Coward, L.A. (1999). A physiologically based approach to consciousness, *New Ideas in Psychology*, 17(3), 271-290

A Physiologically Based Approach to Consciousness

L. Andrew Coward
email: landrewc@aol.com

A Physiologically Based Approach to Consciousness

L. Andrew Coward
System Architecture Consultant
P.O. Box 470, Elmendorf
Texas 79112

Telephone: 972-684-7493
email: landrewc@aol.com

Abstract

The nature of a scientific theory of consciousness is defined by comparison with scientific theories in the physical sciences. The differences between physical, algorithmic and functional complexity are highlighted, and the architecture of a functionally complex electronic system created to relate system operations to device operations is compared with a scientific theory. It is argued that there are two qualitatively different types of functional architecture, and that electronic systems have the instruction architecture based on exchange of unambiguous information between functional components, and biological brains have been constrained by natural selection pressures into the recommendation architecture based on exchange of ambiguous information. The mechanisms by which a recommendation architecture could heuristically define its own functionality are described, and compared with memory in biological brains. Dream sleep is interpreted as the mechanism for minimizing information exchange between functional components in a heuristically defined functional system. The functional role of consciousness of self is discussed, and the route by which the experience of that function described at the psychological level can be related to physiology through a functional architecture is outlined.

Keywords: consciousness; cognitive architecture; memory; dream sleep; physiology; complex systems

The Nature of Scientific Theories

A scientific theory in the physical sciences establishes a description made up of consistent causal relationships between well-defined entities which map into a body of experience. Deeper scientific theories establish descriptions on multiple levels of detail in which entities and causal relationships on one level can be mapped exactly into entities and causal relationships on other levels. For example, phenomenological chemistry establishes entities such as acids, alkalis, salts, and so on, and consistent causal relationships such as acid plus alkali generates salt plus water. At a deeper level, it is possible to create descriptions of these entities in terms of atoms, molecules, and ions in which causal relationships between these detailed concepts exist whenever causal relationships exist between the corresponding chemical entities. At a yet deeper level, equally consistent causal descriptions can be created in terms of quantum mechanics.

An analogous theory of consciousness must establish a set of entities X, Y, Z, and so forth at the psychological level with consistent causal relationships such as the presence of X and Y results in Z. The theory must also establish a corresponding set of entities x, y, z, and so forth at the physiological level such that if X plus Y causes Z at the psychological level of description, then x plus y causes z at the physiological level of description. Descriptions at high level contain much less information than descriptions at a detailed level. Therefore although two different psychological states X and Y must always correspond with different physiological states x and y, several physiological states x1, x2, and x3 could correspond with psychological states that are indistinguishably different at the psychological level. For example, although the same physiological state cannot correspond with both the perception of red and of green, multiple physiological states may all correspond with perceptions of the color red that are indistinguishable at the psychological level.

Given the wide difference in information content between psychological and physiological descriptions, it may be necessary to establish one or more intermediate levels of description to bridge the gap, in a manner analogous with the atomic level descriptions that bridge the gap between quantum mechanics and the phenomenology of chemistry as discussed earlier. Pribram (this issue) points out that a wide range of different phenomena are all labeled "consciousness", and argues that our everyday conscious experience results from a delay between an incoming pattern and the generation of an outgoing pattern that regulates behavior, and that this consciousness of our own experience can be related to the dynamics of the web of dendritic connections. He further argues that Gabor functions are a good way of describing events at the dendritic level in a way that can relate those events to this phenomenon of consciousness. Pribram thus argues that Gabor functions provide the intermediate descriptive level between descriptions in terms of dendritic physiology and descriptions in terms of the psychological phenomena of consciousness.

Establishing consistent sets of entities and causal relationships on many levels is a challenging undertaking. However, the problem bears a striking resemblance to the much less well known problem of establishing a consistent set of functional descriptions on many levels required for the design of functionally complex electronic systems, and it will be argued in this paper that biological brains are functionally complex systems that have been constrained by natural pressures to adopt analogous hierarchies of functional descriptions, and that these description hierarchies are the basis for a scientific theory of consciousness.

The Definition of System Complexity

The term "complex system" is used to label a number of radically different types of system. These different types include physical, algorithmic and functional complexity. In a physically complex system the interactions between components are similar for all components, but because of the very large number of components the evolution of the system state from a starting point is very complex. Regularities may be observable in that evolution, but in general small differences in the starting state may result in very large differences in later states. Chaotic systems in general are of this type, the classic example being the atmospheric weather system (see for example Lorenz 1996), in which although the interactions between gas molecules are known and similar for all pairs of molecules, the evolving state of the atmosphere is very complex. Almost the opposite extreme is a functionally complex system. Such a system moves from a starting state through a series of later states, but although such systems contain very large numbers of components the rules determining the behavior of a component vary widely depending on the component. Thus in electronic systems there are "an exceedingly large number of possible design elements" (Perry & Wolf 1992) and in biological brains every neuron may have a different combination of inputs and respond to different sets of conditions. The later states of a functionally complex system deriving from two slightly different starting points are frequently constrained by these component rules to be similar, or in other words the system response to similar objects or conditions is similar.

