Coordination and Control Structures and Processes: Possibilities for Connectionist Networks (CN)

Vasant Honavar & Leonard Uhr
Computer Sciences Department
University of Wisconsin-Madison

Abstract

The absence of powerful control structures and processes that synchronize, coordinate, switch between, choose among, regulate, direct, modulate interactions between, and combine distinct yet interdependent modules of large connectionist networks (CN) is probably one of the most important reasons why such networks have not yet succeeded at handling difficult tasks (e.g. complex object recognition and description, complex problem-solving, planning).

In this paper we examine how CN built from large numbers of relatively simple neuron-like units can be given the ability to handle problems that in typical multicomputer networks and artificial intelligence programs - along with all other types of programs - are always handled using extremely elaborate and precisely worked out central control (coordination, synchronization, switching, etc.). We point out the several mechanisms for central control of this un-brain-like sort that CN already have built into them - albeit in hidden, often overlooked, ways.

We examine the kinds of control mechanisms found in computers, programs, fetal development, cellular function and the immune system, evolution, social organizations, and especially brains, that might be of use in CN. Particularly intriguing suggestions are found in the pacemakers, oscillators, and other local sources of the brain's complex partial synchronies; the diffuse, global effects of slow electrical waves and neurohormones; the developmental program that guides fetal development; communication and coordination within and among living cells; the working of the immune system; the evolutionary processes that operate on large populations of organisms; and the great variety of partially competing partially cooperating controls found in small groups, organizations, and larger societies. All these systems are rich in control - but typically control that emerges from complex interactions of many local and diffuse sources. We explore how several different kinds of plausible control mechanisms might be incorporated into CN, and assess their potential benefits with respect to their cost.

Introduction

Conventional *von Neumann computers*, programming languages, computer programs, and artificial intelligence systems have powerful centralized control structures built into them. But it is difficult to identify control structures in many systems because they are often inextricably linked with structures that serve other functions. Furthermore, such identification often depends on the larger context in which the particular structure

This research was partially supported by grants from the Air Force Office of Scientific Research (AFOSR-89-0178), the National Science Foundation (CCR-8720060), and the University of Wisconsin Graduate School. This paper was published in the Journal of Experimental and Theoretical Artificial Intelligence, 2 277-302 (1990).

interacts with the rest of the system. However, the broadest sense of the term *control* suggests such functions as synchronizing, switching between, choosing among, regulating, directing, modulating or influencing interactions between, selectively combining (e.g., to create new structures from), and coordinating different yet interdependent structural or functional modules, at different levels of organization in natural or synthetic systems.

As we move from rigid, centralized information processing systems like conventional computers to robust, flexible, distributed, adaptive information processing systems like CN, it becomes increasingly difficult to conceptualize the sort of control structures and processes that might be used effectively in the latter. Lyapunov (1963) characterized the evolution of living nature (and hence, one might suggest, that of intelligent behavior) as a process of proliferation of, and selection among, control systems of various types. Similar considerations motivated the early work in *cybernetics* (Wiener, 1961).

The lack of appropriate control structures and processes is a major reason - possibly the most important reason - why CN of large numbers of simple units have not yet succeeded at handling difficult tasks (e.g., complex object recognition and description, complex problem-solving, planning). It is often asserted (but for no special reason, and in most cases incorrectly), that CN do not need such control. Whatever control mechanisms might be necessary are largely expected to emerge as a function of interactions among very simple individual units, and of learning.

Motivations for Using CN for Perception and Thought

There are several reasons why CN offer a very attractive approach toward developing intelligent systems that perceive and think:

* They are loosely brain-like, with their extremely large numbers of very simple primitive units, and the many links between them that build into the total structure.

* They have the potential of being made far more brain-like, by making the primitive units more neuron-like (Shepherd, 1989), and by introducing micro-circuits and successively larger assemblies that incorporate more brain-like structures and processes (Uhr, 1989; Honavar & Uhr, 1990a; 1990b).

* They are general-purpose in the sense of Turing machines, Post productions, and other equivalent universal computers (McCulloch & Pitts, 1943; Pollack, 1987). This means that anything that is describable, and therefore can (with enough work to analyze and develop a clear description) be programmed for a more conventional computer can therefore be handled by some CN.

* They lend themselves to learning, since their micro-modularity means that they can change themselves a little bit at a time, potentially in a way that moves gradually toward adequate behavior (Hinton, 1987a; Honavar & Uhr, 1990a).

Today’s networks can handle only very simple examples of very simple problems. Most have only a few hundred nodes organized into 1- or 2-layer nets. A few much larger networks have been built (chiefly for vision); these are given appropriate larger structures - e.g., local connectivity within and between layers, array connectivity within a layer, converging tree connectivity between layers (Honavar & Uhr, 1987; 1988; 1989; 1990a; Uhr, 1987) - structures that play major roles in organizing and controlling.

Control: Global, Local, and Distributed in Various Ways

Very carefully worked out and precisely coordinated control is absolutely essential to today's conventional computers, and to the programs that are coded for them. CN have virtually none of the large armamentoria of traditional control mechanisms. Whatever control structures that might be necessary are largely expected to emerge as a function of interactions among individual units, and of Learning. But with many independent units and little overall structure, there can be little control unless evolution and learning together manage to discover and carve out such control structures, or elaborate and sufficiently powerful control structures are built into individual units and sub-networks, wherever necessary, supplemented by more global control structures that infest the structures primarily involved with the processing of information.

Traditional techniques for control could actually be incorporated directly into CN. But these simple solutions are often quite costly in terms of complexity, and violate (or appear to violate) what is generally regarded as the spirit of connectionism: Total behavior emerges from the ways that large numbers of individual units behave; processes, memories, and control are distributed and implicit, they should not be precisely hard-wired. CN are generally regarded as parallel, distributed information-processing systems. However, what exactly this means is a matter of considerable debate and discussion. Processing functions and memory, as well as control processes, can each be distributed in space as well as time to various degrees and in various ways. The interactions between each of these aspects of the system can be complex, and can occur on different spatial and temporal scales.

