing the experiment on animals. The expectation is that the experiment should be successful because of all of the simulation work that was carried out beforehand. The results of the experiment can then be used to refine the computer simulation, so that it reflects an advancing state of knowledge, and so that it can be used to predict more sophisticated results in the future.

One excellent example of exploiting the economy of modeling is found in the research of neuroscientist Gary Lynch and his colleagues (e.g., Lynch, 1986). Lynch is primarily concerned with understanding the neural mechanisms underlying memory, and uses the olfactory system of the rat as his primary research focus. Lynch's research has uncovered many precise details about the neural circuitry that permits rats to remember and process information about different smells. A great deal of this information has been the result of experiments on rat brains. However, computer simulation has also been a central tool in Lynch's research program.

For instance, Granger, Ambros-Ingerson, and Lynch (1989) developed a computer simulation of olfactory cortex. The simulation consisted of 100 input cells (simulating axons of the lateral olfactory tract) randomly and sparsely connected to up to 500 cells in the olfactory cortex. Processing units in the simulation have a number of mathematical properties that model such characteristics as synaptic conductance, dendritic summation, excitatory and inhibitory signal characteristics, spike generation, and the speed of axon transmission. Depending upon the kinds of pulses transmitted to the network, it can learn by modifying the pattern of connectivity between its processing units. Granger et al. found that after learning a set of distinct groups of odors, the simulation's initial response to a cue odor only indicated the category to which it belonged. Subsequent responses to the same stimulus successively subdivided the category into increasingly specific encodings of the original cue. In other words, the model was demonstrating its ability to organize olfactory memories at a number of different levels of detail.

Importantly, the simulation created by Granger et al. (1989) led to at least five different predictions that were specific, and which were also not intuitively obvious. For example, in the simulation only a small number of cells responded to a specific input. As well, different cells responded when the simulation was presented different "sniffs", with the patterns of which cells were firing reflecting similarities and differences among odor cues. It is these sorts of specific, surprising predictions made by the model that can be selected as likely candidates for empirical study in animal systems. In the Lynch lab, there is a constant back-and-forth exchange of information between simulations and experiments, with each information exchange resulting in a more and more detailed understanding of the neural circuitry.

2.2.3.2 Beyond Mathematical Boundaries

In many disciplines there can be a marked competition between theorists and experimentalists. In physics, Lederman (1993, p. 13) observes, "In the eternal love-hate relation between theory and experiment, there is a kind of scorekeeping. How many important discoveries were predicted by theory? How many were complete surprises?" The tension between theory and experiment is also a frequently observed characteristic of psychology (Kukla, 1989; Paivio, 1986, Chaps 1-2).

One reason for this tension is that it is possible for theorists to make predictions about observations that take years for experimentalists to confirm. Many examples of this can be found in physics (e.g., Bodanis, 2000). For example, Einstein's general theory of relativity was first publicized in 1915. One of its major predictions, of the curvature of space, could not be empirically confirmed until observations of star positions during total solar eclipses were made in 1919 and 1922. In the 1930s, Chandrasekhar used special relativity theory to predict that white

dwarf stars could only exist up to a certain mass. He proved that if a star were larger than this limit, then it would ultimately collapse into a denser object (a neutron star or a black hole). This theory was extremely controversial when it was originally proposed, and was not empirically supported until observations in the 1960s that discovered pulsars, and which later demonstrated that pulsars were rotating neutron stars.

In these examples from physics, formal theories anticipated experimental results by years or decades. With the advent of computer simulation techniques, however, it is now possible to experimentally study models of systems whose complexity cannot yet be captured by mathematical formalisms.

In a wide variety of fields, researchers are interested in the properties of systems that have a large number of (often simple) components. Frequently, one component can influence the behavior of neighboring components in a manner that can only be captured by nonlinear equations. Furthermore, the behavior of one component's neighbors can influence the behavior of that component via feedback. In spite of the fact that these systems do not have any component that serves as a central controller, they often exhibit interesting, emergent, and systematic regularities. Examples of such systems include slime molds, insect colonies, and biological neural networks, to name a few. A new discipline, called complexity theory, is concerned with studying the properties shared by these diverse systems (Holland, 1998; Johnson, 2001; Waldrop, 1992).