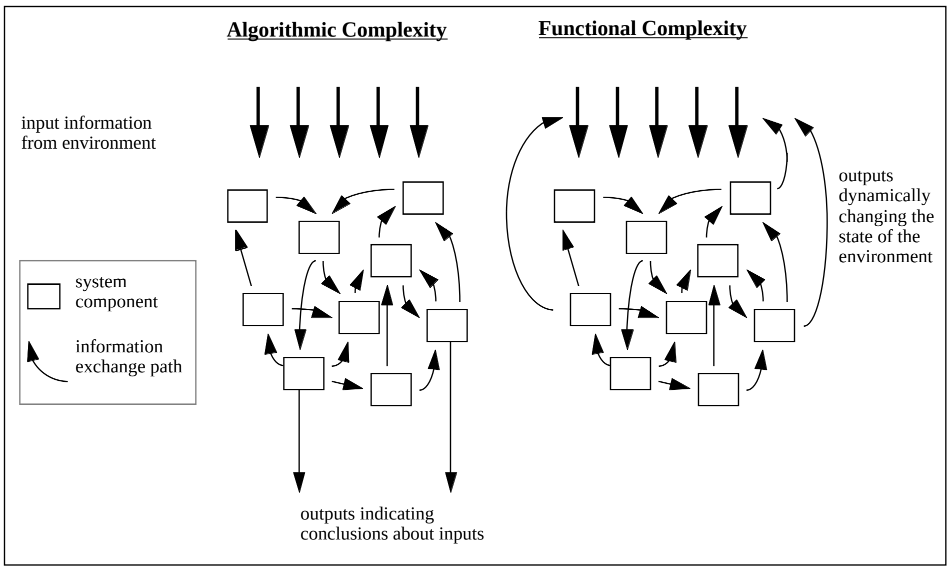

Figure 1. Comparison between algorithmic complexity and functional complexity. An algorithmically complex system is made up of many modules which accept a complex information input and process it with significant exchange of information to reach conclusions about the input. A functionally complex system also accepts a complex information input, but its outputs dynamically change the inputs. Functional complexity can be approximated by a sequence of algorithmically complex problems, but this approximation is inadequate for a real time system unless the processing power available is unlimited.

The important difference between algorithmic and functional complexity is illustrated in figure 1. In an algorithmically complex system, conclusions about a large body of input information are determined by algorithmically complex information processing that may involve information exchange between many modules. An example might be using the output of a retina to identify a face. In a functionally complex system a large body of input information is processed, generally but not necessarily in an algorithmically complex fashion, to generate system actions that can dynamically change some of the input information. An example might be using the output of a retina to generate actions that dynamically change both the orientation of the retina and the environment being perceived. The time sequence in which different system components perform their functions and the timing with which an output is generated both affect the correctness of the output. In computer science this is what is known as a real time system (see for example Avrunin, Corbett & Dillon 1998). One approximation to a functionally complex system is to treat it as a sequence of algorithmically complex problems, with the sequence of conclusions being the system actions, but in the absence of either unlimited time or unlimited processing capacity this approximation is impractical for systems performing complex real time functionality.

The Architectures of Complex Electronic Systems

Over the last 30 years the complexity of commercial electronic systems has increased to the point at which some individual systems have billions of hardware components and tens of millions of lines of software code (Coward 1990; Soni, Nord & Hofmeister 1995). Many of the most complex systems are real time systems. In the late 1960s it first became apparent that the complex functionality represented by software must be patterned and structured in a disciplined fashion at a higher level than the software code itself (Dijkstra 1968). In the 1970s Parnas and his collaborators in a number of papers (see Parnas, Clements & Weiss 1985 for an overview) developed an understanding of how complex functionality could be partitioned into independent modules. As the discipline of software architecture has developed in the 1990s it has become clear that a multilevel descriptive hierarchy is required (Soni, Nord & Hofmeister 1995) and that in the absence of a common, well-defined architecture it is extremely difficult to achieve integration of different functional modules (Garlan, Allen & Ockerbloom 1995).

The architecture of a functionally complex system partitions the operations of the system into a set of modules, each of which performs a roughly equal proportion of the operations of the system. Each module is itself partitioned in a similar manner into subcomponents, and the partitioning process continues until the elements of functionality can be understood in terms of the operation of hardware devices or small groups of devices. This partitioning is illustrated in figure 2.

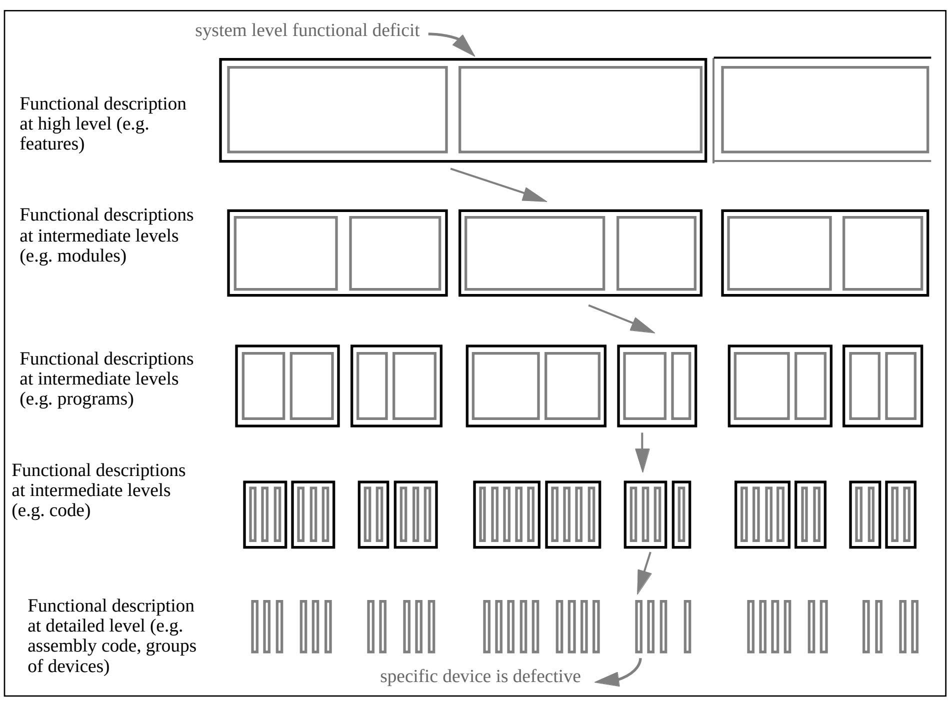

Figure 2. Schematic view of a section of a functional architecture. System functionality is described on a number of levels of detail. Functionality is partitioned into roughly equal components on each level, and the information exchange between components minimized to maintain functional separation. Such an architecture makes it possible, for example, to trace a system deficit from the system level where it is experienced to the detailed level where it can be fixed. The examples are for an instruction architecture.

