Brit. J. Phil. Sci. **41** (1990), 195–222 *Printed in Great Britain*

Connectionism, Competence, and Explanation

ANDY CLARK

ABSTRACT

A competence model describes the abstract structure of a solution to some problem, or class of problems, facing the would-be intelligent system. Competence models can be quite detailed, specifying far more than merely the function to be computed. But for all that, they are pitched at some level of abstraction from the details of any particular algorithm or processing strategy which may be said to realize the competence. Indeed, it is the point and virtue of such models to specify some *equivalence class* of algorithms/processing strategies so that the common properties highlighted by the chosen class may feature in psychologically interesting accounts. A question arises concerning the type of relation a theorist might expect to hold between such a competence model and a psychologically real processing strategy. Classical work in cognitive science expects the actual processing to depend on *explicit or tacit* knowledge of the competence theory. Connectionist work, for reasons to be explained, represents a departure from this norm. But the precise *way* in which a connectionist approach may disturb the satisfying classical symmetry of competence and processing has yet to be properly specified. A standard 'Newtonian' connectionist account, due to Paul Smolensky, is discussed and contrasted with a somewhat different 'rogue' account. A standard connectionist understanding has it that a classical competence theory describes an *idealized subset* of a network's behaviour. But the network's behaviour is not to be explained by its embodying explicit *or* tacit knowledge of the information laid out in the competence theory. A rogue model, by contrast, posits either two systems, or two aspects of a single system, such that one system does indeed embody the knowledge laid out in the competence theory.

1 SCENE SETTING

In the old days, we all knew what it meant to describe the mind as a *syntactic engine*. A syntactic engine was a physical system cleverly designed so that the way some of its physical states gave way to other physical states was always in step with the way that good inferences proceeded in some particular domain. For example, some states might be used to stand for a category such as dog, and the physical system set up so that those states reliably gave way to others which could be interpreted as standing for sub- and super-ordinate categories (such as 'Fido' and 'Mammal').

We understood that such an effect (the mirroring of semantic regularities in syntactic systems) was made possible by the system's being geared to manipulate *symbols* according to *rules*. Symbols were recurrent physical states which we could interpret (*e.g.* as standing for dog) and the system could either embody the rules explicitly or implicitly (see the text).

Connectionist systems (see Rumelhart and McClelland [1986], Smolensky [1988], Clark [1989]) appear to offer a somewhat different way of ensuring that a physical system is semantically well-behaved. In (highly distributed Smolensky-style) connectionist models, there are often no neat recurrent physical states which code for the real world entities which the system is dealing with. Instead of being a syntactic engine in which semantic good behaviour is ensured by having the system directly implement symbolic descriptions of the objects and processes which its inferences concern, the (Smolensky-style) connectionist opts for a *statistical engine* operating on computational objects which do not neatly stand for the objects and processes in the domain. (These objects are often called 'subsymbols'.) Nonetheless, in a central class of cases, the system behaves *as if* it were a symbolic/syntactic engine. (For a particularly clear account of this proposal, see Smolensky [1987] pp. 137–49.)

In what follows I explore some implications of this novel way of being semantically well-behaved. In particular, I ask how well a standard model of explanation in cognitive science (Marr's 3-level model) describes the connectionist's procedure and theory, and whether a *failure* to fit such a model implies a lack of explanatory power. I begin, then, with some general comments on explanation.

2 LEVELS OF EXPLANATION AND THE IDEA OF AN EQUIVALENCE CLASS

Explanation, it seems, is a many-levelled thing. A single phenomenon may be subsumed under a panoply of increasingly general explanatory schemas. On the swings and roundabouts of explanation, we trade the detailed descriptive/explanatory power of lower levels for a satisfying width of application at higher

ones. And at each such level there are virtues and vices; some explanations may be available only at a certain level; but individual cases thus subsumed may vary in ways explicable only by descending the ladder of explanatory generality.

For example, the Darwinian, or neo-Darwinian theory of natural selection is pitched at a very high level of generality. It pictures some very general circumstances under which 'blind' selection can yield apparently teleological (or purposeful) evolutionary change. What is required for this miracle to occur is differential reproduction according to fitness and some mechanism of transmission of characteristics to progeny. The virtue of this top-level explanation, then, lies in its covering an open-ended set of cases in which very different actual *mechanisms* (*e.g.* of transmission) may be involved. In this way it defines an *equivalence set* of mechanisms, that is, a set of mechanisms which may be disparate in many ways but which are united by their ability to satisfy the Darwinian demands.

The natural accompaniment to virtue is, of course, vice; and the vice of the general Darwinian account is readily apparent. We do not yet know, in any given case, *how* the Darwinian demands are satisfied. That is to say, we do not yet have the foggiest idea of the actual mechanisms of heritability and transmission in any given case. Moreover, there may well be facts about some specific class of cases (*e.g.* recessive characteristics in Mendel's peas) which are not predicted by the general Darwinian theory, which gives us still further reason to seek a more specific and detailed account.

Mendelian genetics offers just such an account. It posits a class of theoretical entities (genes, as they are now called) controlling each trait, and describes the way such entities must combine to explain various observed facts concerning evolution in successive generations of pea plants. The specification included, for example, the idea of pairs of genes (genotypes) in which one gene may be dominant, thus explaining the facts about recessive characteristics. (For an accessible account of evolutionary theory and Mendelian genetics, see Ridley [1985].)

We may note in passing that between any two levels (*e.g.* Darwinism and Mendelian genetics) there will almost certainly be other, theoretically significant levels. Thus Mendelian inheritance is in fact an *instance* of a more general mechanism called Weismannist inheritance (see Ridley [1985], p. 23.) But Weismannist inheritance is still *less* general than Darwinian inheritance. Weismannism carves off a theoretically unified subset of general Darwinian cases. And Mendelism carves off a theoretically unified subset of Weismannism. At each stage the equivalence class is strategically redefined to exclude a number of previous members. We can visualize this as a gradual shrinking of the size of the equivalence class, although this may not be *strictly* true, since each new class has a possible infinity of members and so they are, I suppose, identical in size!

Mendelian genetics provides an interesting case for one further reason. It was originally conceived as *neatly specifying* the details of lower level DNA-based inheritance (*i.e.* of the hardware realization of an inheritance mechanism). As Dennett puts it, Mendelian genes were seen as specifying:

> the language of inheritance, straightforwardly realised in hunks of DNA (Dennett [forthcoming] p. 3).

This corresponds to what we shall be terming the classicist vision of the relation between a certain level of abstract theorizing in cognitive science (competence theorizing) and actual processing strategies.

But in fact, according to Dennett:

> there are theoretically important mismatches between the language of 'bean-bag genetics' and the molecular and developmental details—mismatches serious enough to suggest that all things considered, there don't turn out to be genes (classically understood) at all (Dennett [forthcoming] p. 3).

This looks like (and is regarded by Dennett as) an analogue of the connectionist's view of the fate of the constructs of a classical competence theory.

Be that as it may, the point for now is simply this, that beneath the level of Mendelian genetics there is some further level of physical implementation (with God knows what in between, as remarked earlier), and this completes our descent down the ladder of explanatory generality. We start at the top level (level 1) with the general Darwinian theory defining a large and varied equivalence class of instantiating mechanisms. We descend to a more detailed specification of a subclass of mechanisms (Mendelian theory) and thence, one way or another, to the details of the implementation of those mechanisms in DNA (Figure 1). The effect is a kind of triangulation upon the actual details of earth-animal inheritance from much broader explanatory principles governing whole sets of possible worlds.

Explanation in cognitive science, as conceptualized by, among others, Marr, Chomsky and Newell and Simon, has a similar multi-layered structure. For a given class, or class of tasks (*e.g.* vision, parsing etc.) there will be a top level story which comprises 'an abstract formulation of *what* is being computed and *why*', a lower level which specifies a particular algorithm for carrying out the computation, and (still lower) an account of how that algorithm is to be realized by physical hardware. To illustrate this, Marr gives the example of Fourier analysis. At the top level we have the general idea of a Fourier analysis. This can be realized by several different algorithms. And each algorithm in turn can be implemented in many different kinds of hardware organization (see Marr [1977], p. 129).