The idea of a single controller, as in the conventional stored-program serial computer's *Central Processing Unit* (CPU) with a central controller that sends out all the commands and allocates all the rescources for executing a program (e.g., "fetch the next instruction from the location indicated; decode it; fetch its operands from the locations indicated; execute this instruction on these data; store the result in the location indicated;" and so on, repeating until the program is finished) seems to violate the spirit of connectionist models. Furthermore, such systems are extremely susceptible to complete breakdown if the central controller malfunctions.

The next sections do the following:

* Define what CN are, and what the particular CN that are used today look like.

* Examine the fact that CN already have several crucially important precise control mechanisms, ones that are largely overlooked because they are handled by the larger system in which the CN is embedded (this is usually a conventional computer program that actually executes the CN, since CN are almost always simulated on a digital computer).

* Examine how control is handled in other types of systems, including: Those - like conventional parallel and serial computers and programming languages - that exercise carefully worked out, detailed, precise, perfect, total control; Those - like nervous systems, fetal development, social organizations, and simple organisms and cells - that have, or appear to have, intricately interlocking sets of local and successively more global controls - but controls that are far more flexible, much looser, and far from perfect; Those - like evolution - where control appears to emerge primarily from individual behavior and local conditions.

- Examine how CN can fruitfully be given various control capabilities of these sorts, and the projected benefits with respect to the actual added complexities.

Connectionist Networks (CN) Defined and Described: Structures, Processes, Evolve-Learn Rules

A connectionist network (e.g., Rosenblatt, 1962; Feldman, 1982; Rumelhart, Hinton & McClelland, 1986) consists entirely of a directed graph, whose nodes, N, send the results of their simple functions (often reminiscent of neurons) over its links, L, into nodes into which they fire.

The following rather general definition (Uhr, 1990a) makes clear that many other than today's standard type of CN can be constructed; it also makes precise comparisons possible.

A CN is a graph (of linked nodes) with a particular topology, where nodes compute several different types of function (*behave, evolve-learn,* and *control*).

CN = {G,B,E,C} (**G** = graph topology, **B** = behave, **E** = evolve/learn, **C** = control)

The total graph is composed of sub-structures of ... sub-structures, down to and starting with the smallest micro-circuits and the primitive units (individual nodes) from which they are built.

Today's CN are specified (typically only partially) as follows:

- The overall graph structure, $\mathbf{G_B}$ of the sub-net that behaves (much of the total graph, including the entire sub-graphs needed to handle learning and control - are usually left unspecified).
- The single function, **B**, that each individual primitive node computes during the net's behave cycle.
- The single function, **E**, that is computed to determine what changes to make during the evolution-learning cycle - but not the actual sub-net structures that are needed to actually compute and make these changes.

A network's behavior is typically initiated by sending values to the input sensing nodes. Its resulting performance is the set of values sent by its output acting nodes. The net's behavior is completely determined by its topology, the values originally associated with its links, and the functions computed by its nodes - plus any modifications made to these by learning.

Several potentially powerful extensions to today's CN - including more powerful structures and processes for processing, control, and learning - suggest themselves leading up to systems that can be characterized as *generalized connectionist networks* (GCN) or *generalized neuromorphic systems* (GNS) (Honavar & Uhr, 1990d).

Behave, Evolve-Learn, and Control Functions

In today's CN, only the function that each primitive behave unit, **B**, computes is completely defined in terms of both the individual node involved and the global graph structure over the total set of nodes. The processes that are used during the behave cycle (**B**), culminating in output performance, usually add, or in other ways integrate, their inputs, then threshold or transform the results with a logistic or other squashing function. They are connected by directed links over which activation values are fired out, and multiplied by weights associated with these links.

There is no compelling reason to limit ourselves to the simple functions that the individual nodes used in today's CN compute. More generally, a large variety of microcircuits (Uhr, 1989; Honavar & Uhr, 1990b) can be used in their place whenever appropriate.

Learning (**E**) is almost always handled as follows: After the network outputs, an error signal (typically the difference between what should have been output and what was actually output) is used to change the network. This is traditionally handled entirely by slightly changing weights associated with links in the direction of reducing that error.

There is nothing sacred about these functions used for learning, and there are a variety of potentially more powerful alternatives. More generally, learning might change not only the weights, but also the behave functions, the evolve-learn rules, and/or the network topology (e.g., by adding and deleting nodes and links (Honavar & Uhr, 1989; 1990a)), or any combination of these.

Some aspects of control are usually assumed in today's CN (see below): All nodes fire in parallel, at discrete instants in time; the behave and learning cycles alternate.

Structures Used to Link Individual Units into Overall Topologies

Any graph topology might be chosen, but only a few have actually been used to date: 1-layer nets where each node that receives inputs from the external environment links directly to all, or to a sub-set (typically random), of nodes that output to the external environment; Multi-layer feed-forward nets where each layer's nodes link into all, or some, of the next layer's nodes; Cycling nets (1- or 2-layer) with complete, random, or near-neighbor mesh connectivity within a layer of nodes. Other topologies are occasionally used, including near-neighbor converging tree-like connectivity from one layer to the next (Honavar & Uhr, 1987; 1988; 1989; 1990a), structures that expedite the particular functions computed, e.g., Hough transforms (Ballard, 1984), and structures (e.g., arrays, trees, pipelines) that are built in because they appear to be appropriate for the particular type of problem being attacked.

When the nodes in CN are structured into arrays, pipelines, trees, pyramids, etc., the connectivity thus established can serve important coordination and control functions. For example, since all the units in an array receive input at the same time, they fire out results at the same instant, and thus initiate an implicit synchrony. Trees can merge, fuse, or combine different streams of information. Delays associated with processing stages of pipelines can be used to re-synchronize the information flow in independent streams.

Control Mechanisms that Are Already Present in Today's CN

Today's CN already have several very powerful (albeit more or less hidden) control mechanisms built in. Since these are almost always handled by the larger environment of programs and computers that execute the CN they can easily be (and typically are) overlooked.

