The Recommendation Functional Architecture as the Basis for a Neurophysiological Understanding of Cognition

L. Andrew Coward
Nortel Networks, Canada
L-Andrew.Coward.cowarda@nt.com

Abstract

A definition of the scope of a scientific theory of human cognition is proposed in which for any psychological state a corresponding physiological state can be identified, and causal relationships between psychological states have corresponding causal relationships between physiological states. The vital role of a simple functional architecture in functionally complex commercial electronic systems is described, and it is argued that selection pressures have resulted in simple functional architectures in biological brains. However, the functional architecture is qualitatively different from the architectures in electronic systems. Electronic systems have the instruction architecture in which functional components exchange unambiguous information. The only alternative is the recommendation architecture in which functional components exchange ambiguous information. Systems with the recommendation architecture demonstrate phenomena with a striking similarity to psychological experiences such as learning, object recognition, associative memory, dream sleep without recall, constant sensory independent sequences of mental images, and individual differences between the experience of the same conditions. All of these phenomena can be described in a consistent fashion on both psychological and physiological levels. It is therefore argued that biological brains have the recommendation architecture, and that this architecture makes possible a scientific theory of cognition. The nature of representation in such an architecture is discussed.

Characteristics of a Neurophysiologically Based Theory of Cognition

A theory in the physical sciences establishes a correspondence between physical states at different levels of description detail in such a way that causal connections between states at the more detailed level exist whenever there are causal connections between the corresponding states at a higher level of description. For example, the Bardeen, Cooper and Schrieffer (1957) theory of superconductivity established descriptions of electrical current flow at normal and extremely low temperatures. At a high temperature these descriptions were in terms of metals, electrical current, and temperature. At a detailed quantum mechanical level the descriptions were in terms of ordered atomic structure with limited defects in the order, electron states which moved electrical charge, and energies. Causal connections at high level involved the decay of current with time (i.e. electrical resistance) at normal temperatures and the absence of such decay at low temperatures. The corresponding causal connections at the quantum mechanical level involved the movement of charge by individual electrons at normal temperatures and the scattering of individual electrons by defects blocking the movement of charge. Moving associations of electrons which can only form at low temperatures carry the charge at those temperatures, and these associations cannot be scattered by defects because such scattering would require enough energy to break up the association. Such a theory is regarded as highly successful in the physical sciences.

An analogous theory of cognition would need to propose psychological state descriptions X and corresponding physiological state descriptions x for which if a causal connection exists between states X_1 and X_2 then a causal connection also exists between the corresponding states x_1 and x_2. A psychological state description would include emotional, mental, perceptual and activity descriptions. An example might be simultaneously over a short period of time feeling mildly angry, reminiscing about some specific past event, and performing a task involving comparison of two visual images. The corresponding physiological state description would include the activation states of all neurons and the concentrations of all neurochemicals at all points, including dynamic variations, over the same time period.

Although x and X describe exactly the same state, the state description x contains much more information than X. If two states differ at the psychological level then the corresponding states at the physiological level must also differ. However, if two states differ at the physiological level it is possible that because of the much higher level of description information that the difference between the corresponding states at the psychological level may be too small to detect at that level. So although the perception of red and blue at the psychological level cannot correspond with the same physiological state, there may be multiple states at the physiological level which correspond with indistinguishably different perceptions of red at the psychological level.

Differences between similar experiences at the psychological level, for example between two individuals, must correspond with differences at the physiological level in a consistent fashion. When there is a difference between two experiences of the color red described at the psychological level, and those differences are described in a number of instances, then consistent differences must be observed in the corresponding physiological states. An individually unique feel at the psychological level must correlate consistently with individually unique physiological states.

If such correlations exist between physiological and psychological states, a further question is the degree to which some of these correlations can be regarded as representations. There are a number of different definitions of representation which could apply. One major issue, following Peschl and Riegler (1997), centers around the relationship between the state of external environmental reality R and the state of the mind *W*. The classical concept of representation is that $W = f(R)$. An alternative would be if $W = f(R, O)$ where O is the properties of the observer. For example, if R was the presence of an object of category dog, W could depend both on the presence of the dog and whether the mind was feeling aggressive, friendly or fearful towards the dog. Further variables could be the experience and immediate past states of the mind. A second issue is the consistency of the representation. This issue is whether the state W always reflects the presence of R, or is there is a probabilistic relationship in which W indicates R with some degree of probability. A third issue is the robustness of the correlation between W and R. If w is the physiological state corresponding with psychological state W, and small changes in w result in major changes in W, then w can be regarded as a symbol for R, but a very high degree of information integrity will be required at the physiological level to avoid confusion at the psychological level. High information integrity means that if errors in single data elements may correspond with significant differences at high level then there must be a low probability of such errors occurring. In computer systems this information integrity exists, sustained by parity checks and check sums etc. to detect and eliminate errors at the elementary device level. Such information integrity is implausible in a biological system.