There are a number of reasons why an architecture of this type is required. One is illustrated in figure 2. If there is a device failure in a functionally complex system, the failure will manifest itself as a deficit in system functionality. To repair the failure it must be possible to follow a logical path that relates the system deficit to an individual software element, individual device or small group of devices. A second reason is that if the functionality of a system must be modified, it must be possible to relate the desired change specified at a system level to a limited set of changes at a detailed level that will not affect other functionality. Other reasons include the needs to build multiple copies of the system and to make multiple use of certain modules for efficiency reasons (Coward 1999). These needs place a set of requirements on the architecture. As discussed earlier, the architecture must partition functionality into roughly equal components on many levels of detail. Although exactly the same functionality is described on every level, there must be a compression of the information content of the description between the detailed and the higher levels. Finally, the partitioning into modules must be such that the information that needs to be exchanged between modules as they perform their separate functionality must be minimized, otherwise there is no real separation between the modules. Such a functional hierarchy strongly resembles the descriptive hierarchy of a full scientific theory.

Although biological brains are not designed by an intellect driven process, they experience many of the same pressures in favor of simple functional architectures. They are constructed from DNA "blueprints", they must have some ability to recover from construction errors and damage, there are competitive advantages in multiple use of functional components, and in the process of evolution limited random changes at a detailed level must sometimes result in useful functional changes at a system level. For these reasons Coward (1990) has argued that biological brains have been constrained by natural pressures into simple functional architectures.

Alternative Types of Functional Architecture

Biological brains have very little functional resemblance to current commercial electronic systems, other than extreme functional complexity. For example, in electronic systems all information is maintained with high integrity, because errors in single elements of information can have large functional consequences. Functionality in electronic systems is defined in advance by design, and the ability of electronic systems with complex functionality to modify their functionality independently based on experience is minimal.

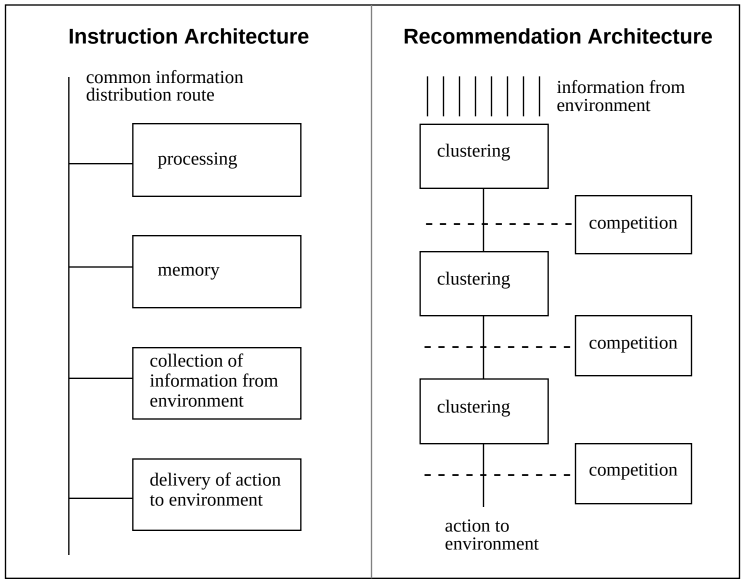

Figure 3. The major functional separations in the two types of functional architecture.

Coward (1990; 1999) has argued that there are two qualitatively different types of functional architecture. The difference centers on the type of information that can be exchanged between functional components. In commercial electronic systems a component receiving information from another component has a completely unambiguous context for that information. As a result, the outputs generated by functional components can be interpreted as instructions, or commands for the system to perform actions. When one component is using an element of information, that information element is potentially ambiguous to any other component, and computation is therefore forced to be sequential. Parallel processing can only occur if system information can be partitioned in such a way that different components act upon different, orthogonal segments. In a complex system this partitioning must be dynamic and is difficult to achieve. Because many components interact with the same information at different times a reference copy of system information must be maintained. This copy is known as memory. The need for efficiency of sequential execution results in a common processing function. Commercial systems therefore ubiquitously exhibit a memory, processing functional separation as shown in figure 3. In such systems it is extremely difficult for functional components to heuristically change their functionality, because of the difficulty in maintaining the unambiguous context for all information exchange.

If the information exchanged between functional components is ambiguous, then information received by one component may be affected by the unknown internal states of other components. Such information can be meaningful but ambiguous in the sense that it only correlates partially or probabilistically with actual external environmental or system conditions. As a result, the outputs of components can only be interpreted as system action recommendations, and a separate process outside the functional hierarchy is required to integrate the recommendations into an action. This separate process makes use of information on the consequences of actions to heuristically define the association between the outputs of the functional hierarchy and actual behavior. The result is a separation between the functional hierarchy and the competitive selection of action as illustrated in figure 3. The functional hierarchy is labeled clustering for reasons discussed below.

If components receive ambiguous information, that information must still be adequate to allow the component to discharge its function. The information content of component outputs must be sufficient for other components to use, and components must have a context within which they can use input information. Heuristic definition of functionality is possible if heuristically defined components have an information context within which they can use a mixture of indications of external environment and internal component conditions, and generate outputs within that context. A central advantage of the imprinting algorithms proposed by Coward (1990) is that it provides a context for information exchange between heuristically defined functional components. The use of ambiguous information also relaxes the requirement for sequential execution. If enough information context and content are available then a useful system action can be generated on the basis of the outputs which components are producing at the moment action becomes essential.

In a functional architecture, the description of functionality on any one level has several aspects. One aspect is the information that is provided to each module on the level and the sources of that information, which may be other modules on the same level or inputs from outside the functional hierarchy. A second aspect is the description of the operations performed by each module on its inputs in order to generate its outputs. A third aspect is the destination of the outputs, which may be to another module on the same level or to outside the functional hierarchy. The compression of the information content of the functional descriptions from one level to the next higher level must be adequate for the higher level but reduce the detail of the descriptions of the information input and output, the detail of the operations on the information, and the detail of the functional result of the component operation. In an instruction architecture, because the information is unambiguous, separate elements of detailed information on one level can be combined into a single less detailed data object on a higher level. The descriptions of operations are limited to the operations on the higher level data objects, and the functional implication is a higher level system action. To give a very simple example, at high level a module could be described as recording the results of a set of calculations. At an intermediate level the function could be described as receiving a file from another module and writing that file on the hard disk of a computer. More detailed descriptions would describe the collecting of all the information elements of the file from different sources, selecting the physical locations on the hard disk at which different information elements would be recorded, and performing the recording process.