The many nonlinear interactions in a distributed system like an ant colony or a brain make it very difficult to summarize the behavior of the system as a whole mathematically. However, it is possible to program a computer to simulate the interactions between system components. This means that the system can be studied, and understood, by making empirical observations about the behavior of the computer simulation even in the absence of formal theory. The fields of artificial life, genetic algorithms, artificial neural networks, and synthetic psychology all depend crucially upon the fact that one can use computers to explore regularities in domains that are currently too complicated to describe in formal equations.

2.2.4 Problems With Exploring Complex Domains

From the preceding section, it is clear that models provide a medium that provides many advantages for researchers interested in exploring complicated ideas in an efficient, inexpensive manner. These ideas can even be explored in advance of any mathematical account of the domain. However, while the ability to explore complex domains is a definite advantage of modeling, it can lead to some interesting disadvantages. Two of these are considered in the subsections below.

2.2.4.1 Bonini's Paradox

Dutton and Starbuck (1971) used the name *Bonini's paradox* to identify one problem with computer simulations of complex phenomena. Bonini's paradox, named after Stanford business professor Charles Bonini, occurs when a computer simulation is at least as difficult to understand as the phenomenon that it was supposed to illuminate. "The computer simulation researcher needs to be particularly watchful of the complexity dilemma. If he hopes to understand complex behavior, he must construct complex models, but the more complex the model, the harder it is to understand. ... As more than one user has realized while sadly contemplating his convoluted handiwork, he can easily construct a computer model that is more complicated than the real thing. Since science is to make things simpler, such results can be demoralizing as well as self-defeating" (Dutton & Briggs, 1971, p. 103).

While any model may fall into this trap, Bonini's paradox is particularly relevant for researchers who use connectionist networks. Connectionist models are introduced in more detail later in this book, and are essentially brain-like networks of simple nonlinear processors that can

learn to solve complex pattern recognition problems. Connectionist researchers freely admit that in many cases it is extremely difficult to determine how their networks accomplish the tasks that they have been taught. “If the purpose of simulation modeling is to clarify existing theoretical constructs, connectionism looks like exactly the wrong way to go. Connectionist models do not clarify theoretical ideas, they obscure them” (Seidenberg, 1993, p. 229).

Connectionist networks can fall prey to Bonini’s paradox for several reasons. First, because connectionist models are usually taught by example, they do not require a researcher to come up with detailed theory of how to perform a pattern recognition task prior to creating the model. In other words, connectionist networks allow “for the possibility of constructing intelligence without first understanding it” (Hillis, 1988, p. 176). Second, one can train connectionist networks that are extremely large; their sheer size and complexity makes it difficult to understand their internal workings. For example, Seidenberg and McClelland’s (1989) network for computing a mapping between graphemic and phonemic word representations uses 400 input units, up to 400 hidden units, and 460 output units. Determining how such a large network works is an intimidating task. This is particularly true because in many PDP networks, it is very difficult to consider the role that one processing unit plays independent from the role of the other processing units to which it is connected (see also Farah, 1994).

Difficulties in understanding how a particular connectionist network accomplishes the task that it has been trained to perform has raised serious doubts about the ability of connectionists to provide fruitful theories about cognitive processing. McCloskey (1991) warns “connectionist networks should not be viewed as theories of human cognitive functions, or as simulations of theories, or even as demonstrations of specific theoretical points” (p. 387). In a nutshell, this dismissal was based largely on the view that connectionist networks are generally uninterpretable (see also Dawson & Shamanski, 1994). It is clear that the success of connectionist networks, or of any other type of model, to contribute to psychological theory, depends heavily upon a researcher’s ability to avoid Bonini’s paradox. Later in this book we will see several examples of how this can be accomplished.

2.2.4.2 The Validation Problem

In Chapters 3 and 4, we will see that two common modeling approaches in psychology are models of data and mathematical modeling. Both use mathematical equations to describe and predict behavioral regularities. The equations represent a theoretical statement about behavior. The validity of the theoretical statement is usually assessed using “goodness of fit”: the equation makes certain predictions about what behavior should be observed in experimental subjects. The validity of the theory depends upon the extent that the predictions are consistent with these empirical observations.