Establishing the correlation between physiological and psychological states in practice is potentially a very complex undertaking, because of the disparity in information content referred to earlier. If multiple physiological states x_1, x_2, x_3, etc. and y_1, y_2, y_2 etc. could somehow be packaged into intermediate states x and y, and these intermediate states packaged into yet higher states and so on until it became possible to package a relatively small number of states into psychological states, then the correlation between physiological and psychological would be possible despite the disparity in information content. Such a hierarchy of states in fact exists in well designed electronic systems. The critical question is whether such a hierarchy exists in biological systems. The reason the hierarchy exists in electronic systems is that without such a hierarchy it is extremely difficult to build, repair or modify the system. The argument developed in the rest of this paper is that the needs to build from DNA, recover from damage, adjust to changing environmental conditions, and add features in the course of evolution has resulted in an analogous hierarchy in biological systems. However, the structure of the biological hierarchy is qualitatively different from any current electronic system.

The requirement for a functional architecture

Some currently operational commercial electronic products have billions of devices in a single system and perform very complex combinations of functions. For example, a large telecommunications

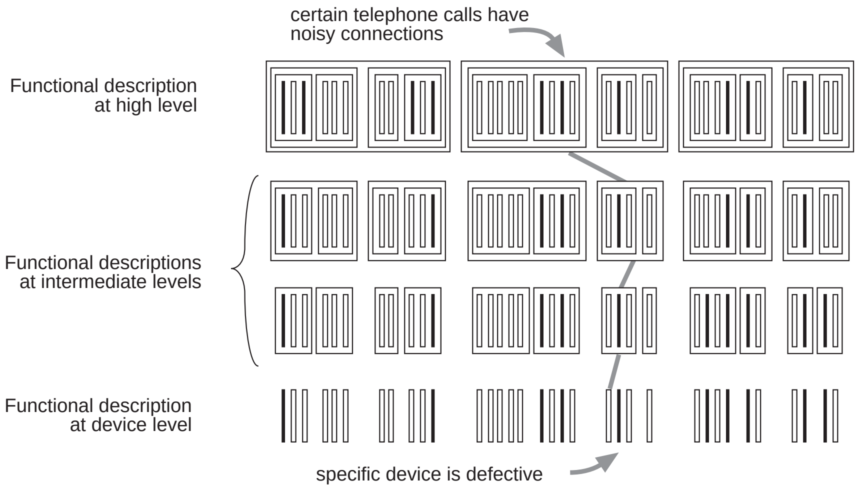

Figure 1: A functional hierarchy in which functionality at high level is separated into components. A component contains all the functionality of its subcomponents. The functionality at one level is precisely equivalent to the functionality at any other level, but with different description detail. The existence of this hierarchy makes it possible to relate system functionality to the operations of individual devices.

switch may contain over 4 thousand million transistor devices and provide telecommunications services to 100 thousand users. These services include actual voice and data services, billing, self diagnostics to ensure that the system is not totally out of service for more than two hours in forty years, and maintenance to adjust for service changes and additional capabilities. Provision of these services requires thousands of interacting functions, where "interacting" means that the functions must exchange information to be able to perform their independent functionality, and may act upon and change the system inputs available to other functions.

Any system which performs a complex combination of interacting functions using very large numbers of devices is forced to adopt a simple functional architecture. A functional architecture is illustrated in figure 1. It divides system functionality into functional components on many levels of detail. At the highest level, total system functionality is divided into major components. Each of these components is divided into subcomponents at the next level of detail, and so on all the way down to the operations of individual devices. Although a component at one level contains exactly the same functionality as its subcomponents, that functionality is defined in a simpler manner at the higher level. In other words, there is compression of the information in the description.

The reason a simple functional architecture is required is that without such an architecture it is not practical to build, repair or modify the system. For example, suppose that one device in the four billion fails. The system result might be connections from some telephones to some other telephones which are sometimes noisy. In order to repair the problem it must be possible to find some simple logical path which links this system deficit to an individual transistor or at least a small set of transistors. The use of software does not change this argument, it adds another dimension to the domain in which simple logical paths must exist. Any error in a construction process would immediately face the same problem. To understand the full issue for construction or modification, consider a system which was not created by design but by random selection and interconnection of devices until a working version in terms of system functions was found by trial and error. The first problem is that the only way to build a second copy is by duplicating the original device by device, connection by connection, there are no generic device selection and connection processes which can be repeated many times. The second problem would arise if it were necessary to modify the system functionality to add a feature or adjust to different environmental conditions. Identifying a small set of device changes which would make the desired functional change without undesirable side effects would be impossible.

For the required simple logical paths to be possible, there are two additional constraints on the functional architecture. The first is that all functional components on one level must perform roughly the same proportion of system functionality. In the opposite extreme, a functional architecture which at the highest level divided system functionality into two components which were the functionality of one transistor and the functionality of all the rest would have minimal value. The second constraint is that information exchange between components, although essential, must be minimized. If information exchange between two components were very high, it would be difficult to determine which of the two contained the defective subcomponent in the event of system failure.

Given functional components at every level which contain all the functionality of their subcomponents at the next more detailed level, compression of description information, roughly equal component size on any one level, and minimized information exchange between components, the result is a simple functional architecture in which system construction, repair, and modification are possible.

Although biological brains are not the result of an intellect driven design process, they are subject to very similar constraints. Copies of biological brains must be constructed from DNA "blueprints". Biological brains have some capability to recover from failures, for example the damage caused by strokes. Biological brains must be able to adjust to individual body differences resulting from growth differences and accidents. In the process of evolution, a random mutation must occasionally result in a useful functional change without catastrophic side effects. As a result, biological brains experience strong selection pressures in favor of simple functional architectures.

