Chapter 1

Modelling Cognition

Richard P. Cooper and John Fox

Overview: This chapter introduces the basic concepts of cognitive modelling. The historical context is briefly reviewed and some ways that cognitive modelling may be used in theory development are described. Pros and cons of the enterprise are then discussed. This is followed by a detailed description of several major approaches to modelling (symbolic, connectionist, hybrid, and architectural). The chapter concludes with some remarks on our own view of the field, and on the role of COGENT, the modelling environment used throughout the rest of this book.

1.1 What is Cognitive Modelling?

This book is about understanding human cognition — the mental processes involved in thought, reasoning, language, and so on. The basic premise on which it is founded is that the development of computer models of cognitive processes can further our understanding of those processes by allowing us to evaluate computational mechanisms that might underlie behaviour.

A model in any field, whether it be engineering, architecture, molecular biology, or cognitive science, is a representation of something that may be used in place of the real thing. Traditionally the representation might be made out of wax, clay, metal, or wood. Thus an architect may produce a wooden model of a building in order to demonstrate or evaluate the building's appearance. Similarly an engineer may produce models of bridges to evaluate the relative strengths of competing designs.

A computer model has the same function as a traditional model, but rather than being made out of clay or wood, it consists of a representation in some precisely-specified computer language. Like the traditional wood or clay model, the representation abstracts away unimportant features or characteristics of the thing being modelled, but it retains all that is essential. Thus, a computer model of an aeroplane wing for use in testing aerodynamics might abstract the colour and weight of the wing, but would retain the important features of size, shape, and curvature, for these features are critical to the aerodynamic behaviour of the wing.

A computer model in cognitive science is much the same kind of thing — it is an abstract representation that may be used in place of the real thing. The difference in cognitive science is that the thing being modelled is a cognitive process. So, cognitive modelling is the development of computer models of cognitive processes, and the use of such models to simulate or predict human behaviour.

1.2 A Sample Model

It is perhaps useful to begin by considering, for illustration purposes, what a cognitive model might look like. Consider the task confronting a doctor when attempting to make a diagnosis. This is a typical example of the kind of high-level cognitive process which we are concerned with in this text. The doctor's task is roughly as follows: a patient arrives at the doctor's office complaining of some symptom (the presenting symptom). The doctor then reviews the patient's medical history, takes some measurements, and asks the patient for more information about his/her symptoms. On the basis of the information collected, the doctor may ask yet more questions or carry out further tests. Finally, the doctor decides what he/she thinks is causing the patient's symptoms, and makes a diagnosis.

A cognitive model of this task might consist of a computer program that could take as input some representation of the stimulus (e.g., the patient's medical history and presenting symptom) and produce as output a representation of the doctor's response (whether to seek further information by querying another symptom or performing a test, or whether to make a diagnosis). A simple model might consist of a series of stages, whereby the medical history and presenting symptom initially lead the doctor to propose several competing hypotheses (e.g., either the patient has asthma or bronchitis). A second stage might then involve selecting the hypothesis corresponding to the disease that is most frequent in people who share the patient's medical history (e.g., that the patient has asthma). The symptom profile of this common disease might then be recalled. The next stage may involve querying the patient about further symptoms associated with the hypothesised disease, or about the precise nature of the presenting symptom.

In a fully fledged computational model each of the above processes would form part of the computer program, and the doctor's expected behaviour could be simulated by running or executing the computer program. If the doctor's actual behaviour differs from that which was predicted by the model, we know that the model is inadequate, and we can examine ways of addressing the model's shortcomings. If the model is accurate, and especially if it is accurate across a range of situations, then we may have some confidence that we understand the cognitive processes underlying performance in the diagnosis task.

This sample model is an abstract representation of the hypothesised cognitive processes of the doctor. The model is a representation of the processes because elements in the model correspond to elements of the hypothesised processes. It is abstract because it leaves out much of the detail of the actual cognitive processing (e.g., the cognitive processing is carried out by neural tissue rather than a computer chip). It is hoped that this detail is not important to the overall behaviour of the doctor. Lastly, if the model is accurate it may be used in place of a doctor, to predict how a doctor's diagnostic behaviour might change over time for example, or to evaluate strategies to lessen the chances of mis-diagnosis.

1.3 What Makes a Good Model?

A good model has two critical properties:

1. It is *complete*, to the extent that the model does not abstract out aspects of the original that have an important influence on the properties or behaviour of the original (e.g., a model of fluid flow in a pipe should not ignore friction between the pipe wall and the fluid); and
2. It is *faithful*, to the extent that the abstraction process does not introduce component properties or relationships that are not features of the original (e.g., a model building made out of children's clay might suggest that a real building is malleable).

These properties have their origin in the way a model abstracts from the thing being modelled. Completeness is about not abstracting details that are important. Faithfulness is about not introducing confounding details during the abstraction process.

It is important to realise that neither of the above properties is absolute, in the sense that a model is always a model for a specific purpose. The colour scheme of a model aeroplane, for example, is more relevant than the minutiae of its aerodynamics if the model is intended to help identify planes from different countries. On the other hand the reverse is true if the model is intended to allow exploration of aerodynamic design changes. For a model to be useful for a specific purpose it must be sufficiently complete and faithful that the model builder can correctly derive or deduce from the model properties of the real object which were previously unknown (e.g., whether a building will be functional or aesthetically pleasing).

1.4 The Rise of Cognitive Modelling

The modern era of empirical/cognitive psychology is often dated to the work of Ebbinghaus (1885), who set himself tasks such as learning lists of nonsense words in order to study the processes underlying memory. Since that time, the study of cognitive processes has gone through several shifts in emphasis and approach.

Early researchers used introspection as their primary empirical technique. At the beginning of the twentieth century introspection was criticised as being subjective and hence non-scientific. The rejection of introspection was accompanied by the rise of the Behaviourist School, which dominated psychology in the English speaking world for much of the first half of the twentieth century. Behaviourists believed that internal mental states could not be studied in an objective manner. They avoided all talk of mental states, and attempted to account for all behaviour in terms of simple stimulus-response links.

Around the middle of the twentieth century it was demonstrated conclusively that, at least for some higher mental processes such as language and skilled behaviour, simple stimulus-response links could not explain the full range of behaviours of which most humans are capable. It was shown that stimulus-response links were necessarily mediated by internal mental states, and hence that internal mental states were essential for causal explanations of these cognitive processes. Behaviourism was thus supplanted by a new psychology. The picture of cognition that arose in this new psychology was one in which the mind was an "information processor" and cognition was "information processing".