There is an important gap between the 'official' account of the top level (level 1 as Marr calls it) and the actual practice of giving 'level 1' theories. For although the official line is that a level 1 account specifies only the what and

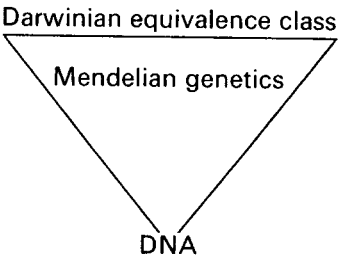

FIGURE 1 The swings and roundabouts of explanatory generality. What DNA-based stories gain in detailed power they lose in cross-world scope.

the why of a computation, this specification can be progressively refined so as to define a more informative (*i.e.* more restrictive) equivalence-class. This more refined version of level 1 theorizing (which yet falls short of a full algorithmic account) has been persuasively defended by Christopher Peacocke under the title of 'Level 1·5' (see Peacocke [1986]).

The contrast Peacocke highlights is between an equivalence class generated by defining a function *in extension* (*i.e.* by its results—the *what*, in Marr's terms) and a more restrictive (and informative) equivalence class generated by specifying the *body of information* upon which an algorithm draws. Thus, to adapt one of Peacocke's own examples, suppose the goal of a computation is to compute depth D and physical size P from retinal size R. And suppose, in addition, that this computation is to occur inside a restricted universe of values of D, P and R. Specifying the function in extension merely tells us that whenever the system is given some D and P as input, it should yield some specified R as output. One way of doing this (and I here adapt a strategy used by Martin Davies (see Davies [forthcoming] and Section 3 following), is to store the set of legal values of R for every combination of values of D and P—a simple look-up table. A second way is to process data in accordance with the equation $P = D \times R$. In saying that the system draws on the information that $P = D \times R$, we are, as Peacocke insists, doing *more* than specifying a function in extension. For the look-up table *does not* draw on that information, yet it falls within the equivalence class generated by the function in extension specification. But we are doing *less* than specifying a particular algorithm, since there will be many ways of computing the equation in question (*e.g.* using different algorithms for multiplication).

It is this grain of analysis (*i.e.* what Peacocke calls level 1·5) that I will have in mind when speaking, in the remainder of this paper, of a *competence theory*. This seems to accord at least with the practice of Chomsky, who coined the term competence theory to describe the pitch of his own distinctive investigations into the structure of linguistic knowledge. And it may well

accord with Marr's actual practice at 'level 1' though not with the official dogma.

A Chomskian competence theory does far more than specify a function in extension. Instead it seeks to answer (at a level of abstraction from the physical mechanisms of the brain and from specific algorithms) the question 'What constitutes knowledge of language?'. In so doing it seeks a 'framework of principles and elements common to attainable human languages' (Chomsky [1986], p. 3). And (in its most recent incarnation) it characterizes that framework as a quite specific

> system of principles associated with certain parameters of variation and a markedness system with several components of its own (Chomsky [1986], p. 221).

It does not matter, for the purposes of this paper, just what principles and parameters Chomsky actually suggests. Rather, we should merely note that if a competence theory is as definite and structured as a Chomskian model (and its his word, after all!) then it is more like a level 1·5 analysis than a simple level 1 account. For it describes, at a certain level of abstraction, the structure of a form of processing (by specifying the information drawn on by the processes) and hence helps 'guide the search for mechanisms' (Chomsky [1986], p. 221). In short, it is more like Mendelian genetics than General Darwinism. Rather than being merely *descriptive* of a class of results, it is meant also to be *suggestive* of the processing structure of a class of mechanisms of which we are a member. Whether competence theories (at least as we currently know them) actually *are* suggestive of the form of human processing is the topic of this paper.

3 THE CLASSICAL CASCADE

A competence theory, then, leads a double life. It both specifies the function to be computed *and* it specifies the body of knowledge or information which is used by some class of algorithms. In classical cognitive science, these two roles can easily be simultaneously discharged. For the competence theory is just an articulated set of rules and principles defined over symbolic data-structures. Since classical cognitive science relies on symbol processing architecture, it is natural (at level 2) to represent directly the data structures (*e.g.* structural descriptions of sentences) and then carry out the processing by the explicit *or* tacit representation of the rules and principles defined (in the competence theory) to operate on those structures. Thus, given a structural description of an inflected verb as comprising a stem plus an ending, the classicist can go on to define a level 2 computational process to take the stem and add -ed to form the past tense (or whatever). The classicist, then, is (by virtue of using a symbol processing architecture to implement level 2 algorithms) uniquely well placed to preserve a very close relation between a competence theory and its level 2

implementations. Indeed, it begins to seem as if that close relation is what is *constitutive* of a classical approach. Thus Dennett visualizes the classicist dream as involving 'a triumphant cascade through Marr's three levels' (Dennett [1987], p. 227). Such a characterization of the essential classicist vision seems to me to fit very nicely with Fodor and Pylyshyn's recent account of the classical/connectionist divide.

Fodor and Pylyshyn argue that there are two fundamental differences between truly connectionist and classical approaches to cognitive modelling. ('Truly connectionist' here rules out those cases where a units and connections sub-structure is used to implement a classical theory). The differences are:

1. 'Classical theories—but not connectionist theories—posit a "language of thought"'.

   This means they posit mental representations (data-structures) with a certain form. Such representations are *syntactically structured, i.e.* they are systematically built by combining atomic constituents into molecular assemblies which in turn (in complex cases) make up whole data-structures. In short, they posit *symbol systems* with a combinatorial syntax and semantics.
2. 'In classical models, the principles by which mental states are transformed, or by which an input selects the corresponding output, are defined over structural properties of mental representations. Because classical mental *representations* have combinatorial structure, it is possible for classical mental *operations* to apply to them by reference to their form'.

   This means that *given* that you have a certain (language-like) kind of structured representation available (as demanded by point 1), it is possible to define computational operations on those representations so as the operations are sensitive to that very structure. If the structure was not there (*i.e.* if there was no symbolic representation) you could not do it! (Though you might make it *look* as if you had by fixing a suitable function in extension.) (Quotes are from Fodor and Pylyshyn [1988], pp. 12–13.)

In short, a classical system is one which posits syntactically structured symbolic *representations* and which defines its computational *operations* to apply to such representations in virtue of their structure.

The computational operations, in any such case, can be described by transition or derivation rules defined over syntactically structured representations. For example:

If (A and B) then (A)
If (A and B) then (B)
If (stem + ending) then (stem + -ed)

The parenthesized items are structural descriptions which will pick out open-ended classes of classical representations. The 'If–then' specifies the operation.

But note that the classicist, under the terms of the act, is *not* committed to the systems *explicitly* representing the 'if–then' clause. All that needs to be explicit is the structured description upon which it operates. Thus a machine could be hard-wired so as to take expressions of the form (A and B) and transform them into the expressions (A) and (B). The derivation rules may thus be implicit, or *tacit*; but the data-structures must be explicit. On this matter, Fodor and Pylyshyn are rightly insistent:

> Classical machines can be *rule implicit* with respect to their programs. . . . What *does* need to be explicit in a classical machine is not its program but the symbols that it writes on its tapes (or stores in its registers). These, however, correspond not to the machine's rules of state transition but to its data structures. (Fodor and Pylyshyn [1988] p. 61.)

As an example they point out that the grammar posited by a linguistic theory need not be explicitly represented in a classical machine. But the *structural descriptions of sentences* over which the grammar is defined (*e.g.* in terms of verb stems, sub-clauses etc.) must be. A successful 'classical cascade' from a linguistic competence theory to a level 2 processing story can thus tolerate having the rules of the grammar built into the machine. Those attempts to characterize the classicist/connectionist contrast solely by reference to the explicitness or otherwise of rules are thus shown to be in error.

Now, however, there is a danger of losing sight of the way in which (for a classicist) the competence theory (or set of derivation rules and data-structures) is meant to bear a close relation to the level 2 implementation. For we said (Section 2 above) that given, say, a simple competence theory like 'P = D × R' it *would not* do to have a system which simply stored, for some finite universe of discourse, all legal values of P, D, and R. Yet such a system certainly has explicit representations of P, D, and R. So *if* it will not do, it must be because it lacks *even tacit* knowledge of the derivation rule 'P = D × R'. The question then is, how do we motivate this difference? What are the constraints on tacit knowledge ascription such that the rule 'P = D × R' *need not* be explicitly represented, but which rules out the look-up table as an instance of tacit knowledge of *that very rule*? The answer will be significant when we come to ask (Sections 4 and 5) whether connectionist systems have tacit knowledge of classical rules.