A Central Clock that Exactly Synchronizes All Processors

First, CN almost always have all units fire in exact synchrony, as though they were driven by a global clock. This is easily handled when a traditional computer program is used to actually execute the CN (it is important to remember that today CN are almost always simulated, using a program for a digital computer). Essentially, the serial processor creates a temporary store into which it puts each node's output; finally, when the entire cycle of parallel processes being simulated is completed it uses this store to update the network so that it can start working on the next cycle.

To achieve exact synchrony in a multi-computer network, the conventional computer's hardware is clocked and controlled by a precise oscillator - e.g., a crystal vibrating at nanosecond or gigasecond rates - plus a network of wires to every processor gate in the entire computer over which these clock pulses are sent. Not only is this clocking network large and precisely constructed, it must be handled with great care and precision. For example, the clocking network for a large network of serial computers must be very carefully designed, so that no asynchrony is introduced by the slightly different distances the clock pulse must travel to different modules.

Switching Between Behaving-Performing and Learning

CN always receive input, behave, and output; then they are given feedback (typically the error signal from which they are to learn). This means that there is a precise global switch that is thrown when the network outputs and stops behaving.

To handle that within the net itself, there would be a need for nodes that received inputs from the output nodes, so that they were triggered to fire into additional nodes that in turn initiated and carried out learning. Then, when the learning cycle was completed, there would be a need for nodes that fired into whatever mechanism initiates firings into the input nodes, so that a new problem could be input, and the next cycle begun. That is, the conventional program that simulates the CN replaces the global switching circuitry needed to alternate between behaving and learning.

Handling Input, Output, Generation of the Error Signal, and Learning from Feedback

This makes clear that two other control mechanisms are already built in: to handle output and use it to generate the error signal to be fed back, and to handle input and generate new problems to present the system. These are almost always handled by programs external to the net (or by humans), as follows:

* They constantly watch and monitor, or are triggered by, the network.
* When the network outputs they compare this with some information about what it should have output (the desired output, often called the *target*), and from this generate an error signal (this is usually simply a vector that is the difference between the output and the target).
* When the net finishes its learning half-cycle they produce the next input (e.g., from a memory store, or using a keyboard, TV camera, or other input device).

In addition, all the processes executed during the entire learning half-cycle are actually handled by the perfectly-controlled larger simulation environment, rather than by the

CN itself.

Handling Particular Issues in Particular Simulations

Often, intricate control structures are instantiated in the program that simulates the CN. For example, pattern recognition networks are sometimes constrained to learn a single set of weights for a local mask at a large number of different locations, so as to be able to detect a pattern independent of translation over the input layer (Rumelhart, Hinton & Williams, 1986; Honavar & Uhr, 1988; 1989). *Gates* control and coordinate the flow of signals between different functional modules in CN implementations of *production systems* (Touretzky & Hinton, 1988; Dolan & Smolensky, 1989). The *meta-pi network* (Hampshire & Waibel, 1989) incorporates structures that enable the network to integrate conflict-arbitrated sub-networks for speaker-independent speech recognition. Several simple control structures have been employed to regulate the form and content of the internal representations learned and, to choose from a variety of learning strategies in CN (Honavar & Uhr, 1990a; 1990e; 1990f).

Control and Coordination in Natural and Artificial Systems

Control can vary, from non-existent to complete.

Probably the purest examples of systems in which global control emerges from completely dispersed local control are interacting molecules in gases and liquids. In oceans and atmosphere, locally often there is uncontrolled turbulence and chaos. The relatively regular winds and waves result from larger, more global forces that constrain and to some extent impose order.

The sub-atomic and chemical processes involved in the evolution of the universe, stars, and planets are marvelous examples of how individual primitives (e.g., fields, quarks, strings) can interact and combine to exercise sufficient distributed control to build successively more complex structures, culminating in the most strking examples - living organisms.

At the other extreme, human-made artifacts (e.g, automobiles, computers, etc.) are examples of intricate designs that include precisely built-in mechanisms that execute virtually complete control over everything. The conventional single-CPU serial computer is an example of complete, un-tolerant control over very large numbers of precisely architectured components such as logic gates and flip-flops.

In contrast to the small amounts of control present in today's CN, nature offers many examples of more powerful control. These are almost always far more flexible, adaptive, robust, and loose than the sort found in today's conventional computers and programs. Several such coordination and control structures found in natural and human-made systems that can be potentially useful in CN are examined below.

Complete Control: Traditional Computers and Programs

Possibly the major lesson from all these examples is that rather than being absent control is pervasive. Also of great importance are the facts that in nature control is rarely if ever complete and rigid, and is typically exercised by a complexly interacting structure of interlocking more-or-less local controls. This is often loosely hierarchical, culminating with some very high level control and coordination - possibly it is better to think of them as influences.

In addition to highlighting the crucial importance of (flexible, shared) control, natural systems offer a number of control mechanisms that might fruitfully be incorporated into CN. At the other extreme, today's computers are controlled in much simpler, precise and global ways that also offer attractive possibilities. But it is interesting to note how very different is the simple, complete, and unambiguous control that we build into today's computers from the complex, subtle, diffuse, partial, multi-determined controls found in natural systems.

The traditional single-processor serial computer's central controller effects perfect and complete control and coordination. It fetches one instruction after another for execution by the processor, and shuttles data between memory and processor one *word* at a time. The individual parts of a computer - its memories and processors, and the logic gates from which they are built - are all precisely structured to execute their specific functions without any error (See Hamacher, Vranesic & Zaky, 1984, for details). This structure, plus the information they receive (clock pulses, instructions), completely control and determine behavior. Programs (which from the present perspective are simply specifications of computers that the computers that execute them simulate) are similarly completely controlled.

The kind of control structures and processes found in typical serial computers is completely centralized, and does not seem appropriate for CN. Indeed one of the major attractions of CN is that they move away from such rigid, inflexible control. To implement complete and perfect control, a precise clock must synchronize all operations in all parts of the computer; a tree or other network of control links must carry the clocked instructions to all parts of the computer, thus giving a global clock that completely synchronizes. But all this dramatically illustrates how much in the way of dedicated mechanisms and processes is devoted in computers and programs to ensure complete, perfectly working, control.