Within information processing psychology, sensory processes (such as vision and hearing) act as input devices, converting information from the surrounding environment into some internal form or representation. Mental processes manipulate and transform these representations, often triggering responses via output processes. This view of cognition has received support from fifty years of careful empirical work and remains current. The last half of the twentieth century, however, witnessed two major changes in approach. First, computer simulation techniques were adopted in order to explore and develop complex theories of cognitive processing and to evaluate competing theoretical accounts of empirical phenomena. The use of these techniques is one of the distinguishing features of the discipline of cognitive science. Second, brain imaging techniques were developed in order to localise cognitive processing and relate the functioning of the mind to the functioning of the brain. This relation is the primary focus of the newly emerged discipline of cognitive neuroscience.

Computer simulation techniques have allowed cognitive models such as that described in the previous section to be developed. To illustrate further consider the list-learning experiments of Ebbinghaus (1885). A cognitive model of list-learning would detail the mechanisms or processes by which the stimuli (the elements of the list) are stored and recalled, and any possible intervening mechanisms or processes involved in information consolidation (e.g., rehearsing the list) or information loss (e.g., through forgetting elements of the list). Such a model is considered in Chapter 2.

1.5 Modelling and Simulation

In many scientific domains modelling provides a way of investigating the rules or laws that govern complex systems that are only partly understood. This is normally achieved through building a model that we think will have similar characteristics to the real system and studying the characteristics of the model. Modelling within cognitive science follows this basic approach, with *simulation* being the principal method of studying a model's characteristics.

A common example of simulation is seen in modern weather forecasting. This involves determining relevant characteristics of the atmosphere, such as temperature, air pressure, and wind speed, across a grid of points, and then using mathematical equations that describe how the characteristics change with time to predict the values for the characteristics a short time later. By applying the mathematical equations over and over, meteorologists simulate the weather and are able to predict characteristics of the atmosphere at some later time. The simulation does not involve real wind currents or real temperatures. Instead, elements of the model (e.g., arrays of numbers) correspond to characteristics of the object of simulation.

The use of simulation in cognitive modelling parallels that in meteorology. A cognitive model specifies a number of processes, the initial characteristics of those processes, and the way in which those characteristics change through interactions with other processes within the model. Simulation involves repeatedly working through all of the interactions to determine how the characteristics of the system change over an extended time period.

Simulation is particularly useful when it is difficult to understand the behaviour of the system being modelled. Any system that is made up of many interacting components can be difficult to understand. If, in addition, some or all of those components have properties which make them individually difficult to understand, then understanding of the complete system is likely to be compromised further. This can happen when the components are:

- heterogeneous (i.e., there are many qualitatively different kinds of component within the system, each of which is idiosyncratic so their behaviour cannot be summarised with some uniform function);
- non-linear, stochastic and/or asynchronous in their response functions (i.e., components produce outputs that cannot be simply extrapolated from previous outputs, and they produce such outputs at times determined by the components themselves, rather than by some external clock);

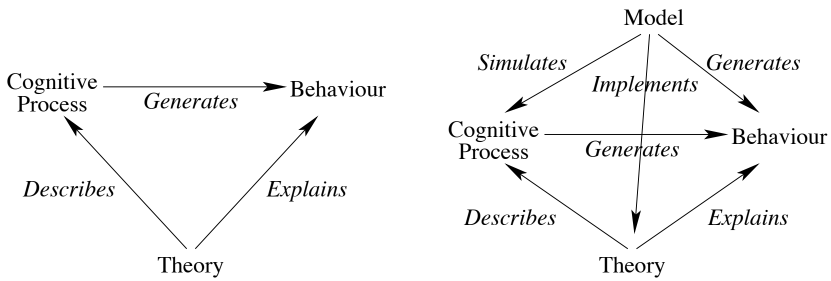

Figure 1.1: Some relations between behaviour, a cognitive process underlying that behaviour, a theory of the process, and a model of the process. The left panel shows the classical relations. The right panel shows the added dimension provided by modelling.

- densely connected to other components (i.e., there are many interactions between components); and
- recursively defined (i.e., components are themselves defined in terms of sub-components with these properties).

Unfortunately many psychological theories posit interacting processes or components that share many or all of these properties. Psychological theories are thus often highly complex. Simulation provides an effective means of determining the consequences or predictions of such interacting processing systems.

Indeed, most aspects of cognitive processing have formed the target of one or more cognitive models. Thus there are cognitive models of perceptual processes (such as those involved in analysing a visual image), attentional processes, motor and action control processes, memory, problem solving, reasoning, categorisation, and so on. In most cases, the techniques of simulation have led to greater understanding of the underlying cognitive processes.

1.6 The Role of Cognitive Modelling

Standard theorising in cognitive psychology is concerned with three types of entity: behaviours, cognitive processes underlying behaviours, and theories of those cognitive processes. Figure 1.1 (left) illustrates some of the relations between these three types of entity. The "classical" view (without modelling) is that a theory explains a behaviour by describing the cognitive processes that generate that behaviour. Modelling adds an extra dimension to this picture. A model generates behaviour, implements a theory, and simulates a cognitive process (as shown in the right panel).

Each object and relation in Figure 1.1 could form the focus of substantial discussion. For present purposes we will focus on just one triangle from the figure: that involving theory, behaviour (or empirical phenomena), and a model that implements the theory and generates the behaviour.

A theory in cognitive psychology typically takes the form of a series of related statements or assumptions, perhaps supplemented with a diagram. As in most sciences a theory is generally intended to "explain" some known empirical phenomena, and make predictions about other phenomena. The explanation consists of an argument demonstrating that the known empirical phenomena are consequences of the assumptions, and the predictions are additional assertions that may be tested through experiment. If the predictions are met then the theory gains further support. If the predictions are not met the theory is falsified and must be modified or abandoned. This is not a bad thing, for it means that we have learnt something: that one or more of the theory's assumptions is false.

This simplified picture is somewhat idealised. In most cases theories within cognitive psychology are under-specified in that the assumptions are incomplete or imprecise. Correspondingly, the relation between assumptions and data (including both prior data and predicted data) is generally qualitative rather than quantitative. Cognitive modelling helps to address these issues, and hence serves an important role in cognitive scientific explanation.

Few theories within cognitive psychology are stated in precise and unambiguous terms. Modelling forces theoretical precision by requiring that a theory be computationally complete. That is, in order to construct a model it is necessary to specify all aspects of a theory in the kind of detail required by a computer program. This means that no details can be left out, and that no aspects of a model may be vague, ambiguous, or open to alternative interpretations. Thus, although a model may abstract away some details of cognitive processing, it must still be "computationally complete".

As discussed below, the requirement for computational completeness is not without a cost, but it also has a subsidiary benefit — that of supporting clear, precise communication. A computational model can be an effective way of expressing and communicating a theory in objective terms. Verbal or diagrammatic theory specifications are generally open to interpretation. Models presented in clear publicly-specified theoretically-neutral computer languages do not suffer from this difficulty. In certain cases they also lend themselves to formal analysis of their properties, allowing theoreticians to derive logically necessary consequences from the theoretical assumptions without even running the computer model.