However, the fact that new modeling techniques such as computer simulation permit the study of systems that cannot be formally described had led to a situation in which this traditional notion of theory validation does not work very well. Mathematical psychologists, for example, are deeply disturbed by the fact that it is very difficult to formulate a procedure for measuring the validity of computer simulations (Estes, 1975; Luce, 1999).

This problem is compounded by the bottom-up strategies used in the simulations that are of concern to complexity theorists. In many instances, these simulations involve defining the interactions between neighboring components in the model, without being concerned with the overall outcome of the simulation. In other words, rather than modeling a particular phenomenon (which we will see is the typical top-down strategy used to create models of data and to propose mathematical models), complexity theorists are interested in discovering what surprising properties emerge from the interactions of known components. In many cases, they may have no idea what kinds of regularities will emerge from their simulation.

This makes it particularly difficult to validate a complexity theorist's simulation, because it may not even be known *a priori* what the model is a model of. This is one of the reasons that many of these simulations are viewed skeptically. For instance, these models have been described as being "fact free science" by evolutionary biologist John Maynard Smith (Mackenzie, 2002). Some have argued that it is impossible to verify or validate these kinds of simulations (Oreskes, Shrader-Frechette, & Belitz, 1994). "Like a novel, a model may be convincing – it may ring true if it is consistent with our experience of the natural world. But just as we may wonder how much the characters in a novel are drawn from real life and how much is artifice, we might ask the same of a model: How much is based on observation and measurement of accessible phenomena, how much is based on informed judgment, and how much is convenience?"

Validating a model is a difficult problem that is a central concern of psychology and cognitive science (Fodor, 1968; Pylyshyn, 1980, 1984). For the time being, let us simply be aware that this problem exists. In several of the later chapters we will have an opportunity to consider how synthetic psychologists approach this problem.

2.2.5 Serendipity

We have already covered two of the main advantages of models: the rigorous specification of theory and the ability to explore complicated domains. There is one further advantage to be considered – the ability of a model to reveal serendipitous discoveries. Lewandowsky (1993) is concerned by the fact that "a widespread opinion among critics is that theories or simulations somehow stand in the way of serendipitous discovery" (p. 238). He goes on to point out the flaws in this view.

In the next three chapters of this book, the notion of serendipity will be important in distinguishing different kinds of models. In particular, I will be arguing that some kinds of models (models of data, mathematical models) provide less opportunity to surprise a researcher than do others (computer simulations). However, as a prelude to that more detailed discussion, let us briefly consider some general aspects of how models can lead to surprises.

2.2.5.1 Emergence And Surprise

One of the reasons that some researchers believe that models cannot generate surprises is because systems like computer simulations are deterministic. If a computer can only follow its program, then it stands to reason that it should be impossible for the program to surprise the programmer (Haugeland, 1985).

The difficulty with this logic is that it assumes that the purpose of the programmer is to create a program that is responsible for carrying out some overall, holistic, behavior. However, sometimes this is not the programmer's goal. Indeed, in many situations the programmer is concerned with programming simple and well-defined local interactions between the components of a system. "Local turns out to be the key term in understanding the power of swarm logic. We see emergent behavior in systems like ant colonies when the individual agents in the system pay attention to their immediate neighbors rather than wait for orders from above. They think locally and act locally, but their collective action produces global behavior" (Johnson, 2001, p. 74).

In many situations, the programmer will have complete understanding of the programmed local interactions, but will be unable to predict the global behavior that the local interactions produce. It is these emergent properties that are surprising, and which are capable of providing new insights.

2.2.5.2 An Example: Banding In Value Units

One example of a serendipitous result from a model comes from my own laboratory's research on connectionist networks. As we will see in more detail later in this book, a

connectionist model is a network of simple processors that send numerical signals to one another. One of the basic tasks of any processing unit in this kind of network is to add up the total incoming signal, and to convert it into an internal level of activity. Mathematically, this is done using an equation called an activation function.