As pointed out earlier, when all nodes have to fire at the same time, and when looping through behaving then learning half-cycles, CN depend upon a global clock. This poses major problems: Without the single controller linked everywhere via a dedicated control circuit that the computer on which we simulate a CN provides without our realizing it, how can completely synchronized, simultaneously firing units be handled? It seems likely that the answer is that asynchronous CN should be developed that have no global control mechanisms of this sort. Another alternative is to use networks of clocks or their derivatives to realize the synchronization or coordination that is needed.

If we do indeed use such a central control mechanism, it can handle many other functions at no additional cost - e.g., broadcasting contextual information, gathering global information, triggering specific functions like choosing or outputting special symbols, introducing precisely controlled delays, synchronizing arbitrary sub-sets of activations.

In addition to these kinds of global control, programs and computers have several hierarchies of successively more local controls - e.g., the ways that individual procedures are organized and linked to one another, and the precise ways that individual processors like adders, floating-point multipliers, and high-speed registers - are structured and linked into the central processing unit. These offer a wide variety of suggestions for micro-circuits, and for schemes for linking structures into larger structures, that might well make CN more powerful.

Multi-computers can no longer have a single central controller without major resulting limitations; but only a few relatively rigid types of shared control have been developed to date (See Uhr, 1984; Almasi & Gottlieb, 1989 for details). The most centrally controlled systems have almost always been built in the form of a large 2-dimensional array of computers (SIMD arrays). There is no inherent reason why the individual computers must be linked into a 2-dimensional array; it is simply that only large array problems (e.g., vision, fluid dynamics) appear to obviously lend themselves to this kind of completely synchronized processing. All processors execute exactly the same instruction (which is typically broadcast from the central controller) but each on the unique data stored in its own memory. Although this kind of computer is designed to handle problems posed it (which are almost always problems that fit such a 2-dimensional array), it cannot work without a second "host" computer - one that typically is thought of as somehow separate, and not a part of the multi-computer. The host does such essential things as interacting with users, compiling programs, displaying results, and giving users an adequate programming-debugging environment. Host and array are typically linked so that each can work separately, executing its own processes on separate information; but each can interrupt the other to receive or send.

Other than these single-controller arrays, virtually all the multi-computers that have been designed to date have linked together separate conventional single-CPU serial computers. This means that each has its own separate clock, separate controller, and separate stored program. The completely independent computers thus linked together typically each completely handle their own control. Each can work quite independently of the rest (once it is given the segments of the larger program that it needs by human beings and/or the operating system - which may perform some some of the global control functions that are required), except when it must send or receive information to or from other computers. Transmitting information and interacting with one another raise enormous problems, since the several computers involved must now be re-synchronized and coordinated. One processor must make a request, and at least one other respond. Without going into the complexities involved, today’s computer networks often need an order of magnitude more time to send even the smallest unit of information between computers than the time required by each computer in executing simple processes.

Pipelines organize otherwise independent computers so that each sends its result to the next in the pipe, and all spend exactly the same amount of time (or sit idle until the processor that needs the most time finishes). Pipeline stages can be used to introduce the desired amounts of delay so that information flowing through multiple pipelines can be synchronized or combined as necessary.

Relatively Loose, Dispersed Control: Biological Evolution

Many people think of biological evolution as a large number of independent individuals moving toward structured complexities (see Bonner, 1988 for an elaboration of this view). It has been argued that evolution is a natural and necessary conseqence of biological information systems (e.g., genetic systems, chromosomal organization, cytoplasmic organization) obeying the second law of thermodynamics (Brooks and Wiley, 1988). Large numbers of separate individuals exercise an extremely dispersed control over one another - by competing for resources, reproducing themselves, adapting to particular habitats. But control is by no means entirely dispersed, residing only in each individual. A crucial component is the larger environment's nurturing rewards, debilitating punishments, and niches that invite filling (Bonner, 1988).

The generation of complex self-organizing populations requires mechanisms that expand the combinatorial space in which genetic recombination and reproduction can take place. Among the mechanisms known to produce genetic variation are: the introduction of new DNA from outside the population by viral transmission or hybridization; the change of existing DNA via mutations; the introduction of novel combinations by recombination, chromosomal mutations, or jumping genes (Brooks & Wiley, 1988). The individuals successful in a given environment reproduce. Both sexual and asexual reproduction increase the variety of the gene pool in a population.

Biological evolution has directly influenced the development of genetic or evolutionary learning algorithms (Holland, 1975; Fogel, Owens & Walsh, 1966) in artificial intelligence. In the context of CN that learn to perform complex tasks in a complex environment, evolution-like processes provide an interesting experimental paradigm. For example, several variants (*phenotypes*) belonging to a family of CN (specified by the *genotype*) may inhabit the same environment, which presents tasks to be learned as well as feedback to aid the learning process. Each CN may be provided with a compact encoding of its structure (general topological constraints on connectivity, e.g., connectivity between layers, number of layers, receptive and projective field properties; types of functions that its nodes compute, e.g., sigmoid, and so on). This encoding of CN structure may be subjected to processes akin to genetic recombination and reproduction (e.g., *crossover*, *mutation*, *inversion*), thereby yielding variant CN. Each variant CN is evaluated in terms of how well it learns to perform the tasks presented to it by the environment. The members of the CN population that are better adapted structurally and functionally to interact successfully with the environment survive and proliferate. This adaptation takes place on a much slower timescale than that of *learning*.

Major, But Flexible Control: Genes and Fetal Development

The developmental program is the basic information structure on which evolutionary processes operate. The many studies directed toward understanding the interaction between the environment of the organism at each stage of development and the products of gene expression offer an impressive array of control mechanisms that determine the neural basis of behavior. The genome of the mammal is believed to consist of approximately 10^5 genes. Each gene is a chain of DNA that encodes information needed for synthesis of a structural protein (e.g., that forms a particular ion channel in the neuron) or enzyme. The cooperative relationship between DNA and proteins requires a 4-letter DNA code that is translated into a 20-letter protein code. A single strand of DNA can act as a template for assembling a copy of itself; a single strand of DNA can act as a template for assembling RNA which itself can serve as a template for protein synthesis (Dulbecco, 1987).