1.7 Further Benefits of Cognitive Modelling

Theories of cognitive processing are frequently complex. As noted above, it can therefore be difficult to accurately determine the behaviour of a theory by reasoning directly from its assumptions. This is especially true for theories that posit multiple heterogeneous sub-processes and when the explanations of empirical phenomena depend upon interactions between these sub-processes. A significant advantage of cognitive modelling is that it allows detailed evaluation of such theoretical proposals. Furthermore, once a model has been developed it is possible to investigate the impact of changes in theoretical assumptions on the model's behaviour. Modelling thus allows both evaluation and exploration of theories and their consequences.

A further benefit of cognitive modelling arises from its use as a supplement to cognitive neuropsychology. Cognitive neuropsychology is concerned with different patterns of behaviour that follow neurological damage, and the use of such patterns to inform theories of normal cognitive functioning. The relevance of cognitive modelling to cognitive neuropsychology lies in the fact that once one has developed (and evaluated) a model of normal cognitive functioning in some domain (e.g., language production), one can "damage" or lesion the model in some principled way and then compare the behaviour of the lesioned model with that of relevant neurological patients (e.g., patients with language production deficits). A successful model will be able to account for both normal and impaired performance.

The relation between modelling and cognitive neuropsychology goes both ways. While modelling can further our understanding of cognitive functioning and its breakdown, the breakdown of cognitive functioning can also provide a source of data against which models may be tested.

1.8 Some Objections to Cognitive Modelling

Much cognitive modelling is motivated by the benefits described above, but cognitive modelling is not without its difficulties. These difficulties primarily arise from the need to make detailed assumptions about representation and processing that are necessary for execution of the model. The detail of these assumptions means that they may be very difficult to justify empirically. Two arguments — that of the behaviourist and that of the cautious scientist — require special consideration.

As we have seen, the behaviourist approach to psychology held that psychological theories should deal only with observables of behaviour. They should not theorise about processes that intervene between the input (stimulus) and the output (response), because such processes cannot be observed. While behaviourism is no longer popular, the need to specify intervening processes in the kind of detail required by cognitive modelling may leave some with an uneasy feeling.

At present there are few cognitive scientists who dispute the existence of intervening processes. It is possible that neuroimaging techniques will one day provide direct evidence for the detail of these processes, but until that day cognitive scientists must adopt other means to provide support for their theorising about intervening processes. These means may include appeals to principles such as simplicity and parsimony. For example, given two theories that claim to account for the same behaviour, one may favour the theory with fewer assumptions or simpler intervening processes. Stronger support may be obtained, however, by using modelling to ensure that the theories are complete and fully consistent with behaviour. In other words, the response to the behaviourist's objection is that intervening processes have proven to be necessary in accounting for the complexity of cognitive functioning. Given that such processes are necessary, cognitive modelling allows us to examine the nature and consequences of hypothesised intervening processes in detail.

The cautious scientist is less sceptical about intervening processes than the behaviourist, but is still uncomfortable about making assumptions that are not fully justified on empirical grounds. To illustrate, a scientist studying memory for lists of words may be happy to postulate a process of memory decay, whereby words of the list gradually become unavailable for recollection. If this process is to be incorporated into a model, however, one must specify details of the process. Is decay a probabilistic process whereby words may spontaneously disappear from memory, or can memory representations vary in their strength, with decay affecting strength? In the former case, what function governs the probability of a word decaying at any specific moment? In the latter case, how does strength change with time, and is there some strength threshold below which words are inaccessible? The cautious scientist may argue that he or she has insufficient evidence to answer these questions, and hence prefer to settle for a less detailed description of the decay process. As a result, he or she will be inclined to balk at models that include specification that goes beyond that which can be justified on purely empirical (or even theoretical) grounds.

The cautious scientist's objection is a less extreme form of the behaviourist's objection. It is an objection to the computational completeness that modelling requires. Whether this is a limitation or disadvantage of cognitive modelling is unclear. It could be argued, for example, that computational completeness is a significant advantage, for it makes clear that the stance of the cautious scientist is incomplete. An additional benefit of cognitive modelling is therefore that it can make clear to us the limitations of our knowledge. Knowing what we don't know is an important stage in understanding.

Of course the counters to these objections do not absolve the modeller from a basic responsibility: to relate models to both theory and data. One of the great strengths of cognitive modelling is the way that it may complement the approaches of empirical and theoretical psychology.

1.9 Approaches to Cognitive Modelling

While those who practice cognitive modelling generally agree on the benefits of the enterprise, they often disagree about how the enterprise should be approached. There are several schools of cognitive modelling, and advocates of one are frequently critics of another. The schools differ in their assumptions about mental representation and their view on the relation between a cognitive model and the brain.

The connectionist school argues that properties of the neural tissues that implement the information processing mechanisms of the mind are critical to the way the mind works. As such, they build models that consist of many simple interacting units functioning in parallel. The units are typically understood as analogues of neurons or neural cell assemblies. Symbolic cognitive models, in contrast, generally make the assumption that information processing can be described in terms of the manipulation of symbolic representations (as defined below). Within the symbolic approach the neural substrate is viewed as an implementation medium that is of secondary importance.

The symbolic and connectionist approaches to modelling share little beyond the basic idea that the functioning of the mind is computational in nature and so can be simulated by a machine. The two approaches are frequently presented as disjoint and even in opposition to each other. However, both approaches have strengths and weaknesses. This has led to attempts to develop hybrid symbolic/ connectionist systems that combine the strengths and circumvent the weaknesses of the individual approaches.

Two further approaches to modelling are the architectural approach and the dynamical approach. The former involves adopting a hypothesised organisation of the complete set of information processing structures that comprise the mind/ brain, and using this to guide the development of models. The latter is more mathematical in emphasis. In its most extreme form it argues against the use of internal mental representations of the form used by any of the other approaches. The claim of the dynamical approach is that mental processing may be described by differential equations in much the same way as the trajectory of a comet may be described by differential equations, but, as in the case of the comet, mental processing does not involve solving equations. Rather, it involves responding to the mental equivalents of forces. The dynamical approach, with its rejection of mental representation, is not well suited to modelling high-level cognitive processes, and will not be discussed further in this book.

1.9.1 Symbolic Models

The symbolic approach to cognitive modelling developed from early work by Newell and colleagues (e.g., Newell, Shaw, & Simon, 1958; Ernst & Newell, 1969) on logical inference and human problem solving. After developing an explicit theory of problem solving, they specified the theory as a sequence of steps that could be performed by a computer program.