Martin Davies (drawing on information provided by Gareth Evans) offers the following suggestion:

> For a Speaker to have tacit knowledge of a particular articulated theory, there must be a causal-explanatory structure in the Speaker which mirrors the derivational structure in the theory (Davies [forthcoming], p. 4).

By 'the derivational structure in the theory' Davies means the transition rules (*e.g.* P = D × R). What is it, then, to embody a 'causal-explanatory structure' which 'mirrors' such a derivational structure? Simply, according to Davies, for

there to be a *causal common factor* (Davies' phrase) in the processing story told for each instance which, at the higher level, is seen as involving the rule of derivation. Thus, in the case of the look-up table, there need be no causal common factor in the processing of all the instances of various values of P, D, and R. Conversely, if there *is* a causal common factor through which all processing is routed, then (subject to some niggling provisos—see Davies [forthcoming] and [1987]) the system is rightly said to have tacit knowledge of the rule. A result which, as Davies notes, sits nicely with our cognitive neuropsychological intuitions. For systems which meet the tacit knowledge constraint so construed are prima facie candidates for a type of breakdown in which damage to the causal common factor causes total loss of capacity to solve a whole class of problems (*e.g.* P, Q, and R specifying). Whereas systems which fail to meet the constraint are prima facie candidates for less systematic deficits—*e.g.* the look-up system may lose its knowledge of some legal combinations of P, Q, and R but preserve its knowledge of others. Similar comments apply to the past tense generation case. Systems with tacit knowledge of the rule 'take stem and add -ed' could lose all capacity to form regular pasts. Systems which do it by look-up would not.

Davies' account (modulo a quibble about *virtual* 'causal' common factors see Section 4 following) seems convincing. If we take it on board, we end up with the following characterization of any properly *classical* cognitive model:

> (Classicism defined by an attitude to competence theories.) A cognitive model is classical iff it has a processing level description which bears a certain rather close relation to the structure of a standard competence theory. A standard competence theory posits a set of rules or principles of derivation defined to apply to a class of structured, symbolic representations according to their form. The close relation required involves (1) the explicit representation, in the processing level description, of the structured representations over which the rules are defined, and it involves (2) the explicit *OR* tacit representation of those rules and principles themselves. A rule or principle is judged to be tacitly represented just in case there is a causal common factor in the processing level description which is in play whenever the rule or principle is invoked in a competence-level specification of a transition.

Such, in tortuous detail (and apologies for that) is the substance of the 'classical cascade' through Marr's levels of explanation. Connectionism dams the cascade. How it does so, and what water-courses result, will occupy us for the remainder of this paper.

4 NEWTONIAN COMPETENCE

The connectionist vision of the relation between a structured competence theory and a level 2 processing story is radically unlike the neat 'cascade' imagined in Section 3. Instead of the level 2 story mirroring the derivational

form of the competence theory, it is seen as relating to it in rather the way Newtonian mechanics relates to quantum physics. The physical universe is not, in fact, Newtonian. But under certain specifiable conditions, it behaves very must *as if* it were. Newtonian principles thus describe and predict the behaviour of physical systems in a range of cases. But, in some intuitive but slightly elusive sense, those principles do not describe the actual forces which determine physical behaviour. The analogy is much favoured by Rumelhart and McClelland who write:

> It might be argued that conventional symbol processing models are macroscopic accounts, analogous to Newtonian mechanics, whereas our models offer more microscopic accounts, analogous to quantum theory—Through a thorough understanding of the relationship between the Newtonian mechanics and quantum theory we can understand that the macroscopic level of description may be *only an approximation* to the more microscopic theory. (Rumelhart and McClelland [1986] p. 125).

To illustrate this point, consider a simple example due to Paul Smolensky. Imagine that the cognitive task to be modelled involves answering qualitative questions concerning the behaviour of a particular electrical circuit. (The restriction to a single circuit may appal classicists, although it is defended by Smolensky on the grounds that a small number of such representations may act as the 'chunks' utilized in general purpose expertise—see Smolensky (1986] p. 241).) Given a description of the circuit, an expert can answer questions such as 'If we increase the resistance at point A what effect will that have on the voltage?' (*i.e.* will the voltage increase, decrease, or remain the same?).

Suppose, as seems likely, that a high-level competence-theoretic specification of the information to be drawn on by an algorithm tailored to this task cites various laws of circuitry on its derivations (what Smolensky refers to as the 'hard laws' of circuitry; Ohm's law and Kirchoff's law). For example, derivations involving Ohm's law would invoke the equation

$$\text{Voltage (V)} = \text{Current (C)} \times \text{Resistance (R)}.$$

We recognized, in Section 3 above, just two ways in which a level 2 processing story might bear an appropriately close relation to such a competence theory. In the simplest case, the processing might involve a symbolic representation of Ohm's law which is read and followed by the system. In the more complex case, it might involve tacit knowledge of Ohm's law unpacked in terms of a causal common factor in a set of state transitions. (Note in passing: Smolensky's own treatment here seems to place uncalled for emphasis on the simple option—see Fodor and Pylyshyn [1988], Pinker and Prince [1988], Davies [forthcoming], Clark [1989].)

Neither cascade is operative in the case of Smolensky's connectionist model of simple circuit problem solving. To see why, we need to look at the form of the model in question.

The model represents the state of the circuit by a pattern of activity over a set of feature units. These encode the qualitative changes found in the circuit variables, *i.e.* in training instances, they encode whether when the resistance at R_1 goes up, the overall voltage falls or rises and so forth. These feature units are connected to a set of what Smolensky calls 'knowledge atoms' which represent patterns of activity across subsets of the feature units. These in fact encode the legal combinations of feature unit states allowed by the actual laws of circuitry. Thus, for example:

> The system's knowledge of Ohm's law . . . is distributed over the many knowledge atoms whose subpatterns encode the legal feature combinations for current, voltage and resistance. (Smolensky [1988] p. 19.)

In short, there is a sub-pattern for every legal combination of qualitative changes (GS sub-patterns, or 'knowledge atoms' for the circuit in question).

It might seem, at first sight, that the system is merely a units and connections implementation of a look-up table. But this is not so. In fact, connectionist networks act as look-up tables only when they are provided with an overabundance of hidden units and hence simply memorize input-output pairings. By contrast, the system in question encodes what Smolensky terms 'soft constraints', *i.e.* patterns of relations which usually obtain between the various feature units (microfeatures). It thus has 'general knowledge' of qualitative relations among circuit microfeatures. But it does *not* have the general knowledge encapsulated in *hard* constraints like Ohm's law. The soft constraints are two-way connections between feature units and knowledge atoms which *incline* the network one way or another, but do not *compel* it; that is, they can be overwhelmed by the activity of other units—that is why they are 'soft'. And as in all connectionist networks, the system computes by trying simultaneously to satisfy as many of these soft constraints as it can. To see that it is not a mere look-up tree of legal combinations we need only note that it is capable of giving sensible answers to (inconsistent or incomplete) questions which *have* no answer in a simple look-up table of legal combinations.

The soft constraints are numerically encoded as weighted inter-unit connection strengths. Thus problem solving is achieved by 'a series of many node (*i.e.* unit) updates, each of which is a *microdecision* based on formal *numerical* rules and numerical computations' (Smolensky [1986], p. 246).

The network has two properties of special interest to us. First, it can be shown that *if* it is given a well-posed problem *and* unlimited processing time it will *always* give the correct answer as predicted by the hard laws of circuitry. But, as already remarked, it is by no means bound by such laws. Give it an ill-

posed or inconsistent problem and it will satisfy as many of the soft constraints (which are all it really knows about) as it can. Thus:

> Outside of the idealized domain of well-posed problems and unlimited processing time, the system gives sensible performance (Smolensky [1988], p. 19).

The hard rules (Ohm's law etc.) can thus be viewed as an external theorist's characterization of an idealized subset of its actual performance (it is no accident if this puts us in mind of Dennett's [1987] claims about the 'intentional stance').