Fetal development exhibits an impressive array of precise yet flexible control. The genes' specifications are followed by a coordinated but flexible array of mechanisms, that realize the DNA-encoded design in terms of appropriate molecule building blocks (e.g., proteins that fold to the needed 3-dimensional shape) that are used to build cells and larger organs. At any given moment in the life of a cell, not all genes are transcribed into messenger RNA and expressed as proteins. This differential activity requires complex interactions among enzymes and the materials they recognize and assemble, as well as the larger chemical environment.

The brain with its billions of neurons and synapses is derived from a single egg cell with its two sets of chromosomes. The nature of cellular transformations that give rise to neurons from egg cells is largely unknown. An embryonic cell behaves much like a *self-reproducing automaton* described by von Neumann (1966). The *state* of cell differentiation is described in principle, by the set of *active* genes. If the genome consists of say, $2x10^5$ genes, and if 10^3 of them are active at any time, this gives 10^{2700} possible states for each cell. The sequence of state transitions that actually take place is tightly controlled by the so called *control genes* which can turn on or off other genes (even whole sets of genes) through the action of *gene activator proteins* and *gene repressor proteins* (Alberts, et al., 1983). Local DNA sequences may also be reversibly rearranged to turn genes on and off, thus producing alternating patterns of gene activity.

Generation of specialized cells from less specialized ancestors is a central feature of development. Such differentiation is influenced by at least 3 types of control: instructions contained in the genes; instructions from the cytoplasm of the developing cell, and instructions from other developing cells in the local extracellular environment (Purves & Lichtman, 1985). The emerging view of cell differentiation is that it is the result of a qualitative change in gene expression. Gene expression could be controlled in a number of ways: Genes themselves may be altered by diminution, amplification, rearrangement, or modification; or gene expression could be changed during transcription or translation. It is hard to separate the relative contribution of each of these factors on any particular aspect of cell differentiation (e.g., the choice of particular neurotransmitters in the case of neurons). The prevailing view appears to be that the differentiating cells retain the entire set of genetic instructions during development, and perhaps throughout life. However, a gradual, controlled, restriction of gene expression causes cells to become more and more specialized (Purves & Lichtman, 1985). This raises the possibility that events such as damage to some part of the brain might revoke the restrictions on gene expression in the surviving cells, allowing them to change their properties, so as to be able to take over some of the functions originally performed by the damaged tissue.

Ultimately, all plastic changes in the structure of the brain require the expression of particular genes at particular sites. What are the mechanisms that turn on particular genes? How are the local changes caused by gene expression communicated, and coordinated with changes in the rest of the system? Do encoding, transcribing, and decoding mechanisms such as those found at the level of genes also operate at higher levels in the system - at the level of neurons, local circuits of neurons, functional modules, etc.? Such mechanisms offer a wide range of powerful control structures for CN. For example, in CN that learn by *generation* of connections, nodes, or highly structured microcircuits (Honavar & Uhr, 1989; 1990a), there are forms of control that are extremely useful in communicating structural changes taking place in one subnetwork to the rest of the network. Such control is effected by subnetworks that *encode* the *structural* changes - e.g., the addition of links between a unit and its neighbors in the CN, that transmit the encoded information to decoding subnetworks elsewhere in the CN, which then trigger the appropriate structural changes locally in their regions of influence. An entire subnetwork that is devoted to this and other similar control functions can be embedded in the

CN in a distributed fashion.

Cellular Migration, Differentiation and Pattern Formation

Cells differentiate and form patterns in the embryo under the relatively local control of chemical gradients that are themselves apparently subject to more global control (Gilbert, 1985). Examples ranging from simple one-dimensional animals such as algae to complex three-dimensional structures such as hydra indicate that cells differentiate according to their position in the organism. This suggests the existence of a means of providing information about position, and cellular mechanisms that interpret, and act upon the basis of, such information (Purves & Lichtman, 1985).

The accurate location of neurons and their processes is fundamental to the formation of correct patterns of neuronal connections. In vertebrates, the migration of neuronal precursors from their sites of origin to their final destinations is a crucial step in this process. Some cortical neurons appear to make this journey by crawling along a particular class of glial cells (radial glia), some others do so without help from such glia. Neuronal precursors seem to be guided by cellular and extracellular cues in the matrix that they move through. However, the nature of the map that they must read and interpret in this process is not well-understood. Growing axons travel substantial distances enroute to their targets, apparently guided by a number of directional and positional cues (e.g., mechanical guides, pathways of differential adhesiveness, electrical fields, tropic gradients emanating from targets) that seem to act in concert (Purves & Lichtman, 1985).

Given the range of control that is needed to ensure the proper arrangement, differentiation, and wiring of populations of neurons, the molecular mechanisms of cellular recognition are of great interest. The prevailing view is that cell surface molecules operate by regulating the changes in the probability of interactions between cells without affecting the absolute limits of connectivity. One mechanism exerting macroscopic control over a developing ensemble of cells is the dynamic production of *cell adhesion molecules* (CAMs) (Edelman, 1987). These molecules link cells into collectives whose borders are defined by CAMs of different specificity. The binding properties of cells linked by CAMs are dynamically controlled by the cells themselves as a result of signals exchanged between collectives. CAMs regulate this binding via a series of cell surface modulation mechanisms, including changes in the prevalence of CAMs at the cell surface, in their cellular position or polarity, and in their chemical structure as it affects binding. Cell binding in turn regulates cell motion and further signaling, and thus the resulting forms. Control of the expression of CAM genes by the specific biochemistry affecting CAM regulatory genes at particular morphologic sites assures the constancy of form in a species. However, since the CAMs do not specify the cellular addresses exactly, variability is introduced during development. Extracellular matrix proteins (or *substrate adhesion molecules*, SAMs) can also be expressed at particular times at particular sites and thus alter cellular morphology. These modulation events accompanied by CAM and SAM expression appear to occur in relatively small cell populations in a defined order. Perturbations in CAM binding can lead to altered morphogenesis and altered morphogenesis can lead to changes in CAM expression and modulation patterns. According to the *regulator hypothesis* proposed by Edelman (1987), CAMs and SAMs link the genes and the chemical and mechanical requirements of cellular pattern formation.