However, biological brains have minimal functional similarity to any current electronic system, and any biological functional architecture must therefore be radically different from conventional commercial architectures. It turns out that there are two qualitatively different types of functional architecture (Coward 1990, 1997) and in fact only two qualitatively different types are possible (Coward

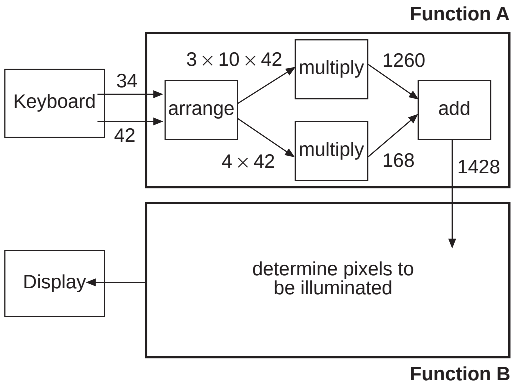

Figure 2: A simple calculator system which exchanges only unambiguous information between components. If the information 168 were communicated directly to function B it would be ambiguous to the recipient and could not be used to generate an unambiguous system command.

1998). The critical difference between them centers around the type of information which is exchanged internally between their functional components. Such information can be either unambiguous or ambiguous.

To understand the difference between the use of unambiguous and ambiguous information, consider the simple calculator system with two major functional components illustrated in figure 2. One functional component A receives two numbers from a keyboard and multiplies them together. The other component B receives information from the first component and determines which pixels on a display are illuminated. Assume that the first function uses a human like multiplication algorithm with different subcomponents choosing appropriate subproducts to calculate, calculating the results, and summing these subproduct results. In the illustration subproducts 168 and 1260 are summed to produce a total of 1428 which is communicated to the second component where it is used to determine the display. All the numbers exchanged in this example were unambiguous to the receiving component or subcomponent.

Consider the number 168. It was unambiguous when exchanged between subcomponents of function A, but it would be ambiguous if it were the only information communicated directly to function B because its meaning would depend on other information about function A which would not be avail-

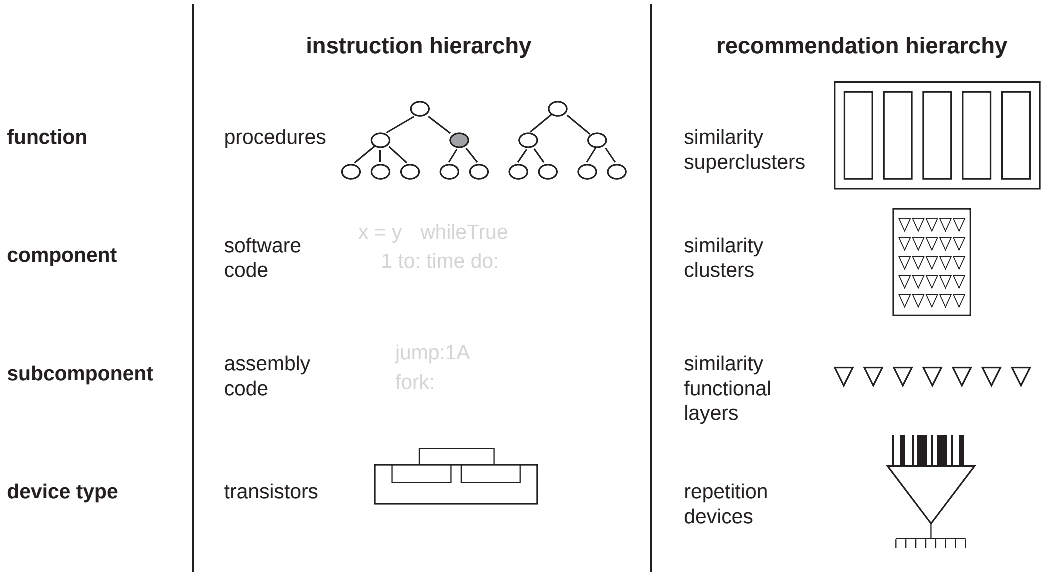

Figure 3: Functional hierarchies for alternative functional element paradigms. In the instruction architecture, instructions are combined into higher level instructions. In the recommendation architecture, repetitions are combined into clusters that generate recommendations. At each level a part of the higher level is shown, in the greater detail of the lower level.

able. It would not necessarily be meaningless. 168 could not occur as an intermediate product for most possible products. For example, there is no combination of two integers which if multiplied together generates 825 as the result with 168 as an intermediate result. But 168 is ambiguous because there are multiple products which could have 168 as an intermediate result: less than half of one percent of all numbers less than 10,000 for example, a small proportion but a significant set.

So a functional component which received and used a number like 168 could not issue an unambiguous system command, such as illuminate a specific combination of display pixels, but it could make recommendations. A system which used ambiguous information would need multiple functional components creating a range of alternative recommendations, and a competitive function to generate a high integrity system action.

Commercial systems always exchange unambiguous information between functional components. Such functional components detect patterns of unambiguous information and generate commands for system actions. Such components are called instructions, and in an instruction functional architecture detailed instructions are combined into higher level instructions through many levels of detail as illustrated in figure 3. Use of unambiguous information requires a reference location where such information is stored for access by any component. This reference location is called memory. Because while one location is using and perhaps changing an element of information, that information element is ambiguous to any other components, components can only operate sequentially, and for efficiency purposes the sequential operation takes place in a processor. The use of unambiguous information therefore forces the memory/processing separation and sequential execution ubiquitous in commercial systems. Parallel processing is only possible if unambiguous information can be partitioned into orthogonal sets with different components operating on different sets. In functionally complex systems this partitioning needs to be dynamic to accommodate changes in information requirements, and is extremely difficult to implement successfully.