Symbolic Propositional Representations

An essential element of the early computational work of Newell and colleagues was the representation of information relevant to a problem in a symbolic, propositional form, and the manipulation by the program of this representation. Consider a simple descriptive statement such as:

the red pyramid is on the blue cube

This may be analysed as a conjunction of *propositions* concerning the objects (a pyramid and a cube), their properties (red and blue, respectively), and the relation between the two (that the pyramid is on the cube). The information may be represented formally as follows:

$$\mathtt{pyramid(p)} \ \& \ \mathtt{red(p)} \ \& \ \mathtt{cube(c)} \ \& \ \mathtt{blue(c)} \ \& \ \mathtt{on(p,c)}$$

Each conjunct of this representation is a symbolic proposition: a statement that consists of symbols (e.g., `p` or `red`) which refer to objects, properties, or relations, and that may be either true or false, depending on the state of the objects to which the proposition applies.

Symbolic propositional representations have two desirable properties: systematicity and compositionality (see Fodor & Pylyshyn, 1988). A representation is systematic if it consists of a number of parts and the result of replacing some of the parts with other parts of the same kind is also a meaningful representation. Thus, if `on(p, c1)` is a meaningful representation and `c1` and `c2` both refer to objects, then `on(p, c2)` will be a meaningful representation. It may not be true, but it will be meaningful. A representation is compositional if it consists of parts and the meaning of the whole is a function of the meaning of the parts. The representation `on(p, c1)` is compositional because it consists of parts (`on`, `p` and `c1`) and its meaning is a function of the meaning of those parts. Representations that are compositional and systematic may be manipulated by rules that are dependent only on the form of the representation and not on the meaning of the representation. It is this manipulation that is central to many symbolic cognitive models.

Symbolic representations may also be embedded to represent information of arbitrary complexity. Thus, the statement that:

Joe believes that the pyramid is green

may be represented by the compound proposition:

$$\texttt{pyramid(p)} \,\&\, \texttt{believes(joe, green(p))}$$

Similarly the statement that:

Joe believes the blue pyramid to be green

may be represented by the compound proposition:

$$\texttt{pyramid(p)} \,\&\, \texttt{blue(p)} \,\&\, \texttt{believes(joe, green(p))}$$

Note that the proposition `believes(joe, green(p))` may be true even if the embedded proposition `green(p)` is false.

Symbolic propositional representations provide a general means of representing information. Symbolic models adopt this general representational device, and supplement it with symbol manipulation rules that operate on representations to transform them or build new representations. A symbolic cognitive model is therefore a model of the mechanisms by which symbolic propositional representations are manipulated and transformed from one form to another.

A Simple Symbolic Model

To illustrate symbolic modelling consider the task of transitive inference (as investigated by, for example, DeSotto, London, & Handel, 1965; Clark, 1969). Subjects performing the task are given two statements concerning individuals and relations between them (such as *Anna is shorter than Beth* and *Caroline is taller than Beth*) and asked to either judge the truth of a third statement (e.g., *Is Anna shorter than Caroline*?) or to generate a true statement concerning the two unrelated individuals (if such a statement exists).

How might a symbolic model perform the transitive inference task? Such a model might first convert the given statements into propositional form. It could then apply rules of inference to the propositions in order to test or derive a conclusion. In the simplest case of deriving a conclusion, this might proceed as follows:

Given statements:
  Anna is taller than Beth
  Beth is taller than Caroline
Propositional encoding:
  `taller(anna, beth)`
  `taller(beth, caroline)`
Inference rule 1 (part of long-term knowledge):
  $\texttt{taller(X, Y)} \,\&\, \texttt{taller(Y, Z)} \Rightarrow \texttt{taller(X, Z)}$
Result of applying rule 1:
  `taller(anna, caroline)`

Verbal decoding:
Anna is taller than Caroline

In this, and all examples throughout this book, symbols beginning with a capital letter (e.g., `X`, `Y`, `Z`) denote variables, which may be mapped onto other symbols (e.g., the individuals `anna`, `beth` and `caroline`) in the application of an inference rule.

Only one inference rule is required for the above simple case. Suppose however that we are dealing with a more complex case, in which the given information involves both *taller* and *shorter*. For example:

Given statements:
Anna is shorter than Beth
Caroline is taller than Beth
Propositional encoding:
`shorter(anna, beth)`
`taller(caroline, beth)`
Inference rule 2 (long-term knowledge):
`shorter(X, Y)` ⇒ `taller(Y, X)`
Result of applying rule 2:
`taller(beth, anna)`
Inference rule 1 (long-term knowledge):
`taller(X, Y)` & `taller(Y, Z)` ⇒ `taller(X, Z)`
Result of applying rule 1:
`taller(caroline, anna)`
Verbal decoding:
Caroline is taller than Anna

This case requires one extra step: use of a second inference rule to transform the *shorter* relation into a *taller* relation. One might therefore predict from this simplest of models that the second case will take subjects longer than the first case. Indeed, this has been found to be the case (DeSotto *et al.*, 1965; Clark, 1969). The account therefore receives some empirical support. Furthermore, empirical evidence points to faster solution times when information is stated in terms of *taller* as opposed to *shorter*, supporting the transformation of *shorter* to *taller*, rather than the reverse.

Symbolic Programming Languages

Several symbolic computer programming languages have been created to simplify the development of systems that use symbolic representations. The two most popular such languages are Lisp (e.g., Winston & Horn, 1981; Wilensky, 1984) and Prolog (e.g., Bratko, 1986; Sterling, 1986; Clocksin & Mellish, 1987). Lisp was developed in the early 1960s by John McCarthy and colleagues at MIT (based in part on work by Newell, Shaw and Simon at CMU). Prolog was developed by Alain Colmeraur and colleagues in France in the early 1970s. There are significant differences between the languages, but both provide ways of representing symbolic and propositional information, as well as variables and mechanisms for binding symbols or propositions to those variables. A great many symbolic models have been developed in Lisp and/or Prolog, and the languages continue to be popular for symbolic cognitive modelling.

Production Systems

Lisp and Prolog are general purpose symbolic programming languages. This means that while each supports the symbolic representation of information, each also employs a generic, flexible, control mechanism that is motivated by mathematical and logical concerns. Consequently the languages impose minimal constraints on models developed within them. This may be appropriate, however it has been argued that general purpose programming languages fail to capture important aspects of the nature or character of mental processing: specifically that mental processing can be understood in terms of the cyclic application of rules to a representation of one's current beliefs. This view has led to the development of production systems, which are general frameworks within which symbolic models may be expressed.

A production system consists of two fundamental components: a propositional database or store (in which current propositions, such as `taller(beth, anna)`, are stored) and a rule database (in which inference rules are held). Production systems function in a cyclic manner, with each cycle consisting of two phases. In the *recognise phase*, an inference rule is selected from the rule database according to a set of standard principles. In the *act phase* the selected rule is applied. The result is typically an alteration to the propositional store. The cycle may then repeat, with a different rule being selected and applied. Processing terminates either when the recognise phase fails to select a rule or when the selected rule explicitly signals the end of processing.