A similar process for compression of description information must exist when the information is ambiguous. In the case of the description of functionality, the compression can be similar to the case for an instruction architecture, except that the outputs are recommendations rather than commands. Compression of the description of data objects is more complex because of the ambiguity of the information at every level. An information object at high level must summarize the content of a larger number of information objects at a more detailed level in a manner in which the content of the summary contains enough information to provide all that is required at the higher level.

The information compression mechanism described in detail in Coward 1990 was that the functional components at every level are sets of information combinations, and the output of a component indicates the presence of a large subset of the information combinations in the set. The functional hierarchy then becomes a hierarchy of repetition similarity, with the most detailed functions being the detection of repetitions programmed into individual devices, the next level being the detection of large subsets of clusters of device level combinations and so on. This hierarchy is analogous with a pattern, category, supercategory hierarchy, with the important differee not meaningful concepts within the functional hierarchy, they are only defined after the outputs of the functional hierarchy have been processed through a competitive function.

A critical issue is how the output information from a functional component at one level can adequately represent on its level the complete information at the more detailed level. A mechanism by which this could be achieved, which was proposed in Coward 1990 for a system that heuristically defined its own functionality, is described in the next section. Discussion of the scaling of neural network functionality rarely considers this issue explicitly, but unless addressed in detail it will not be possible to scale to complex functionality. For example, Anderson & Sutton (1997) argue that chaotic attractors can be higher level abstractions in networks of networks, but they do not propose any mechanism for functionally usable compression of description information using chaotic attractor concepts.

The Heuristic Definition of Functionality: Learning from Experience.

The processes for using information imprinting algorithms to heuristically create a functional hierarchy and associate the outputs of that hierarchy with appropriate behaviors in a recommendation architecture was originally described in Coward 1990. In this process a system organizes its experiences, viewed as a sequence of information inputs, into a hierarchy of repetition similarity. The system then associates different combinations of repetitions with different behaviors.

At a detailed functional level, combinations of information are imprinted on devices. Suppose that 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 it will produce the 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 on the device that 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 an extensive feedback connectivity generate the same functionality as the simple algorithm (Coward 1999). Feedback connectivity of the appropriate type is observed in biological brains (Cauller 1995).

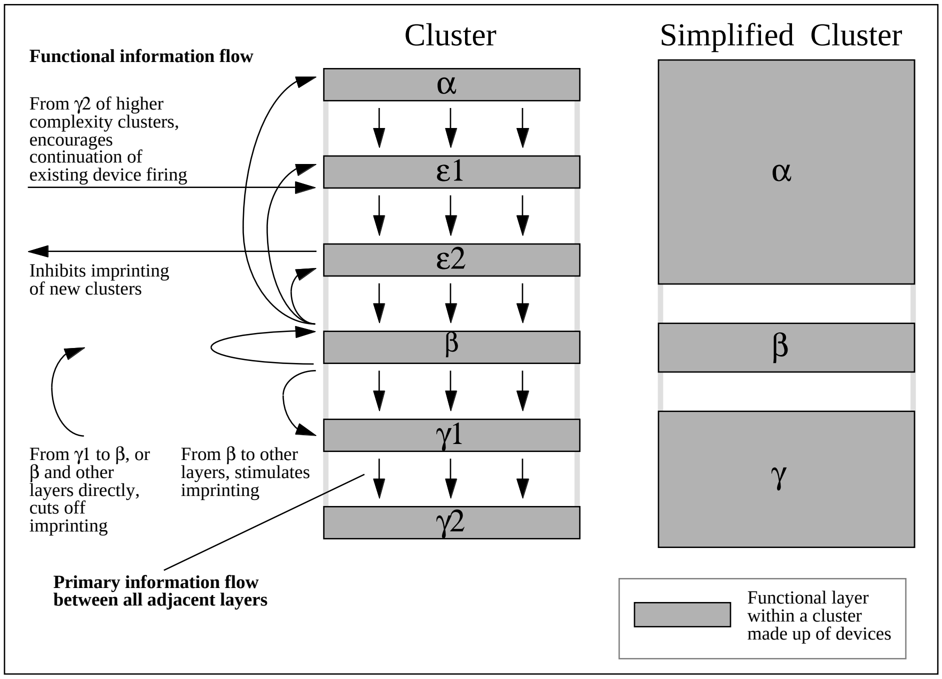

Figure 4. Connectivity of a similarity cluster module made up of layers of devices which can imprint combinations of primary input information. The layers perform the similarity subfunctions described in the text. For a system with moderate functional complexity only three layers could be required. Primary information inputs to devices in a layer come mainly from devices in the preceding layer. Functional inputs to devices come from an average cross section of devices in the appropriate functional layer.

All system input experiences are sorted into combinations of repetition similarities on various levels of complexity by the clustering functional hierarchy. Groups of devices form layers, and sequences of layers form clusters as illustrated in figure 4. A device has information inputs from devices in preceding layers, mainly the immediately preceding layer. A device also has functional inputs from large sets of devices in specific layers. These specific layers may be located in the same cluster as the device or in different clusters. The source of the input is a critical part of the context for the information received via the input. Inputs from the same physical source have a similar (although ambiguous) context.

Clusters are arranged in levels as illustrated in figure 5, and a series of levels makes up superclusters as illustrated in figure 6. The outputs of superclusters are directed to competitive functions. Every experience is a system input information combination, inputs are derived from both the external environment and the physical system being controlled. The cluster set within a supercluster is managed towards the production of outputs from one or more output level clusters in response to every experience. In other words, every experience tends to produce a range of action recommendations.