In a system in which components exchange ambiguous information the functional components detect ambiguous repetitions of information conditions and generate recommendations. The outputs of higher level components represent the outputs of large subsets of their subcomponents. The recommendation functional hierarchy is therefore a hierarchy of repetition similarity, with sets of repetitions forming clusters, sets of clusters forming superclusters as illustrated in figure 3. If the information were unambiguous this would be a pattern, category, supercategory hierarchy (which in fact is one way to formally describe an instruction architecture). Because the information is ambiguous the clusters for example will correlate only partially with identifiable categories, and such categories may be cognitively very complex. For example, a cluster might correlate partially with "object which is bright and moving to the right" but would sometimes produce an output when the condition was not present or no output when the condition was present. Because the outputs are ambiguous recommendations, there must be a separate competitive function to generate high integrity behavior. The use of ambiguous information therefore results in a similarity clustering/competition separation radically different from the memory/processing separation in commercial electronic systems, and the sequential operation imposed by the use of unambiguous information is not present. This architectural separation is shown in figure 4.

Neural Networks

The distinction between clustering and competition in a recommendation architecture is in some ways analogous with the distinction between unsupervised and supervised learning in neural networks. The typical unsupervised learning algorithms such as adaptive resonance (Carpenter and Grossberg 1988) and Kohonen Nets (Kohonen1990) reduce input data to a smaller number of output types. However, they associate those outputs with features rather than ambiguous functionality, and they do not address how to create a multifunctional hierarchy with compression of the information content of descriptions between levels. As a result they have problems scaling to interacting functionality exchanging ambiguous information. For example, if a set of major features output from a Kohonen net are processed through a competitive function to generate functionality, that functionality cannot be integrated with other functionality except in a sequential instruction architecture.

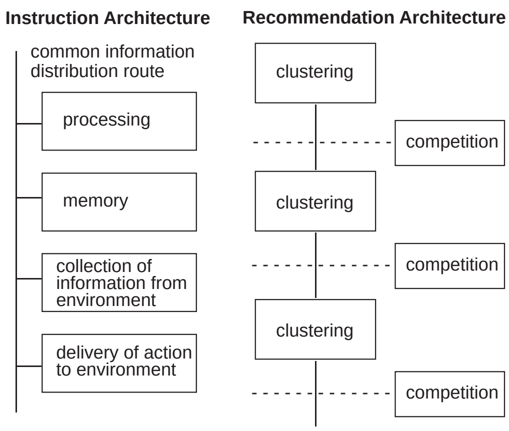

Figure 4: Comparison of the major functional separations in the two possible types of functional architecture.

One source of confusion in the discussion of whether neural networks can handle complexity is lack of clarity around the distinction between algorithmic and functional complexity. This critical distinction is illustrated in figure 5. In an algorithmically complex system, a high volume of input information is processed by a number of components which exchange information. Individual components perform complex algorithms on their inputs to generate outputs, and some of these outputs are system outputs. In a functionally complex system, the algorithms performed by individual components may be simple or complex. However, these components dynamically change the input information available to itself and all other components. To illustrate the difference, an algorithmically complex system might take inputs from a retina and generate outputs indicating the presence of different shapes. A functionally complex system might take inputs from a retina plus inputs indicating orientation of the retina and different components would generate outputs which produced shifts in the orientation of the retina and actions which changed the environment being perceived. The essential difference is that in a functionally complex system individual components produce outputs which change the information available to other components dynamically in the real time in which those other components are also generating their outputs. The output from one component therefore depends on whether other components have already acted on the external environment or the system itself and changed the information derived from those sources. When functionality is partitioned between components in