Second, the network exhibits interesting *serial* behaviour as it repeatedly tries to satisfy all the soft constraints. This serial behaviour is characterized by Smolensky as a set of *macrodecisions* each of which amounts to a 'commitment of part of the network to a portion of the solution'. These macrodecisions, Smolensky notes, are:

> approximately like the firing of production rules. In fact, these 'productions' 'fire' in essentially the same order as in a symbolic forward-chaining inference system (Smolensky [1988], p. 19).

Thus the network will look as if it is sensitive to hard, symbolic rules at quite a fine grain of description. It will not *simply* be that it solves the problem 'in extension' as if it knew hard rules. Even the *stages* of problem solving may look as if they are caused by the system's running a processing analogue of the steps in the symbolic derivations available in the competence theory.

But the appearance is, on the terms set out in Section 3 above, an illusion. The system has neither explicit nor tacit knowledge of the hard rules. It is not hard to see why. Quite clearly, it does not explicitly represent Ohm's law to itself. There is, for example, no neat sub-pattern of units which can be seen to stand for the general idea of Resistance which figures in Ohm's law. Instead, sets of units stand for Resistance-at-R_1, and other sets for Resistance-at-R_2. In more complex networks, the coalitions of units which, when active, stand in for a top (or conceptual) level concept like resistance are highly *context-sensitive*. That is, they vary according to context of occurrence. Thus, to use Smolensky's own example, the representation of *coffee* in such a network would not comprise a single recurrent syntactic item but a coalition of smaller items (microfeatures) which shift according to context. Coffee in the context of cup may be represented by a coalition which includes (liquid) (contacting-porcelain). Coffee in the context of jar may include (granule) (contacting-glass). There is thus only an 'approximate equivalence of the "coffee vectors" across contexts' unlike the 'exact equivalence of the coffee tokens across different contexts in a symbolic processing system' (Smolensky [1988], p. 17). By thus replacing the conceptual level symbol 'coffee' with a shifting coalition of microfeatures, the so-called 'dimension shift', such systems deprive

themselves of the structured mental representations which are deployed both in a classical *competence* theory *and* in a classical symbol processing (*level 2*) account. Likewise, there is no stable representational entity in the simple network described which stands for Resistance (just as in the infamous past-tense network there is no stable, recurrent entity which stands for 'verb-stem' (see Rumelhart and McClelland [1986b], Pinker and Prince [1988], Clark [1989]). The immediate result is that there can be no explicit representation of rules which involve reference to the conceptual level constructs. The lack of *tacit* representation is almost immediate, since the processing can hardly be sensitive to structures which are not there.

To put the point in our favoured terms, the system cannot be said tacitly to represent the rules since there is no causal common factor in its problem solving such that whenever, *e.g.*, Ohm's law would be cited in the competence theory, that single factor is pivotal in the processing which yields the actual result. To see this we need only reflect that different feature units and knowledge atoms will be pivotal in solving problems which relate to the fate of R_1 and ones which relate to the fate of R_2. In this (restricted) sense it *does* have something in common with the look-up tree. For the network fails to embody strict tacit knowledge of the rule because it fails to route all its actual processing through a causal bottleneck corresponding to the derivational bottleneck marked by the repeated citing of Ohm's law. By having multiple causal routes where the competence theory has a single derivational equation, the network loses its claim to strict tacit knowledge of the rule. In that respect, it fails to embody tacit knowledge of the rule for the same reason as does the look-up tree.

Now for the quibble promised earlier. In adopting, as far as I understand it, Davies' characterization of tacit knowledge, I am uneasy about the use of the phrase 'causal common factor'. It has the advantage of making neuropsychological implications seem very immediate. But it may paper over some of the complexities of stacked virtual machines. For my guess is that what would need to be common for the classical cascade to be realized, is *not* a simple physical state so must as a state of *the virtual machine over which the processing story is defined.* After all, even a classical system, courtesy of various niceties of operating systems, may not use the same *physical* state every time it goes through a processing transition marked (in the competence theory) by Ohm's law. However, the level 2 processing description need not (and ought not) signal the difference, since it has no implications as far as the actual algorithm is concerned. It is merely an implementation detail. Contrariwise, the variety of states which, in a connectionist story, may correspond to a single symbolic transition, *must* be signalled in the processing/algorithmic description. After all, the system's real knowledge *is* the knowledge so encoded—a fact which is directly responsible for the much-vaunted fluidity and context-sensitivity of connectionist processing. I am not sure how much of a difference this makes

since virtual machines, as much as real ones, can exhibit distinctive breakdown patterns and hence tie in with the cognitive neuropsychology.

Quibbling aside, we are now in a position to sum up the Newtonian attitude to competence theorizing. A Newtonian connectionist will regard a competence theory as *descriptive* (perhaps at a quite fine grain—recall the discussion of 'macrodecisions') of the course of processing. But she will not regard it as *suggestive* of the actual processing involved. It is not suggestive because the behaviour is not dependent on the system's having explicit or tacit knowledge of the symbolic derivation rules; a fact evidenced in its behaviour outside the idealized, 'Newtonian' domain of well-posed problems and unlimited processing time. This behaviour shows that 'it's really been a "quantum" system all along' (Smolensky [1988], p. 19).

In a revealing footnote (Smolensky [1986], p. 246) the point is cast in terms highly appropriate to our discussion. The characterization of competence as a set of derivation rules applied to a symbol system can be viewed, Smolensky suggests, as providing a *grammar* for generating the high-harmony (=maximal soft constraint satisfaction) states of a system. Thus a competence theory emerges as a body of laws which serve to pick out the states into which the system will settle in certain ideal conditions. This, then, is the full Newtonian attitude to a competence theory: a competence theory is a kind of grammar which fixes on certain stable states of the system. As such it is, in a central range of cases, descriptively adequate. But it does not reveal what Smolensky calls the *dynamics*, or actual processing strategies, of the system. It is not a properly suggestive guide to the level 2 processing story. For the Newtonian then, competence theorizing just ain't what it used to be.

5 ROGUE COMPETENCE

On the Newtonian connectionist model, then, the competence theory functions as a descriptively adequate guide to the output in a somewhat idealized range of cases. This, however, is not the only understanding of competence theories available to a connectionist. And indeed, it is not the understanding implicit in some *other* connectionist treatments of high level problem solving. In this Section I look at a class of alternative treatments which I shall call *rogue* models of competence.

The basic difference between Newtonian and rogue models is simply this. In a Newtonian model, the connectionist network is *itself* capable, under idealized conditions, of behaving in all the ways specified by the competence theory. In a rogue model, by contrast, the basic connectionist network does not *itself* have the capacity (even under idealizations of processing time and well-posed problems) to produce the full range of results required by (*i.e.* derivable in) the competence theory. Instead, it will be claimed that insofar as human beings actually exhibit the full scale classical competence they do so only by deploying

other resources (for example, a linked symbol processor *or* real world structures (like pen and paper) for manipulating symbols). The view of competence models which emerges from a rogue approach is thus that they involve pressing into service extra resources which are not on-line in fast daily problem-solving in the domain.

An example of a rogue model can be found in Rumelhart, Smolensky, McClelland, and Hinton [1986]. The example concerns our capacity to multiply numbers. We might imagine a symbolic competence theory here appealing to the laws of arithmetic. But a basic connectionist model will not resemble such a symbolic store. Rather, it will amount to a well trained pattern matcher which can 'see' the results of some multiplications right away. For example, most of us can 'see' the answer to 7×7, but not to 7984×5431. How then, do we solve the latter kind of problem?

The conjecture is that:

> The answer comes from our ability to create artifacts—that is, our ability to create physical representations that we can manipulate in simple ways to get answers to very difficult and abstract problems. (Rumelhart, Smolensky, McClelland, and Hinton [1986] pp. 44).