A production system's propositional store is generally referred to as its working memory, and the propositions contained in the store are referred to as working memory elements, or WMEs. The rule database (which may include many thousands of rules) is referred to as production memory, and the rules within the database are referred to as productions. Productions correspond to long-term knowledge, including both general knowledge and task-specific knowledge, and, like the inference rules in the previous section, typically contain variables that allow them to apply to many different WMEs.

More formally, a production consists of a set of conditions and a set of actions. For example, a variant on inference rule 2 (from the previous section) might consist of one condition and two actions:

IF: $\texttt{shorter}(\texttt{X}, \texttt{Y})$
THEN: delete $\texttt{shorter}(\texttt{X}, \texttt{Y})$
add $\texttt{taller}(\texttt{Y}, \texttt{X})$

This particular rule employs two kinds of actions (working memory addition and deletion), but other actions are possible (e.g., issuing motor commands, or terminating processing).

The variables contained within the rule (X and Y) mean that the rule may apply to any instance of the *shorter* relation. An *instantiated production* or *production instance* results from *mapping* or *binding* the variables. Thus, in the previous production rule X and Y might be bound to anna and beth respectively, yielding the following production instance:

IF: $\texttt{shorter}(\texttt{anna}, \texttt{beth})$
THEN: delete $\texttt{shorter}(\texttt{anna}, \texttt{beth})$
add $\texttt{taller}(\texttt{beth}, \texttt{anna})$

The result is an instruction to transform a specific instance of the *shorter* relation into the equivalent *taller* relation.

This example demonstrates that productions may add or delete specific propositions to or from working memory. WMEs, in contrast to productions, are generally specific and transient. As processing proceeds working memory evolves through the addition and deletion of individual WMEs as each production is applied. Productions, in contrast to WMEs, are non-specific and long-term. They are non-specific because they contain variables (and hence may apply in a range of situations). They are long-term because once entered in production memory they are generally not deleted.

One of the key elements of a standard production system is its conflict resolution procedure: the process within the recognise phase that governs the selection of one instance of a rule from all possible rule instances. Often, during the recognise phase, the contents of working memory will be such that the conditions of many different rules are met, and even a single rule may have its conditions met by many different WMEs (leading to many different instances of the rule). Different production systems employ different principles for selecting one rule instance from the set of applicable rule instances. Example mechanisms include: avoiding rule instances that have been selected previously, favouring rules with many conditions (and hence which are specific to the current situation) over rules with few conditions (which are more general and may be seen as specifying default or fall-back behaviours), favouring rules whose conditions match recently created WMEs over rules whose conditions match WMEs that have been present in working memory for some time, associating activation values with rule instances and selecting the most active rule instance, and (if all else fails) selecting one rule instance at random from those that have not been ruled out by other principles.

A production system model therefore consists of three components: a conflict resolution strategy (normally viewed as a fixed processing mechanism); a specification of the initial contents of working memory (i.e., the information on which the subject is able to act); and a set of productions (specifying both task-specific knowledge and general knowledge relevant to the task). Detailed examples of specific production systems applied to simple arithmetic are discussed in Chapter 3.

Analogies may be drawn between elements of the production system approach and the possible structure of the mind. In particular, the production system concept of working memory is an analogue of the psychological concept of working memory, and the production system concept of production memory is an analogue of the psychological concept of long-term memory. While these analogies are intriguing, they should not be taken literally. For example, the production system concept of working memory is generally not limited in capacity or subject to decay. The psychological concept is. Similarly, production memory is not normally divided into different types of long-term memory. In contrast, many psychologists distinguish between several forms of long-term knowledge, including procedural and declarative, and within declarative between episodic and semantic.

Production systems date back to the work of Post (1943), with the first implementations developed in the 1950s. However, it is of some interest to note that many production system concepts (such as rule-based processing and condition-action associations) were preempted over 20 years earlier by Selz (1913, 1922), an Austrian psychologist from the Gestalt school (see Chapter 4). Selz argued, at a time when British and American psychology was strongly behaviouristic, that problem solving involved the mental application of condition/action rules, and that associations between stimuli and responses were a function of the properties of and relations between stimuli in a given situation. Both notions are clearly visible in current conceptions of production systems.

1.9.2 Connectionist Models

One fundamental assumption of symbolic modelling is that cognitive functioning is largely independent of the implementation medium (i.e., neural tissue). This allows the development of abstract models that are based on high-level symbol manipulation. The connectionist approach rejects this assumption. Advocates of connectionism argue that properties of neural tissue (such as massively parallel computation through the interaction of many simple processing units) are of critical importance in modelling cognitive processes, and that cognition emerges from the interactions between processing units. From this perspective, they argue, it is a mistake to try to understand cognition purely in terms of the manipulation of symbols.

Table 1.1: Featural representations of some animals

		Animals				
		is mammal	can fly	has fur	has long tail	is vegetarian
Features	Person	1	0	0	0	0
	Cat	1	0	1	1	0
	Dog	1	0	1	1	0
	Bat	1	1	0	0	0
	Bird	0	1	0	1	0
	Mouse	1	0	1	1	1

Parallel Distributed Processing

Neurophysiology tells us that the brain consists of many billions of neurons. Each neuron may be analysed as a simple computing device which receives electrical impulses from other neurons. If the sum of impulses received in quick succession is sufficiently great, the neuron generates its own impulse, and transmits this to other neurons. Individual neurons operate in parallel, and computation is distributed across many interconnected neurons. The connectionist school of cognitive modelling has adopted an abstraction based on this parallel distributed approach to computation, and developed models of cognitive processes in terms of interacting networks of simple computing units.

Feature-Based Representations

The rejection of symbol manipulation by connectionism entails the rejection of symbolic propositional representations. It does not, however, entail the rejection of mental representation. Rather, it calls for a different approach. Standard connectionist networks consist of sets of nodes, with nodes having activation values (which may be thought of as corresponding to the firing rates of neurons). A representation in a connectionist network is thus a configuration of activation values across a set of nodes.

To illustrate, consider the highly artificial case of a network consisting of five nodes, with each node representing one of the following five features: *is mammal*, *can fly*, *has fur*, *has long tail*, and *is vegetarian*. A pattern of activation in which the first node is highly active and all other nodes are inactive might represent (or characterise) a person (or the category of people). Different patterns of activity across the nodes may represent different animals (or categories). Thus, Table 1.1, in which active nodes are indicated by the digit 1 and inactive nodes by the digit 0, illustrates the activities of nodes corresponding to a range of animals.