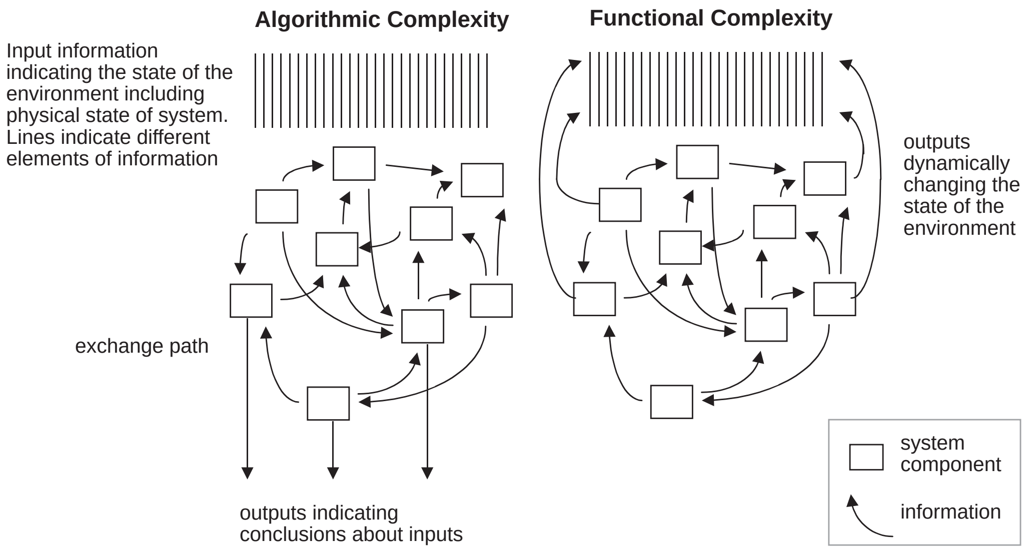

Figure 5: The difference between algorithmically and functionally complex systems. Both receive high volumes of information from an external source. In an algorithmically complex problem individual components perform algorithmically complex processes on their inputs from the external source and/or other components, but do not change the input information from the external source. A subset of components generate system outputs which are conclusions on the external input. In a functionally complex problem individual components perform processes which may be simple or complex on their inputs, but their outputs can dynamically change the external input information.

such a system the ability to get access to appropriate information in real time is a critical consideration in selecting the partitioning. If the problem to be solved is only algorithmically complex then the partitioning problem is reduced to performing the necessary calculations in the correct logical sequence. The information exchange issue is simpler and a functional architecture is not required to the same degree.

In practice actual systems will vary in their degree of functional complexity, but the greater the functional complexity the greater the requirement for a simple functional architecture. A system which performs algorithmically complex processes on a series of static information states is functionally extremely simple. Most problems solved using neural networks are of this type.

Learning: the Heuristic Definition of Functionality

In commercial instruction architectures, instructions themselves are handled as unambiguous information and recorded in memory. As a result it is difficult to build systems with complex functionality which can change their own functionality, or learn. However, learning is straightforward in a recommendation architecture. Such learning depends on an imprinting mechanism at the device level.

Suppose a device with a large number of physical inputs is presented with a combination of information in the form of the activation of a subset of its inputs. If a higher level functional signal (which would be the output of a functional group of devices as described below) is also present, the device produces an output in response to the active information, and in addition is programmed so that the device will produce output in the future if a similar information combination repeats, whether or not the higher level signal is present. In other words, the mechanism imprints an information combination which will be detected if it ever repeats. A very simple version of this algorithm would be if all inactive inputs were deleted. Such a device could only be programmed with one repetition. More biologically realistic versions with many repetitions per device can with the support of the extensive feedback connectivity observed in biological brains (Cauller

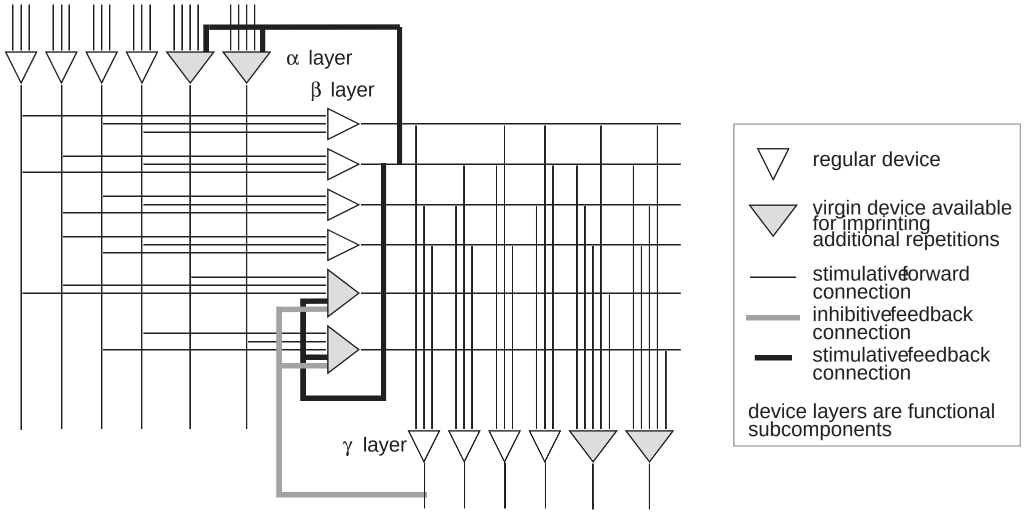

Figure 6: Connectivity of a simple repetition similarity cluster module using the repetition imprinting mechanism at the device level. The layers perform similarity subfunctions as discussed in the text. Single examples of connectivity which performs required layer to device functionality are given, realistic functionality requires many more connections as discussed in the text.

1995) generate the same functionality as the simple algorithm (Coward 1998).

At a somewhat higher functional level, all system input experiences can be sorted into repetition similarity clusters made up of devices which can record information combinations. A simple version of a cluster as illustrated in figure 6 might be made up of a set of devices α which can be programmed to imprint combinations of input information, a set of devices β which can be imprinted with combinations of outputs from α, and a set of devices γ which can be imprinted with combinations of outputs from β and which generate outputs from the cluster. To illustrate the experience sorting process, suppose that a system made up of an arbitrary number of clusters is presented with a series of apples, onions and tomatoes. All objects are different but, for example, a typical apple is generally much more different from an onion than from another apple. The input to the system is sensory input extracted from these objects.