Major, but Dispersed Control: Cellular Functions-Metabolism, Cell-to-Cell Communication, Immune System

A variety of proteins found on the surface of cell membranes play a crucial role in cell to cell recognition and adhesion (Alberts et al., 1983). Other proteins that lie within or across the membrane serve as carriers of chemical messages, and receptors for neurotransmitter molecules (and hence recognizers of messages), among other things. *Microtubules* - long, unbranched tubules that are found within the cell bodies of neurons as well as in *axons* and *dendrites* - are suspected to be involved in the transport of material between the cell body and the outlying processes of the neuron. Two general types of *axonal transport* have been identified: a *slow transport* (at the rate of about 1mm per day) and a *fast transport* (at the rate of several hundred mm per day). The potential for powerful signalling built into these transport mechanisms is quite enormous.

Cells of the immune system are sophisticated recognizers of chemical messages (Alberts et al., 1983). These cells, called the *lymphocytes*, serve to defend the animal against foreign microorganisms that may have invaded it. The lymphocytes produce specific *antibodies* that selectively bind to particular molecules on the surface of the invading microorganisms or on the toxic molecules that they produce. The invaders thus branded as foreign are destroyed by being swallowed by *phagocytes*. To recognize a new type of invader, a new antibody needs to be produced. On the other hand, the system must not produce antibodies that bind to its own cells. The vast diversity of antibodies is believed to be generated by random changes in the DNA coding for specific binding sites on the antibody molecules. In this way, through a specialized form of mutation, millions of genetically different lymphocytes are created, each able to proliferate and form a clone whose members all produce the same distinctive antibody. Of these clones, the ones that react with self molecules are destroyed or suppressed while those that make antibodies against foreign molecules are selected to survive and multiply. Thus the genesis of the individual animal’s immune system has clear parallels in the evolutionary processes. (For dynamical models of the immune system embodying learning, memory, and pattern recognition see (Jerne, 1967; Farmer, Packard, & Perelson, 1986)). If the nature of the combinatorial space to be explored by an adaptive system is highly non-linear (and hence not amenable to the use of typical *gradient-descent* strategies), this sort of adaptive production of appropriate templates that recognize contingencies in a changing environment is likely to be very useful in CN.

Many thousands of distinct chemical reactions carried out concurrently in a cell are closely coordinated (Dulbecco, 1987). A variety of control mechanisms regulate the activities of key enzymes in response to the changing conditions in the cell. One very common form of regulation is a rapidly reversible feedback inhibition exerted on the first enzyme of the pathway by the final product of the pathway. A longer-lasting form of regulation involves the chemical modification of one enzyme by another. Combinations of regulatory mechanisms are known to produce long-lasting changes in the metabolism of the cell.

The wide range of predominantly chemical communication taking place at the cellular level in biological systems outlined above suggest corresponding coordination and control processes that might be useful in CN. They can be instantiated in CN by the use of distributed code-book-like mechanisms (Uhr, 1990b) or by marker-passing networks (Hendler, 1989) that encode, transmit, and decode messages in which the markers or tokens play roles analogous to those of the proteins and neurotransmitters that carry chemically encoded messages in living systems. This requires generalizing most of today's CN to handle multiple types of signals (in place of a single scalar valued activation) on multiple time-scales.

Dispersed But Highly Organized Control: The Brain and Its Attendant Nervous System

Brains have a variety of interacting control mechanisms that offer intriguing possibilities for CN. These are especially appropriate, since handling brain-like processes is a major thrust of CN. Brains do not appear to have simple, centralized controls of the sort found in conventional computers. Instead, there appear to be a great variety of both local and global control mechanisms that compete and cooperate with one another.

Neuromodulators, and Relatively Diffuse, Global Controls

The organization of many neuromodulator systems, as they are presently understood, makes them strong candidates for diffuse global control over large areas of the brain. Forebrain in general, and neocortex in particular, are innervated extrinsically by several separate, widely projecting pathways, each of which is associated with a particular neurotransmitter.

These systems are quite distinct from the classical sensory projections to the cortex via the thalamus. For example, cortical norepinephrine originates in the locus coeruleus, which contains only several hundred cells that send slow-conducting axonal connections to almost every region in the central nervous system (Shepherd, 1988), including the amygdala and the hippocampus - both of which have been implicated in memory and learning in higher mammals. Similar separate systems have been identified for several other neurotransmitters, e.g., dopamine, serotonin, acetylcholine, and GABA (See Shepherd, 1988, for details). These projections offer a number of different ways in which discrete subcortical regions can affect the activity of cortical neurons.

One interesting example of the effect of neuromodulators comes from recent studies of the hippocampal formation: Modulation from a subcortical area can actually alter the functional wiring of the hippocampus, making its operation optimally appropriate for a particular behavioral state of the animal. For example, neuronal transmission from the perforant pathway through the dentate gyrus is more effective during slow-wave sleep than when the animal is quiet and alert. This change in transmission is modulated by norepinephrine and serotonin innervation of the dentate gyrus (Squire, 1987).

The activity of neurons in the locus coeruleus is correlated with the level of vigilance of the animal. It has been suggested that locus coeruleus activity prepares the forebrain for processing important stimuli, and as a consequence, that increased vigilance, or attention, leads to more effective learning (Squire, 1987).

Hormones have significant modulatory effects on memory as well. An example is the action of epinephrine (released in the adrenal medulla), which ordinarily influences the degree of retention after a learning experience. Endogenous opioid peptides (the endorphins and enkephalins) and their antagonists have been implicated in the modulation of memory and learning. The opiate antagonist naloxone improves selective attention thereby facilitating better learning (Squire, 1987).