If we adopt a fixed order of features, we may represent the various types of animals by feature vectors. A person would correspond to the vector $\langle 1,0,0,0,0 \rangle$, and a cat would correspond to the vector $\langle 1,0,1,1,0 \rangle$. Note that this representation is unable to distinguish between cats and dogs: both have the same featural representation. In practice, many more features are typically needed to discriminate between possible represented objects.

Feature-based representations are well able to represent instances of objects or classes of objects (depending on the interpretation of the representation), but they lack the expressive power of symbolic propositional representations. Thus, the representation of relations between objects is only possible through indirect means (e.g., by using separate units to represent the relation and each object that lies in the relation) and those means generalise poorly to the representation of embedded propositions. While these limitations are not insurmountable (see Pollack, 1990), connectionist models nevertheless tend to focus on aspects of cognition that do not require the representation of relational information.

An Illustrative Model

To illustrate a simple connectionist-style model consider the task of category learning (e.g., Kendler & D'Amato, 1955; Kendler & Kendler, 1975). Subjects performing this task are presented with a set of objects or exemplars that differ along several dimensions (e.g., colour, shape, size), and required to learn which objects belong in which categories. After being shown an object, the subject may nominate a possible category. If the subject is incorrect he or she will be told the correct category. Most subjects are able to learn this task after a few tens of trials (assuming the number of dimensions is not too great and the categories share some underlying structure).

Consider the near trivial case in which objects differ along two dimensions (size and colour), with large objects belonging in category A and small objects belonging in category B. A connectionist model of performance on the task might consist of two sets of nodes, with four nodes in one set – the "input" nodes, corresponding to the features *large*, *small*, *black* and *white* — and two nodes in the other set — the "output" nodes, corresponding to the categories A and B. Each node in the input set would have a connection to each node in the output set, as in Figure 1.2.

Connections within connectionist networks are weighted, and activation may "flow" along a connection in proportion to the connection's weight. If the weights of the connections in Figure 1.2 from *large* to A and from *small* to B are near to one, and all other weights are near to zero, presentation of the feature vector $\langle 1,0,1,0 \rangle$ (representing a large black object) to the input nodes will cause the output node for category A to become active. In contrast, presentation of the feature vector $\langle 0,1,0,1 \rangle$ (representing a small white object) will cause the node

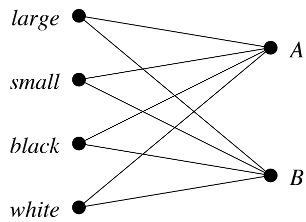

Figure 1.2: A simple network for the categorisation task

for category *B* to become active.

Learning, Generalisation and Graceful Degradation

Connectionist networks often exhibit several properties that make them particularly appropriate for modelling cognitive phenomena: learning from examples, generalisation to new examples, and graceful degradation in the face of ill-formed input or damage to the network.

Learning Suppose the connection weights in Figure 1.2 are initially set to random values. When an input vector is presented to the network, the output nodes will become active. For each input vector there is a unique target output vector (either $\langle 1, 0 \rangle$ or $\langle 0, 1 \rangle$). If the output generated by a given input fails to match the correct output (which is available to the subject, because the subject is given feedback on his or her categorisations), then the connection weights may be adjusted to improve the input/output mapping. For example, if the activation of an output node is greater than in the target pattern, then the weights on all connections feeding positive activation to the node should be decreased by a small amount (and the weights on all connections feeding negative activation to the node should be increased by a small amount). The opposite adjustment may be made if the activation of an output node is less than in the target pattern. Such weight adjustments will result in the network generating a more accurate categorisation when the input pattern is repeated at a later time. This is the basis of delta-rule learning, a simple but effective learning algorithm for one class of connectionist network.

When applied to the categorisation task described in the previous section, delta-rule learning results in strong weights between input features and output categories that co-occur. If, however, a feature is sometimes present and sometimes absent in a category (e.g., black or white, when categorisation is based on size), then the weight from the node representing that feature to the target category will be increased on some trials and decreased on other trials. If the feature is entirely independent of the category, the increases and decreases will average out and the resultant weight will be close to zero.

Generalisation One variant of the concept learning task involves presenting subjects with one set of exemplars in a training phase, and allowing subjects to learn the categories of all exemplars through feedback. Subjects are then presented with new exemplars, which share some but not all features with items from the training set. The subject's task is to generalise from previous experience and categorise these new exemplars. Connectionist networks provide a natural approach to such generalisation tasks, and for many tasks networks can be demonstrated that exhibit generalisation performance similar to that of human subjects.

Modelling Degraded Input Connectionist networks also lend themselves to modelling tasks in which input representations are partially degraded (corresponding, for example, to visual input under suboptimal viewing conditions, or auditory input with high levels of background noise). Feature-based input representations may be corrupted by random perturbations to one or more features with minimal effects on the network's behaviour. This allows the development of networks that show effects of input degradation similar to those obtained in experiments involving human subjects.

Graceful Degradation Large scale connectionist models, coupled with now standard learning techniques, typically develop some redundancy in their functioning. Consequently, such networks are relatively unaffected by minor damage (such as removing some nodes or some connections, or adding noise to activation values). This graceful degradation has been argued to provide a realistic reflection of the behaviour of the brain after natural cell death or even after minor brain damage. Several studies have also shown that larger scale damage to normally functioning connectionist networks can yield behaviours similar to those exhibited by patients with significant neural damage (e.g., Plaut & Shallice, 1993).

Types of Connectionist Network

Many different types of connectionist network have been developed. The network in Figure 1.2 is a simple feed-forward network (also known as a perceptron) consisting of two layers of nodes mediated by one layer of connections. Such networks are limited in their computational power: it can be shown mathematically that there are certain stimulus/response mappings that they cannot perform (Minsky & Papert, 1988). This limitation is not present in multi-layer perceptrons — feed-forward networks consisting of additional layers of units that mediate the stimulus/response relation — but such networks require more complex approaches to learning, and such approaches are generally considered to be biologically and psychologically implausible (Crick, 1989).

The layered structure of feed-forward networks also imposes limits on the sequential behaviour of such networks. Thus, feed-forward networks can only perform stimulus/response mappings in which a single response is associated with a single stimulus. They cannot perform mappings in which different responses are associated with different sequences of stimuli. Recurrent networks (see Jordon, 1986; Elman, 1990) overcome this limitation by including feedback connections between layers. These connections feed a representation of the state of processing at one point in time back into earlier layers of the network. Such connections allow, for example, the pattern of activation of an internal layer of nodes produced while processing one stimulus to affect the processing of the next stimulus. Recurrent networks have been used to model some of the sequential aspects of language.

Networks with feedback connections may also be used for non-sequential tasks by fixing the input vector and allowing repeated cycles of processing until the output vector stabilises. Networks of this form, which are known as attractor networks, have been used to model recognition processes, where a degraded representation of a stimulus may be refined through successive processing cycles (see, for example, Plaut & Shallice, 1993).