Thus, to solve 7984×5431 we might write down the question and *then* solve it by the careful deployment of a series of the simple pattern-matching steps we are good at, *e.g.* beginning by multiplying 4×1 and so on:

$$\begin{array}{r} 7\ 9\ 8\ 4 \\ 5\ 4\ 3\ 1 \\ \hline . \ . \ . \ . \ 4 \\ \quad . \ . \ . \ . \end{array}$$

We may, they go on to say, even learn to do this *in our head* by representing the external symbols to ourselves in some manner. But it is still an essentially 'external' symbolic medium which we are manipulating, and it still constitutes a resource built *on top of* the basic connectionist pattern-matching capacity which we deploy. (Daniel Dennett has recently being saying very similar things about the cases where *sentences* seem to run through our heads. In these case, we do indeed do classical symbol processing. But such processing may constitute an extra resource, not implicated in all our daily, non-linguistic reasoning—see Dennett [1987], pp. 233, 114–15; also Clark [1988].)

The account of complex multiplication is of course highly problematic since the whole thing *seems* to involve knowing symbolic rules governing the serial deployment of the pattern-matching capacities! But we have seen already that much *apparently* symbol-reliant behaviour may be sub-symbolically produced. (But see Clark [1989] for a detailed discussion.) And at any rate, I use the example merely as a gesture at the *kind* of account which would constitute a rogue model.

To give one final example (which I owe to Martin Davies), consider our capacity to parse garden-path sentences like:

the horse raced past the barn fell.

A rogue model of parsing might go something like this. We have on-line a quick and dirty connectionist network which can parse most of the sentences we encounter in daily speech. But it does not have the capacity (even in principle, subject to idealization) to parse a garden-path sentence. However, we *also* have (not on-line, but in the background) a classical symbolic parser (something like an ATN?) which can parse such cases. And when the quick and dirty network fails, this back-up comes on-line to save the day. This fits the phenomenology, in which the sentence at first looks like nonsense, then falls into place. In such a case the classical competence theory correctly describes the structure of the *back-up system*. But it does not describe the on-line network. If, in addition, we imagine that the classical back-up system was active in training up the network, the partial confluence of the two systems over a range of simple cases is rendered unsurprising.

An obvious and related advantage of the rogue approach concerns the psychological plausibility of so-called supervised learning algorithms. These are procedures for training connectionist networks which rely on the back propagation of error messages, and hence rely on a *teacher* (usually a conventional computer) which looks at the system's output and tells it what the output *should* have been like. (For a little more detail, see the discussion of *NETtalk* in Section 6.) Such set-ups have often appeared deeply psychologically unrealistic. For example, when we learn a language, we can do so by being given *positive* examples only (as Chomskians are fond of pointing out). Whence, then, the teacher and the error messages?

The possibility which rogue models open up is that a separate system stores a set of input-output pairings (*e.g.* a set of observed print-phoneme pairings) and uses these to train a connectionist network. The negative instances are thus generated and spotted by the brain itself, rather than by other agents. Terence Sejnowski has recently endorsed such a picture and illustrates it by citing the case of the White Crown Sparrow which hears its father's song one year but does not sing it until the next. The hypothesis is that the bird somehow *stores* the song, but must train up a network to reproduce it—a process which explains the long gap between exposure and reproduction. White Crown Sparrows aside, rogue approaches clearly offer the best hope for the psychological respectability of the back propagation method of connectionist learning.

At its most extreme, a rogue model may divorce human on-line processing from the strict competence model, but *reinstate* the classical competence as a full and proper description of a back-up system. Note that the status of the classical competence theory on a rogue model is quite different from its status

on a Newtonian one. For the rogue modeller, the classical competence theory properly describes an important, though not constantly on-line, class of processing systems. In fact, the importance of these classical resources is, I suspect, not yet fully appreciated even where lip service is paid to their presence. Thus Smolensky [forthcoming] introduces the idea of language as a special medium of knowledge transmission involving processing by a classical virtual machine called the Conscious Rule Interpreter. But the role of linguistic instruction is still presented as somewhat second-grade. Language allows us to formulate rules which, for example, help the novice in the early stages of training (see also Smolensky [1986], pp. 251–2, where essentially the same picture is applied to the previously discussed case of electric circuit problem solving). The *expert*, however, is pictured as using a powerful connectionist network, and seems to need language only to transmit potted elements of her insights to others. This may severely under-estimate the contribution of symbol processing. Such processing may also help the expert to understand and extend her own skills by providing a kind of meta-reflection on her own on-line reasoning. (For some related hypotheses see Karmiloff-Smith [1987] and Dennett [1988].)

The most potent effect of the adoption of a rogue approach is vastly to complicate the currently fashionable debate concerning the 'correct' cognitive architecture of mind (see Fodor and Pylyshyn [1988]). For if a rogue model is adopted, there is no unique answer to such questions. Any good account of human cognitive skills will need to employ *both* kinds of model, and the classical version will not be just a convenient approximation. It is as if the physical world turned out to be Newtonian in some areas and quantum in others, rather than being uniformly quantum-describable but in some circumstances *looking* Newtonian.

To sum up, rogue models deny even the *descriptive* adequacy of classical competence models to on-line processing. But they allow that the classical theory is both descriptive *and* suggestive of the processing of an *additional resource system.* This additional resource system guarantees what might be called our *canonical* reasoning abilities in a given domain. In rogue cases, the competence model *is* what it used to be (an accurate description of *some* processing strategy), but it is not *where* it used to be—for it does not describe the computational form of daily on-line processing.

6 THE METHODOLOGY OF CONNECTIONIST EXPLANATION

Connectionist explanatory strategies, it seems, *cannot* fit into the mould suggested by Newell and Simon. A connectionist cannot begin with a Newell and Simon style competence theory and then simply implement it in a level 2 algorithmic model. The reason, we saw, is straightforward. Such a competence theory consists of a set of transition rules defined to apply to standard symbolic

representations or data-structures. In a classical model, these data-structures are explicitly represented in the machine (classical functional architectures are *precisely* those architectures which make this possible). And the machine then manipulates them in accordance with the rules (which need not *themselves* be explicitly tokened in any such data-structures). In a distinctively connectionist model, by contrast, there will be nothing which neatly corresponds to the classical symbolic data-structures. Instead, context-sensitive, shifting coalitions of units will correspond to single classical representations. This is the dimension-shift described earlier. Since there are thus no neat analogues to the classical symbolic structures, the system *cannot* (not even tacitly) embody knowledge of transition rules defined over *those very structures.* So a classical competence theory cannot be richly suggestive of a connectionist level 2 processing story. If it were, then the 'connectionist' system would amount merely to a fast, robust, implementation of a *classical* cognitive model (see Fodor and Pylyshyn [1988]).

Given all this, we saw that the devout connectionist could adopt one of two positions regarding the classical competence model. These were the Newtonian and Rogue positions discussed above. But a deeper, foundational issue remains unresolved. For the Newtonian and Rogue positions are united in denying that any top level classical competence theory can be richly suggestive of the level 2 processing strategies of the central on-line connectionist network which carries out a given cognitive task. But this (recall Section 2 above) now looks to be a doubly embarrassing loss. For the classical competence theory performed two tasks. First, it figured in a picture of the proper form of investigations in cognitive science (*i.e.* delineate the task at the level of competence theorizing and then write algorithms to carry it out). And second, it figured in a picture of what *explanation* in cognitive science involved. Just having a working program was not, in itself, to be regarded as having an explanation of how we perform a given cognitive task. Rather, we wanted some high-level understanding of what constraints the program was meeting and why they had to be met—an understanding naturally provided by giving the top level competence theory which a given *class* of programs could be seen to implement. The unavailability of the classical competence theory thus threatens to render connectionist models *non-explanatory* in a very deep sense. And it leaves the actual *methodology* of connectionist investigations obscure.

As a brief illustration of the problem, consider an example of Good Old Fashioned Explanation In Cognitive Science (GOFEICS—apologies to John Haugeland). Take Naive Physics. Naive Physics, as everyone knows, is the attempt to discover the knowledge which enables a mobile, embodied being to negotiate its way around a complex physical universe. A well-known instance of this general project is Hayes' [1984] work on the naive physics of liquids. This involved trying to compile a 'taxonomy of the possible states liquid can be in' and formulating a set of rules concerning movement, change and liquid

geometry. The final theory included specifications of fifteen states of liquid and 74 numbered rules or axioms written out in predicate calculus. This amounts to a detailed competence specification which might eventually be given full level 2 algorithmic form. Indeed, Hayes ([1985], p. 3) is quite explicit about the high level of the investigative project, insisting that it is a mistake to seek a working program too soon. The explanatory strategy of naive physics is thus a paradigm example of the official classical methodology recommended by Newell and Simon. First, seek a high level competence theory involving symbolic representations and a set of state transition rules. Then write level 2 algorithms implementing the competence theory, secure in the knowledge that we have a precise higher level understanding of the requirements which the algorithms meet and hence a real grasp of why they are capable of carrying out the task in question. It is this security which the connectionist lacks, since she does not (*cannot*) proceed by formulating a detailed classical competence theory and then neatly implementing it on a classical symbol processing architecture.