By 1989, Don Schopflocher and I had developed a method of training connectionist networks that used a different activation function than is found in typical connectionist networks (Dawson, 1990; Dawson & Schopflocher, 1992). We called our architecture networks of value units, using terminology borrowed from Ballard (1986), because the activation function tuned the processor so that it had a strong response to a narrow range of incoming signal, and had a very weak response when the incoming signal was too strong or too weak to fall in this narrow range (for more details, see Chapters 10 and 11).

After this architecture had been published, we continued to study it because it had several advantages that we wanted to exploit. However, one problem that we were concerned about was Bonini's paradox: the networks that we trained had an internal structure that was very difficult to understand. We expended a great deal of fruitless effort trying to develop techniques for figuring out the "program" that was encoded in the connection weights of our networks.

In the winter of 1993, we literally stumbled upon an emergent property of the value unit architecture that aided network interpretation immeasurably. One of my philosophy graduate students, Istvan Berkeley, had trained a network of value units to solve a logic problem developed by Bechtel and Abrahamsen (1991). He had devoted hundreds of hours to examining the structure of this particular network. One kind of data that we collected in this process was analogous to "wiretapping" of neurons by neuroscientists: we simply recorded the activity of each processor within the network to each stimulus that the network was presented.

In an effort to help interpret the network, Don Schopflocher took a copy of the "wiretapping" data, and attempted some multivariate analyses. This didn't provide any breakthroughs. However, Don did notice that in the data a lot of numbers were repeated. He didn't make anything of this, and neither did I. In fact, I pretty much ignored this observation. Importantly, the very next day, Istvan – who had been looking at the very same data – came to me and repeated, almost word for word, Don's observation. Being told the same thing twice finally captured my attention, and I took the data and started to perform some graphical analyses.

In very short order, I had selected a particular type of graph called a jittered density plot. One such graph can be drawn for each one of our processing units. In a jittered density plot, each dot in the graph represents the unit's response to one stimulus pattern. The x-position of the dot indicates the actual level of unit activity. The y-position of the dot is randomly selected, and is used to try and prevent dots from overlapping each other as much as possible.

Now, for a standard processing unit, a jittered density plot is not very informative, because it is not very structured. Usually it is just a smear of dots throughout the whole graph. Our serendipitous finding was that the jittered density plots for value units were much more structured. Rather than being an uninformative smear, as in the example above, we found that the plots for the processors in Istvan's network were organized into tight bands, usually with a great deal of space separating one band from another.

We were tremendously excited and surprised by this result, and our excitement grew and grew as each new jittered density plot came out of the printer. A whole new set of questions jumped to mind. Why did the bands emerge? Was there anything in common among the subset of patterns that fell into one band? In answering these questions, we discovered that the bands provided a method for identifying the kinds of features that were being detected by each unit in the network. We were then able to use these features to determine how the network was solving the logic problem, and to make an argument that connectionist networks might be more symbolic

than was traditionally thought (Berkeley, Dawson, Medler, Schopflocher, & Hornsby, 1995; Dawson, Medler, & Berkeley, 1997).

More recently, we have developed a much stronger formal understanding of why banding occurs, and have used it to predict and discover banding for other problems and for other architectures (McCaughan, Medler, & Dawson, 1999). We have also developed more sophisticated interpretation techniques than the purely local ones that we reported in 1995 (Dawson, Boechler, & Valsangkar-Smyth, 2000; Dawson, Medler, McCaughan, Willson, & Carbonaro, 2000; Dawson & Piercey, 2001; Medler, McCaughan, Dawson & Willson, 1999). However, all of these advances have depended upon our original lucky discovery. Don Schopflocher and I had no idea that we were going to produce this result when we developed our learning rule in 1989. Indeed, we were using this algorithm for approximately 4 years – and encountering numerous dead ends in network interpretation – before we chanced upon this discovery.

2.2.6 Luck: Good And Bad

For the other two advantages of modeling, the rigorous specification of theory and the ability to explore complex phenomena, we have outlined accompanying disadvantages. What possible disadvantages might one find with an approach that permits serendipitous discovery? The subsections below briefly consider three different kinds of concerns.