Networks need not have a layered structure. Associative networks (Hopfield, 1982; Hertz, Krogh, & Palmer, 1991) do not distinguish between input and output nodes. Any node within an associative network may be connected to any other node and functioning of the network corresponds to fixing or clamping the activation values of some nodes and allowing activation to flow along connections until the activations of unclamped nodes stabilise. Associative networks effectively act as pattern completion devices. They may be trained (again using standard learning algorithms) to store a number of activation patterns. After training the presentation of a partial pattern results in reconstruction of the original pattern. Interference between stored patterns may occur, giving associative networks some properties similar to human memory (Hopfield, 1982).

A different style of processing is evident in interactive activation networks. Nodes within these networks generally represent relatively high level concepts (e.g., letters or words: McClelland & Rumelhart, 1981), and "compete" for activation through processes of self excitation and mutual inhibition (see McClelland, 1992). Thus within McClelland and Rumelhart's interactive activation model of word recognition (McClelland & Rumelhart, 1981), nodes representing words receive excitation from nodes representing letters (which in turn receive activation from nodes representing features of the visual input), but word nodes mutually inhibit each other. Mutual inhibition ensures that only one word node may be active at a time. Similar inhibitory processes operate at the letter level to ensure that only one letter at each position of the word is active at a time. Interactive activation networks are appropriate for modelling tasks in which multiple sources of information interact to yield a single discrete outcome.

1.9.3 Hybrid Symbolic/Connectionist Models

Symbolic and connectionist approaches to modelling have both strengths and weaknesses. Arguably, the strengths of one approach are the weaknesses of the other, and *vice versa*. There is therefore the possibility that more adequate cognitive models may be developed by adopting a hybrid approach, in which both symbolic and connectionist aspects are incorporated. Cooper and Franks (1993) distinguish two types of hybrid model, corresponding to two ways in which symbolic and connectionist approaches have been combined. Physically hybrid models consist of separate symbolic and connectionist subsystems. These subsystems typically perform different functions and interact to yield the behaviour of the system as a whole. Non-physically hybrid systems, in contrast, consist of a single system (which is fully symbolic or fully connectionist), but that system can be described as functioning in both symbolic and connectionist terms.

The rationale for the physically hybrid approach is as follows. Symbolic models have achieved their greatest successes in relative high-level cognitive domains, such as reasoning and problem solving. Low-level domains, such as perception, are better modelled by connectionist approaches. This view is supported by the fact that there are relatively few tasks or domains for which both symbolic and connectionist models exist. It also suggests that tasks that can be decomposed into a mixture of high-level sub-processes and low-level sub-processes might be best modelled by hybrid systems in which separate subsystems perform the separate sub-functions.

One system that employs the physically hybrid approach is Sun's model of common-sense reasoning (Sun, 1994). The model uses two subsystems: a symbolic system for representing reasoning rules (such as *all men are mortal*) and a connectionist system for representing the "sub-conceptual content" of the elements involved in those rules (e.g., the concept of *Socrates* and the category of *men*). Sub-conceptual content is represented using fine-grained feature-based representations. Links between the subsystems allow the simulation of flexible rule-based reasoning.

The non-physically hybrid approach is well illustrated by the connectionist production system of Touretzky and Hinton (1988). Touretzky and Hinton showed how connectionist techniques could be used to implement the structures and processes of a typical symbolic production system (including working memory, production memory, and symbolic rules containing variables). In principle, a symbolic production system model of a specific task could be simulated by Touretzky and Hinton's system by providing the system with an appropriate, feature-based representation of the symbolic rules. The functioning of such a system could be legitimately described in both connectionist and symbolic terms.

1.9.4 Cognitive Architectures

The last decade has seen the rise of an alternative approach to cognitive modelling that is orthogonal to the symbolic/connectionist distinction. This is the use of cognitive architectures (Newell, 1990). Cognitive architectures are theories of the large-scale structure and organisation of cognitive processing. They are theories of the functional subsystems that make up the mind/brain, and the modes of interaction between those subsystems.

The concept of cognitive architecture derives from an analogy with that of computer architecture. A computer architecture consists of a configuration or structuring of a number of components (a central processing unit, a data bus, RAM, disk drives, input and output devices, etc.). A cognitive architecture similarly consists of such a configuration, where components may include a short-term or working memory, a long-term memory, a language subsystem, perceptual and motor subsystems, one or more learning mechanisms, and so forth.

Cognitive architectures attempt to provide a general framework or set of constraints within which models of specific tasks or domains may be developed. The basic approach was first championed by Newell (1990) (see also Newell, 1973). Examples include Soar (Laird, Newell, & Rosenbloom, 1987; Newell, 1990), ACT-R (Anderson, 1983, 1993; Anderson & Lebiere, 1998), CAP (Schneider & Detweiler, 1987; Schneider & Oliver, 1991), and EPIC (Meyer & Kieras, 1997; Kieras, Meyer, Ballas, & Lauber, 2000). Soar and EPIC are symbolic architectures based on production system concepts. ACT-R, which is described further below, is a hybrid architecture. CAP is a connectionist architecture.

The architectural perspective on cognition views behaviour on any particular task as the product of a general architecture working with task-specific knowledge. Development of a model of a task within an architecture therefore involves supplying the architecture with appropriate task-specific knowledge. For architectures based on production systems, this generally involves supplying an appropriate set of production rules. For other architectures it involves supplying the knowledge in the form of input/output patterns with which the architecture may be trained.

Of the above mentioned architectures, ACT-R is currently the most highly influential. ACT-R is a physically hybrid architecture. At its centre is an activation-based production system. This consists of the standard production system components (as described in Section 1.9.1), augmented with a learning mechanism and perceptual and motor subsystems. What makes ACT-R distinctive is that elements in working memory have activation levels, and these activations may be propagated to production instances. Conflict resolution is then effected by firing the first production instance that becomes sufficiently active. Production firing results in the addition of new working memory elements, the excitation of existing elements, or the execution of motor commands. ACT-R has been used to model a wide range of phenomena, including choice tasks, arithmetic, memory for word lists, analogy, and dual task performance (Anderson & Lebiere, 1998).

1.10 Strategies for the Use of Simulation

There are thus several distinct approaches that may be adopted in developing a cognitive model. There are also several distinct ways in which modelling and simulation may be used to advance our knowledge and understanding. Simulation provides a collection of tools and methods that can be used within different scientific disciplines for different scientific purposes and even at different stages in the development of a field. The following paragraphs consider some of the different strategies for using computer simulation within cognitive modelling. The strategies are discussed in order of increasing scientific power.