Hence the problem: how *should* the connectionist proceed, and what constitutes the higher level understanding of the processing which we need in order to claim to have really *explained* how a task is performed? What is needed, it seems, is some kind of connectionist analogue to the classical competence theoretic level of explanation.

I believe that such an analogue exists. But it remains invisible until we perform a kind of Copernican revolution in our picture of explanation in Cognitive Science. For the connectionist effectively inverts the usual temporal and methodological order of explanation, much as Copernicus inverted the usual astronomical model of the day by having the earth revolve around the sun instead of the other way round. Likewise, in connectionist theorizing, the high level understanding will be made to revolve around a working program which has learnt how to negotiate some cognitive terrain. This inverts the official Marr-style ordering in which the high level understanding (*i.e.* competence theory) comes first and closely guides the search for algorithms. To make this clear, and to see how the connectionist's high level theory will depart from the form of a classical competence theory, I propose to take a look at Sejnowski's NETtalk project.

NETtalk is a large, distributed connectionist model which aims to investigate part of the process of turning written input (*i.e.* words) into phonemic output (*i.e.* sounds or speech). The network architecture comprises a set of input units which are stimulated by seven letters of text at a time, a set of hidden units, and a set of output units which code for phonemes. The output is fed into a voice synthesizer which produces the actual speech sounds.

The network began with a random distribution of hidden unit weights and connections (within chosen parameters), *i.e.* it had no 'idea' of any rules of text to phoneme conversion. Its task was to learn, by repeated exposure to training

instances, to negotiate its way around this particularly tricky cognitive domain (tricky because of irregularities, sub-regularities and context-sensitivity of text→phoneme conversion). And learning proceeded in the standard way, *i.e.* by a back-propagation learning rule. This works by giving the system an input, checking (this is done automatically by a computerized 'supervisor') its output, and telling it what output (*i.e* what phonemic code) it *should* have produced. The learning rule then causes the system to minutely adjust the weights on the hidden units in a way which would tend towards the correct output. This procedure is repeated many thousands of times. Uncannily, the system slowly and audibly learns to pronounce English text, moving from babble to half-recognizable words and on to a highly creditable final performance. (For a full account, see Rosenberg and Sejnowski [1987] and Sejnowski and Rosenberg [1986].)

Consider now the methodology of the NETtalk project. It begins, to be sure, by invoking the results of some fairly rich prior analysis of the domain. This is reflected in the author's choice of input representation (*e.g.* the choice of a seven letter window, and a certain coding for letters and punctuation), in the choice of output representation (the coding for phonemes) and in the choice of hidden unit architecture (*e.g.* the number of hidden units) and learning rule. These choices highlight the continued importance of some degree of prior task analysis in connectionist modelling. But they are a far cry from any fully articulated competence theory of text to phoneme conversion. For what is noticeably lacking is any set of special purpose state transition rules defined over the input and output representations. Instead, the system will be set the task of learning a set of weights over its hidden units such that the weights perform the task of mediating the desired state transitions. For this reason I shall characterize the connectionist as beginning her investigations with a level ·5 'task analysis', as opposed to a level 1 (or 1·5) competence theory. It is worth remarking, however, that the level ·5 specification, though *less* than a full-blown symbolic competence theory, may *still* embody a psychologically unrealistic amount of prior information. For when a human learns to perform a task she does not know, in advance, how many hidden units to allocate (too many and you form an uninformative 'look-up tree', too few and you fail to deal with the data) or the best way to represent the solution. In this sense, the level ·5 specification may be doing more of the problem solving work than some connectionists would like to admit. For present purposes, however, the point is just that the level ·5 model forms the basis upon which, courtesy of the powerful connectionist learning rules, the system comes to be able (after much training) to negotiate the targetted cognitive terrain. At this point, the connectionist has in her hand a working system—a full-scale level 3 implementation.

Suppose we were to stop there. We would have a useful toy, but very little in the way of increased understanding of the phenomenon of text-phoneme

conversion. But, of course, the connectionist *does not* stop there. From the up and running level 3 implementation she must now work *backwards* to a higher-level understanding of the task. This is Marr-through-the-looking-glass. How is this higher level understanding to be obtained? There are a variety of strategies in use and many more to be discovered. I shall mention just three. First, there is simple *watching*, but at a microscopic level. Given a particular input, the connectionist can see the patterns of unit activity (in the hidden units) which result. (This, at any rate, will be the case if the network is simulated on a conventional machine which can keep a record of such activity). This, as Sejnowski points out, provides a kind of data which neuroscientists are hard pressed to gather. For neuroscience has excellent techniques for recording single cell activity. But it is not well placed to record patterns of simultaneous activity across large numbers of cells. (See also Churchland [forthcoming—1989].)

Second, there is *network pathology*. While it is obviously unethical deliberately to damage human brains to help us see what role sub-assemblies of cells play in various tasks, it seems far more acceptable to damage artificial neural networks.

Lastly, and perhaps most significantly, the connectionist can generate a picture of the way in which the system has learnt to divide up the cognitive space it is trying to negotiate. It is this picture, given by so-called 'hierarchical cluster analysis', which seems to me to offer the closest connectionist analogue to a high-level, competence-theoretic understanding.

Cluster analysis is an attempt to answer the question, 'What kinds of representation have become encoded in the network's hidden units?' This is a hard question since the representations, as noted earlier, will in general be of somewhat complex, unobvious, dimension-shifted features. To see how cluster analysis works, consider the task of the network to be that of setting hidden unit weights in a way which will enable it to perform a kind of set partitioning. The goal is for the hidden units to respond in distinctive ways when, and only when, the input is such as to deserve a distinctive output. Thus in text-to-phoneme conversion, we want the hidden units to perform very differently when given 'the' as input than they would if given 'sail' as input. But we want them to perform *identically* if given 'sail' and 'sale' as inputs. So the hidden units' task is to partition a space (defined by the number of such units and their possible levels of activation) in a way which is geared to the job in hand. A very simple system, such as the rock/mine network described in Churchland [forthcoming—1989] may need only to partition the space defined by its hidden units into two major subvolumes—one distinctive pattern for inputs signifying mines and one for those signifying rocks. The complexities of text-phoneme conversion being what they are, NETtalk must partition its hidden unit space more subtly (in fact, into a distinctive pattern for each of 79 possible letter to phoneme pairings). Cluster analysis, as carried out by Rosenberg and

Sejnowski [1987] in effect constructs a hierarchy of partitions on top of this base level of 79 distinctive stable patterns of hidden unit activation. The hierarchy is constructed by taking each of the 79 patterns and pairing it with its closest neighbour, *i.e.* with the pattern which has most in common with it. These pairings act as the building blocks for the next stage of analysis, in which an average activation profile (between the members of the original pair) is calculated and paired with *its* nearest neighbour drawn from the pool of secondary figures generated by averaging each of the original pairs. The process is repeated until the final pair is generated. This represents the grossest division of the hidden unit space which the network learnt—a division, which in the case of NETtalk turned out to correspond to the division between vowels and consonants (see Figure 2).

Cluster analysis thus provides a kind of picture of the shape of the space of the possible hidden unit activations which power the network's performance. By reflecting on the various aspects of this space (*i.e.* the various clusterings) the theorist can hope to obtain some insight into what the system is doing. It may, for example, turn out to be highly sensitive to some sub-regularity which had hitherto been unnoticed or considered unimportant. It is as if we are provided with a tracing of the shape of the cognitive space we are attempting to understand. The tracing must be interpreted and that is a real and at times difficult task. But it is not shooting in the dark, for we can see what inputs are associated with what configuration (even if it is a higher level configuration revealed by cluster analysis).