A simplified version of a cluster could have just three levels as also illustrated in figure 4. In the simple cluster the functionality of levels α and ε2 is combined in α, ε1 is not required because it manages the presence of multiple repetitions per device, and the functionality of levels γ1 and γ2 are combined in γ. Devices in layer α can be imprinted with combinations of input information, devices in layer β can be imprinted with combinations of outputs from α, and devices in layer γ can be imprinted with combinations of outputs from β and generate the outputs from the cluster. Such clusters would be appropriate for systems of moderate functional complexity. To illustrate the experience sorting process, suppose that a system made up of an arbitrary number of clusters in one level as illustrated in figure 5 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.

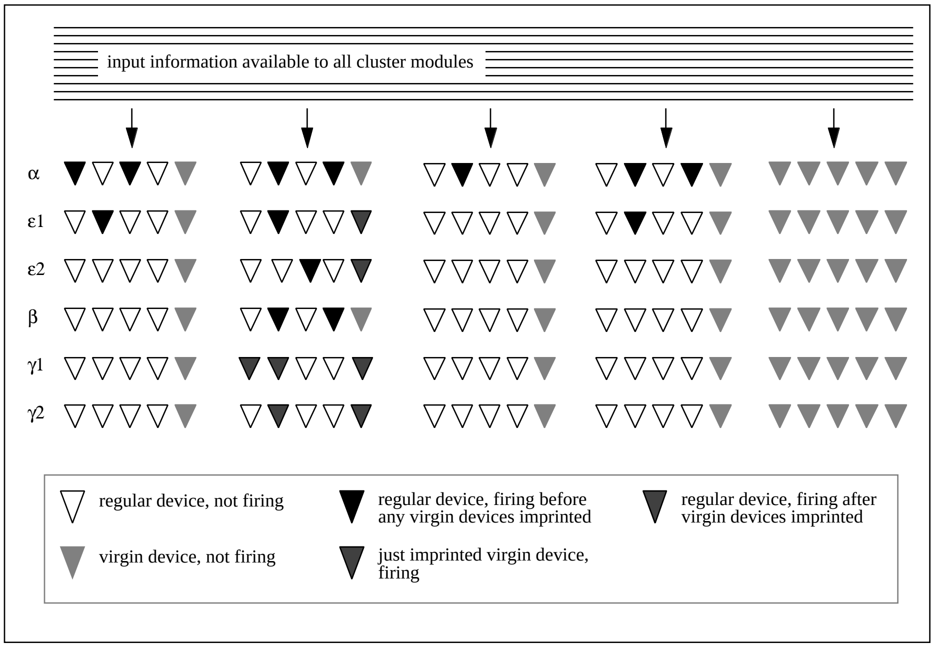

Figure 5. The sorting of information representing a sequence of experiences into repetition similarity clusters. Information repetitions present during the experience of a condition or object are available to a range of clusters, and the clusters with large subsets of these repetitions already imprinted are activated to imprint additional repetitions until an output is produced from one or more clusters. Conditions are thus heuristically sorted into clusters.

For the discussion, focus attention in figure 5 on just the α, β and γ2 layers. Suppose that several clusters have been 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 γ2 activation and therefore no output. In any cluster in which there is significant β activation and no γ2 activation, imprinting of additional information combinations occurs at any level until an output results. The combination of significant β activation and no γ2 activation is the higher level functional signal in figure 4. 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. Such new clusters begin with large but random connectivity from one layer to devices in the next layer. Cluster imprinting is inhibited by ε2 output from a peer cluster as illustrated in figure 4 in order to limit unnecessary proliferation of clusters. The process could be initiated with only unimprinted clusters, which are imprinted until an output was produced in response to every object (Coward 1999). This process sorts experience into a set of repetition similarity conditions implemented as clusters as illustrated in figure 5. The cluster 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.

The ambiguous component outputs generated using the imprinting algorithm are adequate for communication between heuristically defined components for several reasons. The first reason is that outputs can communicate both detection of an information condition meeting an internally defined similarity criterion and variations within the criterion. Secondly, a hierarchy of functional signals based on similarity at different levels is supported. Thirdly, the algorithms allow modulation of the similarity clustering based on its discrimination effectiveness for functional purposes without using direct feedback of consequence information. Such direct use would change the context of information received from changed modules by modules not affected by the change.

As a result the functionality of the system can be dynamically defined. An experimental but plausible response can be generated immediately in response to a novel experience provided that there are some elements of similarity to past experiences.

To illustrate the competitive mechanism, suppose that the output from any cluster in the set were a 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 1999).

Note that the competitive process does not change the clustering. Such a change would destroy the intercomponent information context and make it impossible to handle complex functional combinations.

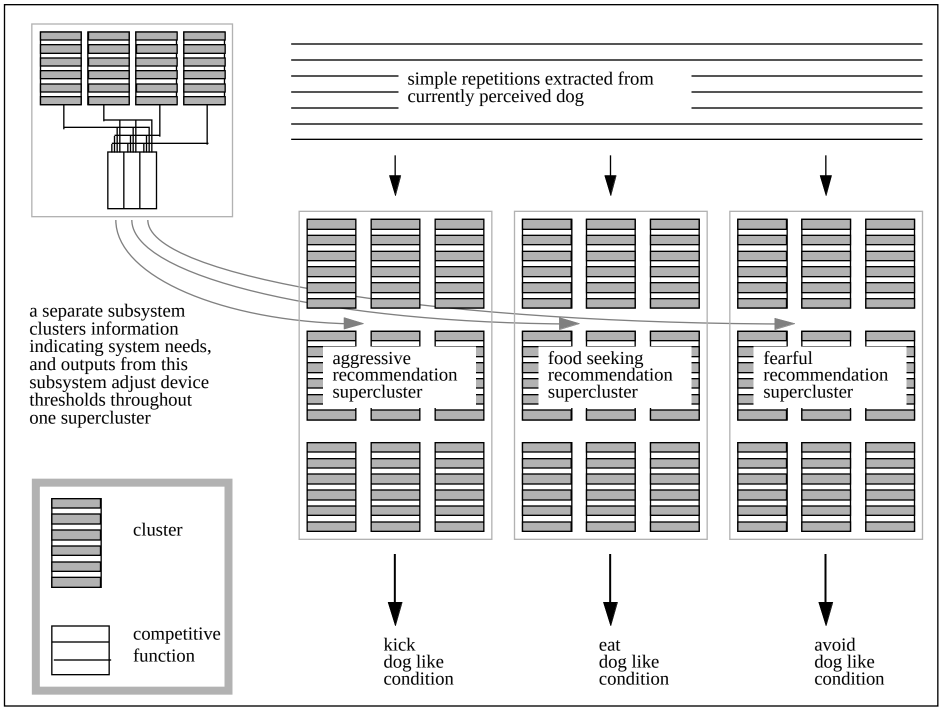

Figure 6. Schematic view of a set of superclusters generating alternative behavioral recommendations. Within one supercluster the outputs of clusters on one level are sorted into clusters on another level and so on, creating a hierarchy which generates behavioral recommendations of the supercluster type towards the currently perceived object or condition. The supercluster output resulting from the activation of a set of clusters within the supercluster corresponds with the recommendation. A separate clustering/competition subsystem can modulate the relative probability of different types of recommendation.