Suppose that several clusters have been already established, and another apple is perceived. The new apple has some sensory characteristics in common with past objects, and α devices are activated in many of the clusters. In some clusters activation is limited to this α level, in others there is some β activation, but suppose that no cluster has any γ activation and therefore output. In any cluster in which there is significant β activation and no γ activation, imprinting of additional information combinations occurs at α, β, and γ levels until an output results. The combination of significant β activation and no γ activation is the higher level functional signal referred to earlier. If no cluster has significant β activation, a new cluster is imprinted to produce an output and from then on is available to respond to additional objects. The process could be initiated from no clusters at all, with clusters being added until an output was produced in response to every object. This process thus sorts experience into a set of repetition similarity conditions implemented as clusters as illustrated in figure 7. The similarity conditions are defined heuristically, and do not correspond with cognitive categories, but can be used by a competitive function with pleasure/pain feedback to generate behavior appropriate to cognitive categories.

To illustrate the competitive mechanism, suppose that the output from any cluster in the set were l recommendation to eat the perceived object, and that five clusters had been created which sorted the system perceptions of apples, onions and tomatoes. Because of the ambiguous nature of the clusters, the typical output in response to an apple might be strong from cluster one, weak from cluster two and moderate from cluster four, while the typical response to an onion might be weak from cluster one, strong from cluster two, and weak from cluster five. A very small number of trials with pleasure/pain feedback on the consequences of an action can lead to acceptance of cluster based eating recommendations in response to apples and rejection of recommendations in response to onions, even when no two apples and no two onions are identical. Electronic simulations confirm this learning effectiveness (Coward 1996).

Note that the competitive process does not change the clustering. Such a change would complicate the simple functional architecture and make it impossible to handle complex functional combinations.

The output of clusters can be clustered into more detailed clusters, and at a higher functional level hierarchies of clusters form superclusters. Superclusters generate different types of behavioral recommendations, such as aggressive, food seeking, and fearful as illustrated in figure 8. Each supercluster sorts experience into its own repetition similarity cluster hierarchy. In response to perceiving a dog, different superclusters will generate a configuration of cluster outputs which are the recommended behavior of the supercluster type towards the currently perceived dog.

Although in principle the same cluster hierarchy could be used for all types of behavior by making different use of the same outputs, there are functional advantages to the parallel hierarchies. The advantages can be understood by recalling that the sorting into repetition similarity clusters is both heuristic and ambiguous, and that a simple functional architecture needs to minimize information distribution. Consider now the cluster hierarchies generating recommended responses of food seeking and aggressive types, and in particular the clusters which generate recommended responses involving apples. Such responses might be to eat the apple or to throw the apple. Now suppose that eating behavior in response to the same set of cluster activations resulted sometimes in pleasure and sometimes pain. Such a situation might arise if two types of apple with radically different tastes differed perceptually only in skin texture, and the perception of skin texture was information not included in current cluster inputs. Detection of the condition 'contradictory results from acceptance of identical recommendations' is a recommendation to add additional input information to the cluster inputs and recluster. Acceptance of such a recommendation would be functionally valuable in the food seeking supercluster, but would simply add information distribution with no functional advantage if the apple is to be used only as a missile. Hence functional optimization is better if independent superclusters generate recommendations for different behavioral types. There is evidently a tradeoff here between functional optimization and greater use of clustering resources.

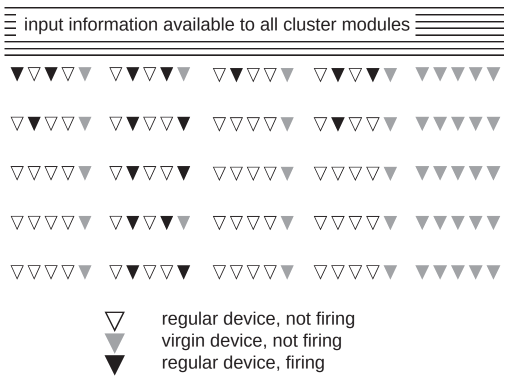

Figure 7: Information repetitions extracted from a condition or object are presented to a range of clusters, and the clusters with the strongest activation imprint additional repetitions to produce an output. Conditions are thus heuristically sorted into clusters.

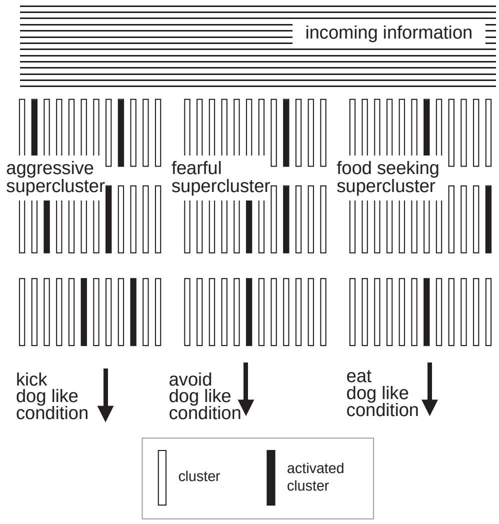

Figure 8: Parallel hierarchies of heuristically created clusters generate alternative behavioral recommendations towards the same perceived object or condition. The activation of a set of clusters corresponds with the recommendation.