We are thus given members of each class in question—the task is then to find perspicuous, conceptual level terms in which to describe the conditions of class membership.

A fully interpreted cluster analysis, I would like to suggest, constitutes the nearest connectionist analogue to a classical competence theory. Like a competence theory, it provides a level of understanding which is higher than (*i.e.* more general than) the algorithmic level. For the 'algorithmic' specification, for a connectionist, must be a specification of (a) the network configuration and (b) the unit rules and the connection strengths. But there is a many–one mapping between such algorithmic specifications and a particular cluster analysis. For example, a network which started out with a different random set of weights would, after training, exhibit the *same* partitioning profile (hence have an identical cluster analysis) but do so using a very different set of individual weights. Unlike a classical competence theory, however, the cluster analysis will typically *not* look like a set of state transition rules defined over conceptual level entities. Instead, it will be more like a kind of geometric picture of the shape of a piece of cognitive terrain. Those theorists who think that a high level explanation must be like a set of sentences and rules may find this hard to adjust to.

On the other hand some radically anti-sentential theorists (*e.g.* Churchland

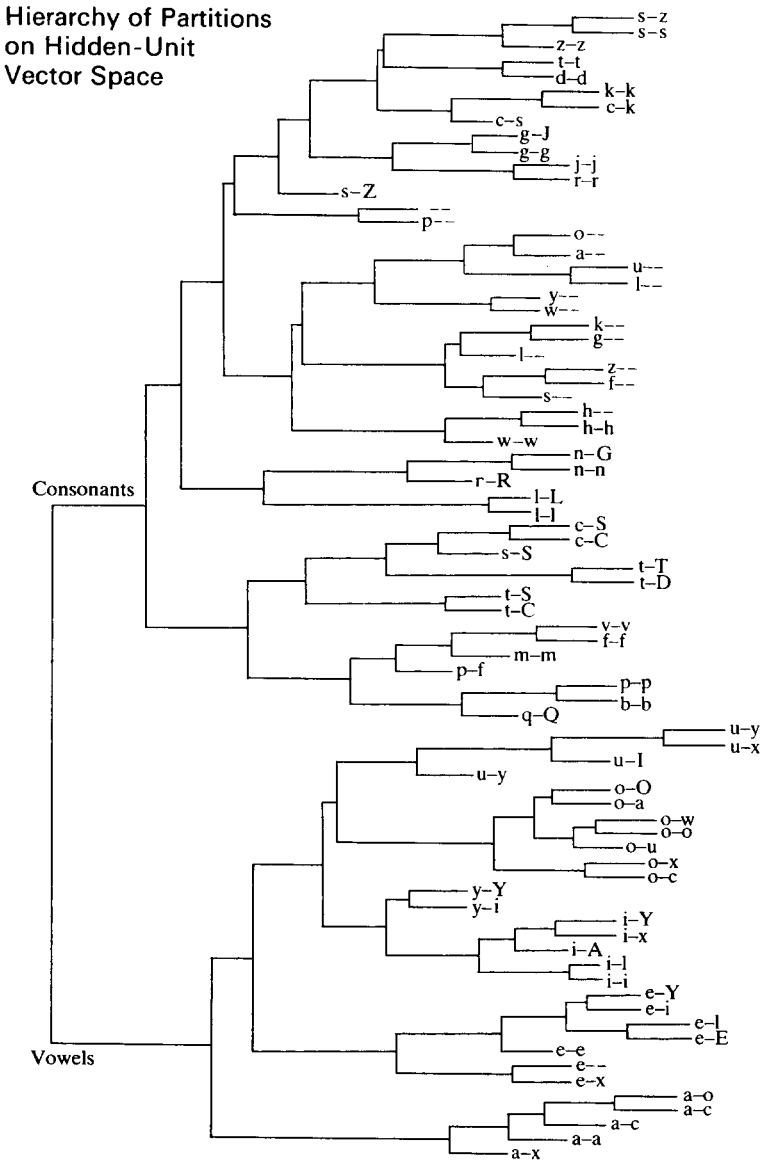

FIGURE 2 The results of a cluster analysis of NETtalk (from Churchland [forthcoming—1989], after Rosenberg and Sejnowski [1987]).

[forthcoming—1989]) may consider that an interpreted cluster analysis gives away *too much* to ordinary propositional discourse. Churchland argues that the correct level of understanding lies at the level of the connection weights. For, he insists, those are all the system 'really' knows about; it has no representation of its own partitionings. Moreover, the way two systems learn given new inputs can vary even if they have identical cluster analyses at time t_1. For the connection weights (which, we saw, stand in many–one relations to cluster analyses) are the pivotal unit of cognitive evolution.

This, however, looks like the ordinary swings and roundabouts of high level explanation. In opting at times for a level of analysis which groups particular connection weight specifications into equivalence classes governed by common cluster analyses we naturally trade specificity for generality. Just as pure Darwinism leaves recessive characteristics unexplained, but highlights general principles covering a class of evolutionary mechanisms, so cluster analysis leaves some details of cognitive evolution unexplained but highlights the gross sensitivity which enables a class of networks to negotiate successfully a given cognitive terrain. Some such high level understanding seems essential if connectionism is to be deeply explanatory of cognitive performance. A mere specification of a set of connection weights is surely not an *explanation*, even for the anti-sententialist.

The main point I want to stress is, however, independent of any view about the merits or demerits of cluster analysis. It concerns the methodological inversion of traditional cognitive science. The connectionist, by whatever means, achieves her high level understanding of a cognitive task by reflecting on, and tinkering with, a network which has *learnt* to perform the task in question. Unlike the classical, Marr-inspired theorist, she does not begin with a well worked out (sentential, symbolic) competence theory and then give it algorithmic flesh. Instead, she begins at level ·5, trains a network, and then seeks to grasp the high level principles it has come to embody. This is an almost miraculous boon for cognitive science. For the discipline has been dogged by the (related) evils of ad-hocery and sententialism. Forced to formulate competence theories as sets of rules defined over classical, symbolic data structures, theorists have plucked principles out of thin air to help organize their work. Connectionist methodology, by contrast, allows the task demands to trace themselves and thus suggest the shape of the space in a way uncontaminated by the demands of standard symbolic formulation. We thus avoid imposing the form of our conscious, sentential thought on our models of unconscious processing—an imposition which was generally as practically unsuccessful as it was evolutionarily bizarre.

In sum, the connectionist, in being compelled to make do without the comfort of a classical competence theory is deprived neither of high-level explanatory power nor of methodological soundness. On the contrary, the methodology of connectionist explanation is perfectly geared to the avoidance

of ad-hoc organizing principles and sentential, linguistic bias. There remain important and unresolved questions concerning the best ways to extract and couch such high level explanations as connectionism may provide. But techniques such as cluster analysis, network pathology and activation recording are already being developed and will no doubt become well-understood. Once they do, the Copernican revolution in cognitive explanation will be well under way.

7 CONCLUSIONS: THE CASCADE, THE DAM AND THE DIVIDED STREAM

Classicists and Connectionists, it seems, must differ fundamentally in the way they expect actual processing (level 2) models to relate to traditional competence theories. The paper began by displaying the classicist vision of this relation and two connectionist alternatives. These may conveniently be pictured as follows.

Relation one: the cascade

Dennett describes the classicists' vision as one of a 'triumphant cascade through Marr's three levels' (Dennett [1987], p. 227). The cascade flows easily given the presence of a classical symbol processing architecture. The axioms or rules of the competence theory are linguistically expressed formulas for deriving one symbol from another. Various algorithms (level 2) may then implement that derivational structure. They may do so explicitly (by tokening the rule) or tacitly (by processing explicit symbol strings in accordance with the rule). In this classical vision, level 2 is a neat echo of level 1.

Relation two: the dam

Newtonian connectionism dams the classical cascade by introducing a dimension shift between the items (symbol strings) operated on by the level 1 derivational rules and the items (subsymbols) 'operated on' by a connectionist network. The level 1 theory may describe some (idealized) aspects of the network's behaviour. But the network embodies neither explicit nor tacit knowledge of the derivational rules nor the conceptual level structures over which they are defined.