In the context of CN, mechanisms analogous to those of neuromodulators and hormones that operate on a much slower time-scale than the timescales associated with the firing of individual neurons can be instantiated by additional control subnetworks that are embedded in the CN. These can be used to serve as a basis for a variety of functions - e.g., to focus attention, to selectively enhance signals, to initiate, to regulate, and to terminate plastic changes in the network structures, to change behave (**B**) and learn (**E**) functions in specific network modules, and so on.

Pacemaker Cells, and Relatively Regular Oscillators and Synchronizers

Periodic oscillations or rhythms form the basis for various forms of cyclic activity in animal life (e.g., locomotion, sleep, etc). *Circadian rhythms*, that are entrained by the periodic changes in illumination during the 24-hour period appear to govern the timing of specific activities of animals (including mammals) during the day-night cycle. Based on a large body of experimental work, the concept has emerged that circadian systems are composed of multiple oscillators, each with properties to some extent specific and distinct from others (Shepherd, 1988).

Individual pacemaker neurons, that spontaneously fire in a regular, repeated rhythm, appear to serve as local pulsers, or clocks. In the sea slug (Aplysia), such pacemaker neurons are found at the base of the eye. Although the period of the rhythm is maintained close to 24 hours even in the absence of external stimulation, exposure to normal dark-light cycles appears to help regulate, making more precise, the exact period of the rhythm.

Even in the absence of obvious sensory stimulation, and even during sleep, the cerebral cortex shows intense electrical activity, many neurons exhibiting regular *spontaneous firing*. The activity of these impulse generators shows enough temporal regularity to be attributed to oscillators. For example, in Aplysia, bursts of 10 to 20 impulses can be recorded every 5 to 10 seconds from the so called *bursting neuron*. Each burst of impulses is superimposed on an oscillation generating mechanism, a basic *oscillator* or a *pacemaker* whose potential oscillates between two extreme values on either side of the threshold for the nerve impulse. The timescale of changes in the potential of the neuron is about an order of magnitude slower than the timescale of individual impulses. A volley of impulses is fired when the potential is over the threshold and the burst ceases when the potential falls below the threshold. Oscillators using similar principles have been found wherever they have been looked for in a wide variety of nervous systems, from sea slugs to mammals (Changeux, 1984).

Transitions between oscillating and non-oscillating states are quite common in physiological systems (Glass & Mackey, 1988). Such systems have two stable states - oscillating and non-oscillating - and the transition between the two states can be effected by signals extrinsic to the system.

Given one relatively stable primary clock (e.g., the circadian rhythm), other clocks whose periods are multiples or submultiples of the primary clock can easily be derived by means of neuronal or connectionist networks analogous to the frequency multiplier/divider circuits (Kohavi, 1970) used in digital computers. Such networks of clocks can potentially perform elaborate control functions in CN. These functions include synchronizing large populations of units as the resulting waves propagate in space; initiating complex information processing operations during the oscillating state -

the pulses generated serving as control signals that may gate other network modules; effectively instantiating the equivalent of an iterative *while loop* of the sort used in conventional programming languages (a set of processes are active as long as the oscillor pulses are available and are turned off when the oscillator goes into its resting state), and so on. Interacting networks of several such oscillator systems oscillating at different frequencies can effect coordination between information processes at different time-scales.

Global Electrical Waves

It has been known for a long time that EEG waves of electrical activity can be recorded from the brains of animals (including human subjects). The dominant rhythm in the resting subject is at a frequency of 8-13 Hz (termed the *alpha rhythm*), and is most prominent when the recording leads are over the occipetal lobe of the brain (where the primary visual cortex is located). The largest amplitude synchronous waves are present during the deepest sleep (called the *slow-wave* or *S sleep*). A desynchronized EEG signals light sleep (called the *D sleep*), or a state of arousal or waking. The EEG rhythm is believed to arise mainly in the thalamus, as a result of the intrinsic pacemaker property of cells and the properties of synaptic circuits within the thalamus. In the cortex, the thalamic input causes rhythmic synaptic depolarization of the apical dendrites of the cortical pyramidal cells (see Shepherd, 1988, for details).

Arousal (and sleep) are behaviors that affect the entire animal. Arousal seems to be mediated by the *reticular activating system*, stimulated by sensory collaterals, and activated through thalamic nuclei.

Global slow waves may also synchronize, and even serve to spread information and coordinate different functions (e.g., neuron firing, or synaptic changes involved in learning). Certain activity patterns in brain structures have for a long time been suspected to play a role in triggering the synaptic plasticity necessary for learning. Recent studies have found evidence for at least one class of mechanisms that trigger one form of synaptic plasticity - *Long-term potentiation* (LTP). Cells in the hippocampus often fire in very short bursts of about 4 spikes - with the bursts occuring in phase with a 4-7 Hz EEG rhythm (the so-called *theta rhythm*) while an animal (rat) is engaged in learning. Electrical stimulation that mimics the theta rhythm - i.e., bursts that are separated by approximately 200 msec - induce a robust LTP effect provided other conditions for LTP (e.g., nearly simultaneous pre and postsynaptic activity) are met (Lynch & Larson, 1989). The theta rhythm in the hippocampus as well as olfactory cortex is synchronized with the rate at which the rats sample olfactory stimuli by sniffing. It has been suggested that behavior (e.g., sniffing) itself transmits control signals to the structures involved in learning to prepare them for long-term changes in memory.

Synchronized sampling of sensory inputs of the sort described above can potentially be used in CN to integrate information from different sensory modalities. The same clock-like mechanisms may be used to control and coordinate the various network micro-modules involved in behaving and learning (e.g., computing the error for updating the weights).

Control and Coordination of Multiple Information Sources In Brains and Computers

The brain is constantly combining diverse types of information, in what appear to be loosely coordinated yet precise and very effective ways. Thus whole regions of local information are combined by retinal processes involved in detecting spots on contrasting backgrounds, and by cortical processes involved in detecting oriented edges and other features (Kuffler, Nicholls, & Martin, 1984; Shepherd, 1988).