'Memory' in a Recommendation Architecture

What is memory in a recommendation architecture? The first point to note is that all the information combinations active at the time an object is experienced are permanently recorded. Many of these are repetitions of combinations recorded in earlier experiences of other objects, and a small subset are combinations instantaneously imprinted at the time of the experience. Therefore a permanent trace of any past experience is available. If somehow that trace could be completely reactivated, the system state would be indistinguishable from the original experience. In practice as discussed below, only subsets of the trace can be activated independent of sensory input, but such subsets will nevertheless strongly resemble the original. The permanent trace allows the system to distinguish between objects which have been seen before and other objects. If minimal imprinting is required to generate recommendations, the object has been seen before. The phenomena which result from the permanent trace thus strongly resemble declarative memory in human brains, in which in general an object seen once will be recognized as familiar if seen again at any later time, can be bought to mind in the absence of the object only if the right combination of stimulative memories are present, but once brought to mind the mental image has some qualitative similarity to the original experience, for example in its ability to generate similar emotional states.

The second point is that because the permanent trace is the basis for generating behavioral recommendations, it is physically distributed across the superclusters discussed earlier. Local damage will typically affect only one or two superclusters, and could not remove all of the trace. Previously experienced objects will therefore still be recognized as familiar. Such local damage will reduce the ability of the affected superclusters to produce recommendations. Local damage will therefore affect behavior but not declarative memory, again a strong resemblance to biological brains (Harlow 1868, Lashley 1950).

The third point depends on the fact that repetitions are combinations of ambiguous information. Suppose the system perceives a dog and imprints to generate recommendations, but a cat and a tree are also present. Some information from cat and/or tree may be incorporated in the imprinted repetitions. If later a cat and a tree were seen at the same time, the information overlap might generate a weak dog related recommendation, which would be experienced as a weak mental image of a dog. This phenomenon is reminiscent of associative memory in biological brains, but amplification would be needed to generate a significant image. The mechanisms and value of such amplification are discussed in later sections.

The fourth point depends on the heuristic definition of the permanent memory traces. Because the process is heuristic, the combination of repetitions which will be activated in a system in response to, for example, the color red will depend on the past experience of the system. The information content of those repetitions will include information from whatever objects happened to be present at the same time in the past, subject to whether the option to include such information was provided by the information distribution management process discussed in the next section. Two systems will differ in the similarity definitions of clusters which are established and in the information combinations making up device level repetitions. The type and combination of repetitions activated in response to an experience will therefore by very individual specific.

A final point relates to the learning of skills. In a recommendation architecture such learning depends on creating the appropriate associations between clusters and behaviors through a competitive function. The relative ability of different clusters to gain control of behavior is adjusted by pleasure/pain type feedback on the competitive function. Unlike the imprinting mechanism in the clustering function, no permanent trace is created of past states of the competitive function. There is therefore no permanent record of past states of procedural memory.

In summary, there is a very strong resemblance between phenomena in a system with the recommendation architecture which heuristically defines its own functionality and the phenomena of memory in biological brains. For a more extensive discussion see Coward (1990).

Dream Sleep: The Management of Information Distribution

As discussed earlier, a simple functional architecture requires minimized distribution of information. In a system which heuristically defines its own functional components, the distribution of information between such components must also be heuristically minimized.

Consider two aspects of this problem. The first is that at the device level, the inputs most likely to result in functionally useful repetitions are inputs similar to those forming functionally useful repetitions in other devices at the same functional level. Provisional inputs to devices can be assigned randomly, but a statistical bias on inputs which have frequently participated in firing devices at the same level in the past would increase the probability of useful combinations and reduce information distribution. At a higher level, the output of a cluster may be information valuable in achieving a functionally valuable similarity condition in another cluster. However, connecting every cluster to every other cluster would massively complicate the functional architecture. The alternative is to assume that simultaneous activation is a probable indicator of functional value, and assign provisional inputs to complex clusters from simpler clusters which have frequently been active at the same time in the past.

The requirement is therefore to achieve provisional connectivity between two functional components whenever there is frequent correlated activation is the past. Coward (1990) has argued that providing an environment in which frequent past correlation can be determined is a primary function of dream sleep, achieved by performing a fast rerun of an averaged past, with a bias towards the most recent past. Work by Skaggs and McNaughton (1996) provides evidence in favor of this proposal.

The general absence of memory from dream sleep is as expected for a process which is configuring the resources which will be required for memory in a subsequent wake period. Electronic simulations have confirmed that a process of this type substantially improves the effectiveness of learning (Coward 1996).

Role of Sensory Independent Mental Images

How can the weak associative activations described in an earlier section be amplified into a full mental image, and what would be the functional value of such amplifications? First, how could they be amplified? Consider the operation of signals between two systems with the recommendation architecture. A system sees a lion. A set of repetitions are activated by the sensory input which generate recommendations to run from the lion, and also shout "lion". The hearing of the shout "lion" by a second system generates in that system a subset of the same repetitions which would be activated if that second system had actually seen the lion. These repetitions generate the same recommendations as if a lion had been seen. However, because the first system could hear its own shout, a feedback route has been established. Such a feedback route could amplify the weak associative activations if those activations corresponded with recommendations to speak the signal for the associatively activated object.

What would be the functional value of such feedback driven activations? Consider now how a system with the recommendation architecture would make tools. The problem to be solved is how to activate tool making recommendations within the system. The simplest mechanism is to use an existing tool as a model. Repetitions extracted from seeing a finished tool and a piece of rock combine to generate carving recommendations. However, if the finished tool is broken or lost a model is not available. If current inputs to the system from the environment were similar to conditions under which a tool was used in the past, the associative overlap process could lead to the word "tool" being spoken. The word is heard by the speaker and activates a large set of the type of repetitions which would be generated by perceiving a tool. These repetitions can then generate tool making recommendations in the absence of a model.