Relation three: the divided stream

Rogue models represent a more complex state of affairs in which actual performance is dependent on *two* systems. One, the daily, on-line system, relates to the competence theory in the way described by the Newtonian

connectionist, *i.e.* it matches some of the implied behaviour, but without embodying the classical knowledge. The other is an additional resource, perhaps created by the exploitation of external symbols, which simulated a classical machine. As such, it is capable of embodying the derivational rules and conceptual level structures specified in the competence theory. Rogue models complicate the debate over the 'correct' architecture of cognition by suggesting a multiplicity of interactive (virtual) architectures.

One way or another, then, the connectionist must distance herself from the details of the classical competence model. Such models are not properly suggestive of the form of on-line connectionist processing, though they may be either descriptive of (a subset of) the results of such processing, or descriptive of some other cognitive resource. But this dislocation of connectionism and competence theorizing raised a serious problem. For the classicist had a methodology which guaranteed a useful and accurate higher level understanding of the cognitive phenomenon modelled. The connectionist, by contrast, may seem to have working systems but no higher level understanding of them—hence, in a certain sense, no *explanations* of cognitive phenomena.

This worry loses some of its force once we manage to perform a kind of Copernican revolution in our thinking about explanation in cognitive science. Under Marr's influence, Cognitive Scientists are likely to expect some high level understanding of a task to *precede* and *inform* the writing of algorithms. Classical competence theoretic specifications aim to do just that job. The connectionist, however, effectively inverts this strategy. She begins with a minimal understanding of the task, trains a network to perform it, and *then* seeks, in various principled ways, to achieve a higher-level understanding of what it is doing and why. This may involve careful recording of network activity, the examination of the network's behaviour after various forms of damage, and plotting the way the network's hidden units divide up the cognitive space they are negotiating. This last activity (cluster analysis, as described in the text) clearly provides a kind of higher level understanding since there is a many–one relation between a given cluster analysis and the set of connection weights which could implement it. The connectionist starts with a level ·5 model, moves rapidly to a level 3 implementation and must then work backwards to detailed higher levels of understanding.

This explanatory inversion, I want to suggest, actually constitutes one of the major *advantages* of the connectionist approach over traditional cognitive science. It is an advantage because it provides a means by which to avoid the *ad hoc* generation of axioms and principles. Instead of having to decide on a rather arbitrary set of symbolic, language-based axioms to organize some cognitive task (recall naive physics) the connectionist can let the task itself organize the network, and only *then* attempt to formulate various higher level pictures of its activity. Such pictures, moreover, may depart (in ways we have yet to fully

imagine) from the traditional picture of a theory as a set of propositions. Instead, they may be more geometric, or pictorial, or may use language in unexpected, apparently clumsy ways (see Churchland [forthcoming—1989]).

There is, we may finally conjecture, a fairly deep reason why attitudes to competence polarize connectionists and classicists. It is that a competence model is a traditional *theory*, expressed in propositional or logical form. Classicists believe that thinking just *is* the manipulation of items having propositional or logical form; connectionists insist that this is just the icing on the cake and that *thinking* ('deep' thinking, rather than just sentence rehearsal) depends on the manipulation of quite different kinds of structure. As a result, the classicist attempts to give a level 2 processing model which is defined over the very same kinds of structure as figure in her level 1 theory. Whereas the connectionist insists on dissolving that structure and replacing it with something quite different.

A curious irony emerges. In the early days of Artificial Intelligence, the rallying cry was 'Computers *do not* crunch numbers, they *manipulate symbols*!' This was meant to inspire a doubting public by showing how much computation was like thinking. Now the wheel has come full circle. The virtue of connectionist systems, it seems, is that 'they do not manipulate symbols, they *crunch numbers*'. And nowadays we all know (don't we?) that thinking *is not* mere symbol manipulation! So the wheel turns.

School of Cognitive & Computing Sciences
University of Sussex
Brighton

REFERENCES

CHOMSKY, N. [1986]: *Knowledge of Language: Its Nature, Origin and Use*, Praeger Publishers, Connecticut.

CHURCHLAND, P. [forthcoming—1989]: 'On the nature of theories: a neurocomputational perspective'. In P. M. Churchland (*ed.*) *The Neurocomputational Perspective*, MIT Press, Cambridge, Massachusetts.

CLARK, A. [1988]: 'Thoughts, sentences and cognitive science', *Philosophical Psychology*, Vol. I, no. 3, pp. 263–78.

CLARK, A. [1989]: *Microcognition: Philosophy, Cognitive Science and Parallel Distributed Processing*. MIT/Bradford Books, Cambridge, Massachusetts.

DAVIES, M. [1987]: 'Tacit knowledge and semantic theory: can a five per cent difference matter?', *Mind*, **96**, pp. 441–62.

DAVIES, M. [forthcoming]: 'Connectionism, modularity and tacit knowledge', *British Journal for the Philosophy of Science*.

DENNETT D. [1987]: *The Intentional Stance*. MIT/Bradford Books, Cambridge, Massachusetts.

DENNETT, D. [1988]: 'The evolution of consciousness', *Jacobsen Lecture*, University of London, May 1988. *Tufts University Current Circulating Manuscript* CCM-88-1.

DENNETT, D. [forthcoming]: 'Review of Psychosemantics', *Journal of Philosophy.*

FODOR, J. and PYLYSHYN, Z. [1988]: 'Connectionism and cognitive architecture', *Cognition*, 28, pp. 3–71.

HAYES, P. [1984]: 'Liquids' in *Formal Theories of the Commonsense World* ed. J. Hobbs (Ablex, Hillsdale, NJ, 1984).

KARMILOFF-SMITH, A. [1987]: 'Beyond modularity: a developmental perspective on human consciousness'. Transcript of talk given to the Annual Meeting of the British Psychological Society, Sussex, April 1987.

MARR, D. [1977]: 'Artificial Intelligence: a personal view', In J. Haugeland (*ed.*) *Mind Design.* MIT/Bradford Books, Cambridge, Massachusetts, 1981.

PEACOCKE, C. [1986]: 'Explanation in computational psychology: language, perception and level 1·5', *Mind and Language*, Vol. 1, No. 2, pp. 101–23.

PINKER, A. and PRINCE, S. [1988]: 'On language and connectionism', *Cognition* 28.

RIDLEY, M. [1985]: *The Problems of Evolution.* Oxford University Press, Oxford.

ROSENBERG, C. and SEJNOWSKI, T. [1987]: 'Parallel networks that learn to pronounce English text', *Complex Systems*, I, pp. 145–68.

RUMELHART, D. and MCCLELLAND, J. [1986]: 'PDP models and general issues in cognitive science'. In J. McClelland, D. Rumelhart and the PDP Research Group (*eds.*), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT/Bradford Books, Cambridge, Massachusetts, 1986, Vol. I, pp. 110–46.

RUMELHART, D. and MCCLELLAND, J. [1986b): 'On learning the past tenses of English verbs', In J. McClelland, D. Rumelhart and the PDP Research Group (*eds.*), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT/Bradford Books, Cambridge, Massachusetts, 1986, Vol. II, pp. 216–71.

RUMELHART, D., SMOLENSKY, P., MCCLELLAND, J., and HINTON, G. [1986]: Schemata and sequential thought processes in PDP models. In J. McClelland, D. Rumelhart and the PDP Research Group (*eds.*), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT/Bradford Books, 1986, Cambridge, Massachusetts, Vol. II, pp. 7–57.

SEJNOWSKI, T. and ROSENBERG, C. [1986]. 'NETtalk: a parallel network that learns to read aloud'. *Johns Hopkins University Electrical Engineering and Computer Science Technical Report.* JHU/EEC-86/01.

SMOLENSKY, P. [1986]: Information processing in dynamical systems: foundations of harmony theory. In J. McClelland, D. Rumelhart and the PDP Research Group (*eds.*), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* MIT/Bradford Books 1986, Cambridge, Massachusetts, Vol. I, pp. 194–281.

SMOLENSKY, P. [1988]: 'On the proper treatment of connectionism', *Behavioral and Brain Sciences*, 11, pp. 1–73.

SMOLENSKY, P. [1987]: 'The constituent structure of connectionist mental states', *Southern Journal of Philosophy*, Vol. XXVI, Supp. pp. 137–62.