The first kind of simulation method might be called a "fishing trip", by analogy with the angler who casts a fishing rod into a pond with little idea of what may be in the pond, or even if there is anything of interest in the pond at all. The angler may be gambling but is probably not wasting his or her time, because after a few casts some useful information may have been found about the pond. Either there are fish in the pond, or there probably aren't, and one should try fishing elsewhere.

The fishing trip method of simulation therefore consists of attempting to develop a model of some task or behaviour in order to learn more about the task. Fishing trips can be useful to cognitive scientists just as they are to anglers, particularly when they are trying to make sense of a new approach or a new scientific area. In trying to build a simulation of some task, for example, we may discover that we are unclear about what we think the problem is, what the possible solutions are, that the kind of theory we were thinking about is too simple, or even that there is some very good reason why it cannot work at all.

The second strategy involves implementing a pre-existing (verbally specified) theory, and determining if the theory behaves as claimed. This form of modelling is particularly useful when the verbal theory is highly complex. This form of modelling is a kind of sufficiency test: it allows one to determine if a set of theoretical assumptions (as outlined in the verbal specification) is sufficient to account for the target behaviour. If they are sufficient, all well and good. If they are not, one may then go on a fishing trip in an attempt to find how they might be altered.

A different approach to simulation involves carrying out an *a priori* analysis of the properties of the kind of theory that is being considered. One may develop a model and determine, for example, how sensitive its behaviour is to the underlying theoretical assumptions. In this way one may identify critical parameters and appropriate values for those parameters. Equally one may identify theoretical assumptions that are secondary — assumptions that have non-significant effects on the behaviour of the model.

Finally, the most powerful use of simulation techniques is in supporting the conventional use of the hypothetico-deductive method that is widely used by scientists across many disciplines. This method involves using a model to simulate behaviour beyond that which was employed in the development of the model, and thereby generating predictions or hypotheses. These hypotheses may then be tested by conducting behavioural experiments with real people. If people behave as the model predicts then the model gains empirical support.

1.11 Closing Remarks

In this chapter we have discussed the nature and roles of cognitive modelling and described the principal modelling techniques that have been adopted by the cognitive science community.

Historically, different sub-communities have taken rather different approaches to modelling. Some have preferred to take a "top-down" approach starting with high level cognitive functions like reasoning or problem solving and trying to understand the kinds of cognitive processes that are needed to implement such functions, and traditionally the methods adopted have emphasised symbol manipulation and representation of knowledge. Other scientists have preferred a "bottom-up" approach, to give a detailed account of observed behaviour in tasks like concept learning or reading; these are often well-explained by connectionist or statistical mechanisms. Arguments between these two communities can be quite vigorous, frequently centering on the question of whether the mechanisms of mind/brain are "really" symbolic or "really" connectionist.

Another dimension along which cognitive models can be compared concerns whether they set out to provide unified accounts of cognitive processes, typically as large-scale information processing architectures, or whether they are "micro-models" of small scale phenomena. Debates are again lively. Advocates of micro-modelling argue that cognitive scientists should put explanation of natural phenomena first, and not build theoretical palaces that cannot be justified empirically. Other scientists emphasise the need to develop unified theories of intelligence and cognition, and the information processing principles that make any kind of intelligence possible. This dimension actually implies a trade-off between broad theoretical generality and detailed empirical adequacy, so examples of models can be found at all points in between the two extremes.

In this book we set out to be inclusive rather than disputatious. To do this we have tried to demonstrate an integrated approach to cognitive modelling which we hope will have something to offer to researchers of all persuasions. In particular we introduce a set of tools and methods, collectively called the COGENT cognitive modelling environment, which we think can be used to support many cognitive modelling styles. We believe that COGENT can offer this for several reasons.

First, we view the set of tools that COGENT offers as merely that, tools; we don't need to take a position on whether the different kinds of representation that the system offers embody some sort of truth about the mind or brain. As we have discussed, computational models are intrinsically abstractions from reality, and modelling always represents some kind of approximation. It is up to the individual COGENT user to decide to what extent his or her model embodies reality.

Second, we observe that different ways of thinking and modelling are productive in different sub-areas of cognitive science, so COGENT supports several standard approaches. COGENT users may build rule-based, activation-based, simple connectionist or even conventional numerical simulations. Indeed they may even build hybrid models that combine different representations.

Third, we have tried to provide sufficient representational and computational power to permit scientists to build simulations at grossly different scales, from micro-models to unified architectures or anywhere in between. Indeed a user can arrange that different parts of the system of interest are modelled at different levels or detail. COGENT supports a highly modular approach to modelling cognitive systems, with any or all modules programmable at a coarse level, or in finer detail by recursively composing modules out of smaller components to whatever level is required.

Finally, we believe that COGENT is equally sympathetic to the theoretician and the empirically minded scientist. We believe that it offers an unprecedented range of formal tools with the expressive power to accommodate many different theoretical frameworks. On the other hand we know that science proceeds through systematic experiment, with careful collection and analysis of data in varied and controlled conditions, and rigorous comparison of predictions and observations. Apart from the modelling tools COGENT provides it also includes facilities for managing computational experiments, automatically running simulations under varying assumptions and storing the data, comparing simulation results with laboratory data, and so on. We hope that subsequent chapters will clearly demonstrate these capabilities, and help to build bridges between the many different sub-communities of cognitive science.

1.12 Further Reading

Dawson (1998) provides an excellent introduction to the computational theory behind cognitive science, including chapters on symbolic modelling, connectionist modelling, and the relation between the two. Dawson also discusses the key issue of levels of description, which is addressed in several places throughout this book. Further background is provided in the opening chapters of Green and Others

(1996).

There are relatively few texts that focus on symbolic modelling. Scott and Nicolson (1991), which provides a set of "cognitive science projects", is one exception. A more advanced text, which focuses on the production system approach, is Klahr, Langley, and Neches (1987). Anderson and Lebiere (1998) is also of considerable interest. This book presents a number of symbolic models spanning several high-level cognitive domains. It uses the ACT-R cognitive architecture to provide a unified framework for the models.

The connectionist approach is better served by texts. Much of the connectionist revival in the 1980s can be traced to the two volumes by McClelland, Rumelhart and the PDP research group (McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986b), and these remain of significant scientific interest. More recent texts include McLeod, Plunkett, and Rolls (1998) and O'Reilly and Munakata (2000). The second of these is also strong on the rationale of cognitive modelling.

Van Gelder (1998) presents a manifesto for the dynamical approach to modelling. The manifesto is accompanied by a number of critical commentaries. Examples of the dynamical approach can be found in the edited collection of Port and van Gelder (1995).

The case for cognitive architectures is presented by Newell (1990, 1992). Newell's focus is on one specific "candidate" architecture, Soar, but his arguments are phrased in general terms. Some concerns about the architectural approach are expressed in the commentaries accompanying Newell (1992), and by Cooper and Shallice (1995).