If physical routes internal to the system develop to carry the feedback independent of externally spoken signals, then the range of mental images which can be activated is not limited by the current signal vocabulary. A mental image of different parts of the tool can be activated even if they have no name, making much more detailed control of carving possible.

Such feedback routes activate the simple repetitions which have frequently been active when the currently active complex repetitions have been active in the past. An additional functional advantage can develop once this capability is in place. Suppose such a system perceives a dog, and generates a range of recommendations: to pat the dog, kick the dog etc. If a recommendation to kick the dog can activate the type of repetitions which were frequently active in the past when the system has kicked objects, these repetitions would be added to current sensory repetitions and expand the range of alternatives, for example to include avoiding the dog. In other words, it becomes possible to perform a much more extensive search of individual specific memory to generate behavioral recommendations.

Such a search takes time, and delays response. It is therefore only valuable for generating behavior under extremely complex conditions in which a delay can be tolerated. Such a situation exists for complex social interactions between systems. A system with this search capability would experience a constant succession of mental images independent of sensory input.

Nature of Representation in a Recommendation Architecture

The response to an external condition in a system with the recommendation architecture is the activation of a large set of device level information combinations. In a system with the ability to heuristically define its own functionality, most of these combinations are repetitions of combinations recorded in earlier experiences and a small subset are combinations recorded during the experience of the condition and which will be available as possible repetitions in future experiences. Combinations include information from both the external condition and the system itself, in other words $W = f(R, O)$.

In what sense can such activations be regarded as representations? Peschl and Riegler (1997) identify three points on which the traditional concept of representation can be seriously questioned: linguistic transparency; referential representation; and embodiment, construction and dynamics of knowledge. In the recommendation architecture, groups of repetitions (i.e. clusters) generate outputs. The ambiguity of information within the clustering function means that a wide range of different combinations of cluster activations may act through a competitive function to generate the behavior of speaking the same category naming word. So activations in the recommendation architecture support the suggestion of Peschl and Riegler that "… the processes responsible for generating … linguistic categories [do not] … have to be based on these categories".

The function of an activation is to generate behavioral alternatives, not to map the environment, and the activation in response to similar external conditions will vary considerably depending on the needs of the organism (e.g. hunger, sense of threat, sense of weakness etc.). There are therefore "… [no] neurons (or groups of neurons) whose activations correlate with external events in a stable and referential manner".

A system with the recommendation architecture "actively extracts and constructs those environmental regularities which are relevant to its particular survival". There is a sense in which no condition in the external environment ever exactly repeats a past condition. An organism must extract repetition in order to guide behavior, and in this sense patterns and categories are artifacts of a mental architecture dependent on repetition (Coward 1990).

Conclusions

Biological brains are strongly constrained by selection pressures to adopt simple functional architectures. The recommendation architecture is the only qualitatively different alternative to the instruction architecture ubiquitous in commercial electronic systems. The functionality in a system with the recommendation architecture strongly resembles psychological experiences such as learning, object recognition, associative memory, dream sleep without recall, constant sensory independent sequences of mental images, and individual differences between the experience of the same conditions.

In a recommendation architecture these phenomena can be described in both psychological and device terms, and causal connections at the psychological level are reflected in causal connections at the device level. It is argued, following Coward 1990, that biological brains, and in particular human brains, have the recommendation functional architecture. As a result, complete descriptions of human psychology in equivalent neurophysiological terms are possible.

References

Bardeen, J., Cooper, L.N. & Schrieffer, J.R. (1957) *Physical Review* 108: 1175.

Carpenter, G.A. & Grossberg, S. (1988) The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network, *IEEE Computer* 3: 77–88.

Cauller, L. (1995) Layer I of primary sensory neocortex: Where top-down converges with bottom-up. *Behavioral Brain Research* 71: 163–170.

Coward, L. A. (1990) *Pattern Thinking,* New York: Praeger.

Coward, L. A. (1996) Understanding of Consciousness through Application of Techniques for Design of Extremely Complex Electronic Systems. Presented at *Towards a Science of Consciousness*, Tucson, Arizona.

Coward, L. A. (1997) The Pattern Extraction Hierarchy Architecture: a Connectionist Alternative to the von Neumann Architecture. In: Mira., J., Morenzo-Diaz, R. & Cabestanz, J. (eds.) *Biological and Artificial Computation: from Neuroscience to Technology.* Berlin: Springer, pp. 634–43.

Coward, L. A. (1998) A functional architecture approach to neural systems, to be published.

Harlow, T. M. (1868) Recovery from passage of an iron bar through the head. *New England Medical. Society* 2: 327–46.

Kohonen, T. (1990) The Self-Organizing Map. *Proceedings of IEEE* 78, pp. 1464–80.

Lashley, K. S. (1950) In Search of the Engram, *Symposia of the Society for Experimental Biology* 4: 454–82.

Levine, J. (1983) Materialism and Qualia: The explanatory gap. *Pacific Philosophical Quarterly* 64: 354–361.

Peschl, M. & Riegler, A. (1999) Does Representation Need Reality? Rethinking Epistemological Issues in the Light of Recent Developments and Concepts in Cognitive Science. *This volume.*

Skaggs, W. E. & McNaughton, B. L. (1996) Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science* 271: 1870–1873.