The outputs of a cluster can be regarded as information inputs to a set of later clusters, which will create similarity subconditions of the condition defined by the main cluster. Such subclusters could use the outputs of more than one source cluster. At a higher functional level such hierarchies of clusters form superclusters which generate different types of behavioral recommendations, such as aggressive, food seeking, and fearful as illustrated in figure 6. A supercluster sorts experience into its own repetition similarity cluster hierarchy. In response to perceiving a dog, for example, different superclusters will generate a configuration of cluster outputs which are the recommended behavior of the supercluster type towards the currently perceived dog.

In principle one cluster hierarchy could be used for all types of behavior by making different use of the same outputs. The functional advantages of parallel hierarchies for different superclusters 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 the clusters which generate recommended food seeking and aggressive type responses involving apples. Such responses might be to eat the apple or to throw the apple. Suppose that 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. In this situation eating behavior in response to the same set of cluster activations would result sometimes in pleasure and sometimes pain. If detection of the condition 'contradictory results from acceptance of identical recommendations' were a recommendation to add additional input information to the cluster inputs and generate additional cluster outputs in the presence of these inputs, then acceptance of such a recommendation would be functionally valuable in the food seeking supercluster, but would not occur if the apple is to be used aggressively. Note that the addition of outputs does not change existing outputs, which could be to clusters not concerned with apple related recommendations. The information context for these clusters is therefore unaffected. Hence functional optimization is improved if separate superclusters generate recommendations for different behavioral types.

The descriptive hierarchy at a detailed level is made up of devices, at the next level layers (α, β, γ, and so on), at the next level clusters, at the next level superclusters and so on. Between each level there is the necessary compression of description, but in such a way that the compressed information on one level is adequate to represent on its level the full information at more detailed levels. For example, the outputs from the $\gamma2$ layer of a cluster represent the functionality of all the devices in the cluster to other clusters, and those outputs are adequate to represent that functionality. The $\gamma2$ layer outputs from clusters in the output layer of the superclusters contain enough information to represent the operations of the supercluster and are interpreted by a competitive function into behaviors.

Layers are functional components to which a functionality can be assigned. The α layer manages the selection of input information to the cluster, the β layer detects enough repetition similarity to add a condition to the cluster, the $\gamma1$ layer detects enough output to inhibit further imprinting and the $\gamma2$ layer produces outputs sufficiently diverse to make the information usable to other modules. As discussed in detail in Coward 1999, another necessary function is detection of enough similarity to inhibit creation of new clusters which is performed by layer $\varepsilon2$. The function performed by level $\varepsilon1$ is the management of the orthogonality of multiple repetitions programmed on the same device. A sufficiently complex system would thus require six independently optimized layers for effective system operation as illustrated in figure 4.

In a general sense the separation between clustering and competition is analogous with the distinction made between unsupervised and supervised learning in neural networks (see for example Card, Rosendahl, McNeill & McLeod 1998), and the mechanism of sorting experience into similarity clusters has some resemblance to unsupervised learning algorithms such as adaptive resonance (Carpenter & Grossberg 1988) or Kohonen nets (Kohonen 1990). There are, however, some critical differences. In unsupervised learning there is no discussion of how to establish a multilevel functional hierarchy with compression of description between levels and minimized information exchange between modules, and there is no discussion of how to maintain an adequate context for ambiguous information exchange, if addressed at all the information exchange between modules above the device level is typically unambiguous. Such algorithms therefore cannot be scaled to complex functionality. In addition, the imprinting algorithms exhibit phenomena with a strong similarity to memory in biological systems as discussed below.

Memory like phenomena

In a recommendation architecture using the imprinting algorithms any information combinations that are 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. An experience similar to a past experience will require little imprinting to generate output recommendations. The degree of imprinting therefore allows the system to distinguish between objects that have been perceived before and unfamiliar objects. Reminiscing about an object experienced in the past requires reactivation of the combination of repetitions generated by the object without the presence of the object. If that combination were completely reactivated, the system state would be indistinguishable from the original experience. Even an activated subset of the combination will resemble the original activation. However, the activation of such subsets is much more difficult in the absence of the appropriate sensory input. The phenomena which result from the imprinting algorithms thus resemble declarative memory in human brains, in which an object seen once will generally be recognized as familiar if seen again at any later time, can be recalled only if the right combination of stimulative memories are present, but once recalled the mental image has some qualitative similarity to the original experience.

Because the permanently recorded combination of repetitions is created in the course of generating alternative behavioral recommendations, it is physically distributed across the different superclusters. Local damage will affect only one or two superclusters, and previously experienced objects will therefore still be recognized as familiar by the undamaged superclusters. 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 resemblance to biological brains (Harlow 1868, Lashley 1950).

The imprinting process establishes the connectivity that can support associative activation of reminiscent images because repetitions are combinations of ambiguous information. Suppose a system perceives a dog and imprints to generate recommendations at a time when 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 weak dog related recommendations, which would be experienced as a weak mental image of a dog. This phenomenon resembles 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.

Because repetitions are created heuristically, the combination of repetitions that 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 often include information from objects that happened to be present at the same time in the past as experiences of the color red. Two systems with a different sequence of experiences will differ in the information combinations making up device level repetitions, and in the cluster and higher level combinations. The type and combination of repetitions activated in response to an experience will therefore be specific to the individual system.