2.2.6.1 Is Good Luck Bad Science?

One concern that is often raised when serendipity is a key component of a research program is that the program doesn't seem to be very scientific. The traditional view of science is that it is a careful, gradual, goal-directed advancement of knowledge, in which current information is used to generate and test new hypotheses. Hypotheses "are the first rungs of the ladder of science, becoming theories as the harder factual sides of the ladder are extended, and finally facts when the ladder makes firm contact with structures established by other ladders of hypothesis" (Hocking, 1963, p. 3).

However, "science seldom proceeds in the straightforward logical manner imagined by outsiders. Instead, its steps forward (and sometimes backward) are often very human events in which personalities and cultural traditions play major roles" (Watson, 1968, p. ix). Put another way, "the discoveries of penicillin, X-rays, and America have apparently failed to alert students of memory to the possibility of serendipitous findings within their own field" (Watkins, 1990, p. 333).

Nevertheless, there is still some sense that if the advancement of one's research field depends overtly on serendipity, then this reflects a weakened dependence on theory or on prior knowledge. This simply isn't so. In very general terms, we will see that advances in synthetic psychology come about by taking a set of components, by letting them interact, and by observing surprising emergent phenomena. However, the role of theory and prior knowledge in this endeavor is still fundamentally important, because it guides decisions about what components to select, and about the possible dynamics of their interaction. In the words of Benjamin Franklin, diligence is the mother of good luck.

2.2.6.2 Good Luck, Bad Control

We will see later that one of the modern arguments in favor of adopting a synthetic approach to modeling, rather than analyzing a system into its components, is the opportunity for generating simpler theories. "Analysis is more difficult than invention in the sense in which, generally, induction takes more time to perform than deduction: in induction one has to search for the way, whereas in deduction one follows a straightforward path. A psychological consequence of this is the following: when we analyze a mechanism, we tend to overestimate its complexity" (Braitenberg, 1984).

However, if many of the advances of synthetic psychology are going to depend upon emergent surprises, then this view tells only half the story. There are many solid theoretical and empirical arguments that make the point that analytic approaches are difficult, and lead to overly complicated theories. However, a synthetic approach may be no less difficult. The tacit view of proponents of the synthetic approach, like Braitenberg, is that if one can build a system, then one must be able to understand it. However, we have already seen that this view is not completely correct. The idea that models can lead to serendipitous results comes from the situation in which a modeler has a very precise understanding of a system at one level (i.e., the level of the components), but has little understanding of the system at another level (i.e., a higher level at which emergent surprises can be seen).

In other words, modelers in synthetic psychology are likely going to be in a situation in which they have a high degree of control of their systems at a microlevel, but have much less control of their systems at a macrolevel. Furthermore, they may have little understanding about how microlevel processes result in macrolevel behaviors. We will see later in this book that the only way to deal with this problem is to combine synthetic and analytic approaches. After one discovers an emergent surprise in a synthetic model, a good deal of effort is going to be required to analyze the model in order to account for how the surprise emerged. Finding lucky surprises will not suffice. Synthetic psychology is charged with explaining the surprises too.

2.2.6.3 Going Beyond The Model

One final concern with the serendipity of modeling is that it requires a researcher to go beyond the direct intent of his or her model. This is a problem because this requires the researcher to move against a tradition that is a strong, tacit component of experimental psychology, as we will see in the next two chapters. When many psychologists think of modeling, their view is that the purpose of a model is to fit or mimic experimental data. The reason for this belief is that it is central to two types of models that have a long history in psychology, models of data and mathematical models. In general, if a model of data or a mathematical model does not fit the data, then the model is abandoned.

The possibility of discovering new and surprising characteristics of a model requires that this very narrow view of what a model is intended to do, or of how a model should be evaluated, must be either abandoned or suspended. This is because the only way that a model can surprise is if one examines how it deals with situations that it was not originally intended to face. Once my students have developed a model of some phenomena, I always ask them to find out what they can “get from the model for free”. My request is an attempt to encourage them to determine whether their model has any interesting or surprising emergent properties that they may not have considered. I also tell them that if a model doesn’t have any surprises, then it may not be a very good model. My own experience is that this is true – but to be aware of this truth, one must abandon the notion that the only purpose of a model is to fit data that has already been collected from subjects!