The learning of a skill depends on creating appropriate associations between clusters and behaviors through a competitive function, and no permanent record is created of past states of the competitive function. There is therefore no permanent record of past states of skill learning. This inability to access past states is characteristic of procedural memory in biological brains.

The strong similarities between the phenomenology of a system with the recommendation architecture in which functionality has been heuristically defined and the phenomenology of memory in mammal brains tends to confirm the hypothesis that such brains have been constrained into the recommendation architecture.

The Management of Information Distribution

In a system that heuristically defines its own functional components, the distribution of information between such components must also be heuristically minimized. At the device level, minimized information distribution implies that device inputs should as far as possible be functionally relevant. Imprinting selects the currently active subset of a randomly assigned set of inputs. If the random assignment were biased in favor of inputs that have frequently participated in firing devices at the same level in the past the functionally irrelevant distribution of information would be reduced.

At the cluster level, the outputs of a particular cluster could be information valuable in achieving a functionally useful similarity condition in another cluster. However, connecting every cluster to every other cluster involves excessive information distribution. One alternative is to assume that simultaneous activation is a probable indicator of functional value, and assign provisional inputs to complex clusters from simpler clusters that have frequently been active at the same time as the complex cluster in the past.

The requirement is therefore to achieve provisional connectivity between two functional components whenever there is frequent correlated activation in the past. Coward (1990) 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 & McNaughton (1996) provides evidence in favor of this proposal.

The observation that memory of dreams is generally not present is consistent with a process that is configuring the resources 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 1999).

Functional Role of Consciousness of Self

As discussed by Pribram (this issue), consciousness can be defined in many different ways, but Pribram focuses on the everyday experience of consciousness. It is important for any theory to define more precisely both the phenomenology and the postulated functional role of this everyday experience. Many authors back to William James have identified the experience of constant successions of sensory independent mental images and the ability to imagine both self and a wide range of imaginary conditions as important elements in the phenomenology of everyday consciousness. The ability to remember both real and imaginary experiences is fundamental to this phenomenology.

A functional role for this phenomenology was proposed by Jaynes (1976). His proposal was that the ability to imagine both self and a wide range of imaginary conditions is a mechanism to explore a range of behavioral options in a very complex environment. He suggested that the complex environment which has been the spur to the development of consciousness as defined was the social environment of the individual created by the interaction of large human societies. This functional model resembles the suggestion of Pribram (this volume) that consciousness of self derives from a delay between the input pattern and generation of an output pattern regulating behavior.

In a previous section it was pointed out that the ambiguity of information exchange could result in weak associative images. These images require amplification before they could play a strong functional role. Signaling between independent organisms suggests a mechanism for such amplification. Suppose that two hominids have developed to the point at which they can signal the presence of different types of threat (lion, snake, other hominids and so on) for which different behaviors would be urgently required. Sensory input resulting from the presence of a snake results in an action recommendation to shout "snake". To be functionally effective, hearing the word "snake" must generate in the brain of the hearer a set of repetitions similar to those generated by seeing a snake. Such repetitions will produce appropriate behavioral recommendations. Because the speaker can hear his own words, a feedback loop exists.

An early functional application of such a feedback loop could be in tool making. The problem facing the tool maker is how to activate a set of repetitions within his brain that will generate tool making behavioral recommendations. The simplest approach would be to look at an existing tool and use the resultant repetitions to generate tool making actions, in other words copying. In the absence of an existing tool this process is not possible. However, if conditions under which a tool has been used in the past can generate weak associative images of the tool that can be amplified by the tool maker speaking the name of the tool, then tool making in the absence of an original becomes possible. If such a feedback path were internalized physiologically, then the repetition set or mental image could also be focused on parts of the tool for which there was no vocabulary, making much more sophisticated tools possible.

Such feedback paths originally established for tool making would also be appropriate for application to determining behavior in complex social situations. A behavioral recommendation derived directly from sensory input could by analogous feedback activate repetitions imprinted in the past when similar recommendations had been generated. The functional effect of such feedback would be a search through a more extensive range of memory for behavioral guidance, and would be experienced by the system as a constant succession of sensory independent mental images. Repetitions imprinted when the brain was perceiving the behavior of the body or even the behavior of other individuals could become associated into clusters that would be activated by feedback when a current recommendation was similar a past recommendation which resulted in the corresponding body behaviors. Such a mechanism would be experienced as the ability to imagine self.

The framework provided by the recommendation functional architecture thus provides the basis for relating a phenomenological description of consciousness in terms of mental images through intermediate descriptions in terms of ambiguous similarity clusters to neuron level descriptions. In addition, as described in Coward 1997 and summarized in this paragraph, it makes it possible to describe a process of incremental addition of functionality that can be the basis for a theory of the evolution of human cognition.

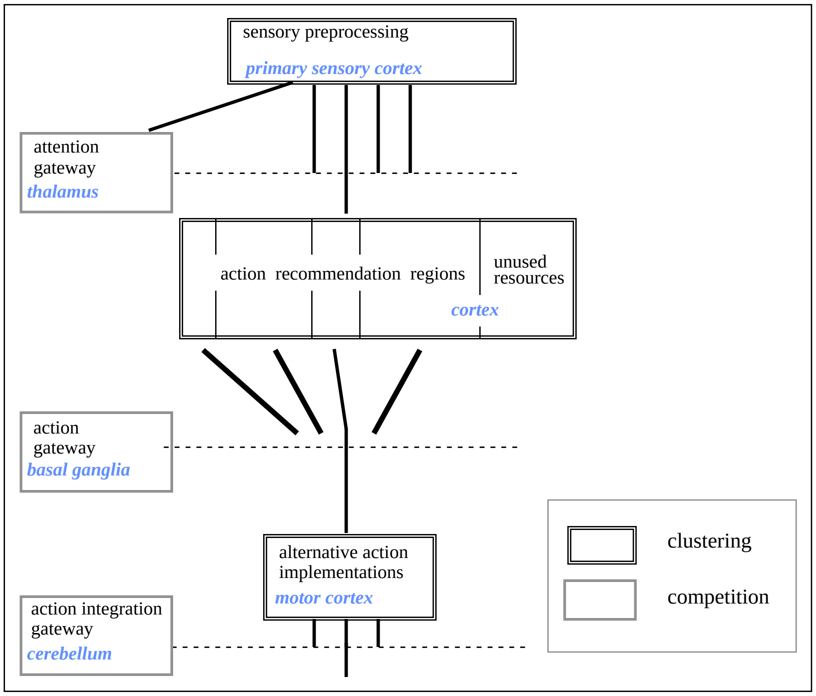

Figure 7. The mapping of major physiological structures in the mammal brain into the recommendation architecture.

Relating Physiology to Psychology

In any system with a functional architecture, functional optimization will tend to result in a physical structure that reflects major functional separations. For example, memory and processing subsystems are ubiquitous in commercial electronic systems. In biological brains, physical separations can be observed which resemble the functional separations of a recommendation architecture. The mapping of major physiological structures into clustering, competition separations proposed in Coward 1990 is shown in figure 7. At a more detailed level, cortex columns functionally resemble similarity clusters (Coward 1990; Tanaka 1993). Layering in the cortex resembles the functional separation into six layers within such clusters discussed earlier. DeFilip (1997) has described strong cross connectivity of spiny stellate cells in layer IV of the cortex, and inhibitive back projection by bouquet cells close to output. This connectivity resembles the functional outputs from γ and β layers discussed earlier. The back projections from higher cortex areas to lower that terminate in layer I, for example in the primary sensory cortex (Cauller 1995) resemble the feedback connectivity required to maintain repetition orthogonality (Coward 1999).

At a psychological level, a sequence of phenomena could be a dog coming within sight of a person, the person noticing the dog and being reminded of a childhood experience of being chased by a dog, and deciding to ignore the dog. A deeper level of description would be sensory input into the eyes of the person beginning to include information from a dog. The information from the dog is interpreted as a coherent sensory domain and competes with other domains for access to the functionality that generates behavioral recommendations. Success in that competition generates multiple behavioral alternatives, some of which are fed back and activate a significant subset of the information recorded in a childhood experience. These additional activations form part of the activation that defines the accepted behavior to ignore the dog. This intermediate level description can be translated into a description in terms of clusters and superclusters, then into a yet more detailed description in terms of neurons, synaptic connections, and neurochemicals. Thus phenomena such as attention, perception, intention, and thought can be mapped to physiological activity using the recommendation architecture approach.

Conclusions

The emerging discipline of software architecture demonstrates the importance of a functional architecture made up of a hierarchy of functional descriptions for functionally complex systems. There are two qualitatively different types of functional architecture, the instruction architecture based on the exchange of unambiguous information between functional components and the recommendation architecture based on the exchange of ambiguous information. Because of the requirement to build, repair and modify electronic systems, such systems are designed to have the instruction architecture. Analogous natural selection pressures have constrained biological brains into the recommendation architecture. The hierarchy of functional descriptions that make up a functional architecture form the basis for a scientific theory of consciousness that relates psychological phenomena to physiology.

References

Anderson, J.A., & Sutton, J.P. (1995). The network of networks model. In *Proceedings of the World Congress of Neural Networks: Washington, DC*, 1, 145-152, Hillsdale NJ: Erlbaum.

Avrunin, G.S., Corbett, J.C., & Dillon, L.K. (1998). Analysing Partially-Implemented Real-Time Systems. *IEEE Transactions on Software Engineering* 24, 8, 602-614.

Card, H.C., Rosendahl, G.K., McNeill, D.K., & McLeod, R.D. (1998). Competitive Learning Algorithms and Neurocomputer Architecture. *IEEE Transactions on Computers*, 47, 8, 847-858.

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

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

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

Coward, L.A. (1997). Unguided Categorization, Direct and Symbolic Representation, and Evolution of Cognition in a Modified Connectionist Theory. In A. Riegler and M. Peschl (Eds.), *Proceedings of the International Conference New Trends in Cognitive Science*, 139-146, Vienna: Austrian Society of Cognitive Science.

Coward, L.A. (1999). A Functional Architecture Approach to Neural Systems. *International Journal of Systems Research and Information Systems*, in press.

De Felipe, J., (1997). Microcircuits in the Brain. In J. Mira, R. Morenzo-Diaz, and J. Cabestanz (Eds.), *Biological and Artificial Computation: from Neuroscience to Technology*, 195-206. Berlin: Springer.

Dijkstra, E.W., (1968), The Structure of the Multiprogramming System. *Communications of the ACM* 18, 8, 453-457.

Harlow, T. M. (1868). Recovery from passage of an iron bar through the head. *New England Medical Society*, 2, 327-346.

Garlan, D., Allen, R., & Ockerbloom, J. (1995). Architectural Mismatch, or Why it's hard to build systems out of existing parts. In *Proceedings of the 17th International Conference on Software Engineering*, 179-185, New York: ACM.

Kohonen, T. (1990). The Self-Organizing Map. *Proceedings of IEEE*, 78, 1464-1480.
Lashley, K.S. (1950). In Search of the Engram. *Symposia of the Society for Experimental Biology* **4**, 454-482.
Parnas, D.L., Clements, P.C., & Weiss, D.M. (1985). The Modular Structure of Complex Systems. *IEEE Transactions on Software Engineering* SE-11, 3, 259-266.
Perry, D.E. & Wolf, A.L. (1992). Foundations for the Study of Software Architecture. *ACM SIGSOFT Software Engineering Notes*, 17, 4, 40-52.
Lorenz, E. (1996). *The Essence of Chaos*. Seattle: University of Washington Press.
Skaggs, W. E. & McNaughton, B. L. (1996). Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience. *Science*, 271, 1870-1873.
Soni, D., Nord, R.L., & Hofmeister, C. (1995). Software Architecture in Industrial Applications. In *Proceedings of the 17th International Conference in Software Engineering*, 196-207. New York: ACM.
Tanaka, K., (1993). Neuronal Mechanisms of Object Recognition. *Science*, 262, 685-688.