In addition, information is combined from the two eyes, the two ears, and at higher levels the eyes and ears plus other sensory and motor systems. Information from adjacent and more distant parts of a single retinal input is combined as different higher-level features are detected, as is information from the many different processing areas (e.g., in the separate visual pathways of areas that appear to be specialized to process shape, or motion, or color). All these are precisely enough controlled to combine and re-combine with great precision - apparently with mechanisms that, as necessary, adapt to particular situations. But all this takes place without any evidence of the kind of simple central control that is built into today's computers.

Precisely controlled dynamic routing of information (e.g., between processor and memory, between different processors and different memories, between the processor and I/O devices) is essential to the normal functioning of conventional digital computers. The possibility of dynamic switching processes has been suggested in the context of several different aspects of visual perceptual processes in the brain - for example, attention and stereopsis (e.g., Julesz, 1984). But since the complete separation between a conventional computer's stored-program memory, its serial processor, and its various input and output devices is not found in CN (or in brains), switching functions must be highly distributed.

Dynamic switching can easily be accomplished by neurons that are selectively inhibited or de-inhibited (or, alternately, excited) so as to drive their potential below (or above) the threshold of firing, or by dynamically altering the threshold itself.

A *shifter* circuit (Rashevsky, 1960; Anderson & Van Essen, 1987) has been suggested as a means of linking a layer of neurons (e.g., the retinal ganglion cells) with another layer of neurons (say in cortical area V1, via the LGN) in a manner that allows for dynamic shifts in the relative alignment of the two arrays of neurons, but without loss of local spatial relationships. The shifts are produced in increments along a succession of relay stages that are linked by diverging excitatory links. The direction of shift is controlled at each stage by inhibitory neurons that selectively suppress appropriate sets of ascending inputs. To keep the explanation simple, consider a one-dimensional array of values that get shifted to the left or right by a certain amount. At each level l ($l = 1, 2,..$), the left branch of the ascending axon contacts the right dendrite of a cell l steps to its left at the next layer, and the right branch of the ascending axon contacts the left dendrite of a cell l steps to its right at the next layer. At all levels, a given inhibitory cell contacts exclusively left side dendrites or exclusively right side dendrites of the excitatory cells at the same level. Thus, depending on which inhibitory neurons are active and which are inactive, the input array of values gets shifted to the left or right, but will automatically be kept in register for the whole array. (The extension to two dimensions is rather straightforward.) The inhibitory neurons can, in turn, be controlled by top-down feedback as a function of some assessed properties of the input. Such a shifter mechanism is consistent with the known anatomy and physiology of the primate visual pathway, and offers a possible explanation for the existence of the surprisingly large number of cells in the geniculo-recipient layers of the primary visual cortex (Anderson & Van Essen, 1987).

Shifter circuits can serve a variety of functions including: to register inputs from two eyes prior to binocular integration; to prevent motion blur (which would occur otherwise when the images on the retina move constantly) by introducing a compensatory critical shift whose velocity is equal but opposite to the locally measured retinal velocity field; to provide scaling and spatial blurring (e.g., by having shifter circuits hard-wired for several different scales so that information at multiple scales is available for subsequent analysis, or by controlling the activation of inhibitory neurons at each stage dynamically, so as to provide different degrees of spatial averaging); and possibly as a basis for mechanisms that shift attention.

A pyramid-like hierarchy of such shifter circuits could exert very precise control over the particular computations applied to particular segments of the sensory input. Functionally equivalent control structures are used in many pipelined image processing computer systems (e.g., the PIPE (Kent, Schneier & Lumia, 1985)) for programmer-exerted control over the particular processes/transforms applied to particular segments of the image.

Reciprocal connections between layers of neurons in the brain are ubiquitous. Feedback links between layers can serve a number of subtle control functions (Honavar & Uhr, 1990c): The feedback may selectively enhance, inhibit, or in some other way modulate the outputs of a layer of neurons as a function of some assessment that is made at a higher layer. This provides a means of conditional execution of several information processing functions as well as subtle dynamic regulation of network parameters - e.g., the amplification of the signal that is being propagated.

Attentional processes play a crucial role in the perceptual and cognitive functions of the brain. The reticular complex of the thalamus is thought to control the focus of focus of attention or the *attentional searchlight* (Crick, 1984). Almost all the input to the cortex (with the exception of olfactory input) passes through the thalamus. For each projection from a region of the thalamus to a region of the cortex, there is a corresponding projection in the opposite direction. Attention is thought to be controlled by the rapid burst of firing in a subset of thalamic neurons. This burst activates rapidly modifiable synapses (Malsburg synapses) which then assist in the formation of transient *cell assemblies* that represent conjunctions of multiple features (e.g., color, shape) in the sensory input. Such transient cell assemblies can link neural representations of higher level concepts with their lower level constituent sub-concepts in the dynamic fashion required by the psychophysically inspired models of attention (Treisman & Gelade, 1980). Transient cell assemblies also provide one possible solution to the *variable binding problem* in CN.

A model of attention involving a *saliency map* suspected to be located in the parietal lobe or area IT has been proposed by Koch & Ullman (1985). Conspicuous neural activity at particular locations in this map (through the activation of feedback links) results in the identification of the features in different feature maps that are responsible for the neural activity at the attended location in the saliency map.

Feedback inhibition can also enable a system to attend to specific features in the input in an orderly fashion according to the strength of the neural response to each feature. Consider a simple winner-take-all network that results in the inhibition of all the features except the one with the strongest response. After a sufficient delay, feedback inhibition from the neuron with the strongest response shuts off the neurons that contributed to that response. This in turn silences the neuron that originally had the strongest response, and the neuron with the next strongest response gets selected through the winner-take-all mechanism. This process can continue until none of the neurons in the winner-take-all network has sufficient activity to emerge as the winner. This has the effect of gradually masking out the strongest features in the input so that the weaker features get attended. This effect has been observed in models of the olfactory cortex that learn to detect weak hidden odor signals even when they are dominated by much stronger odors (Granger, Ambros-Ingerson, Staubli & Lynch, 1989).

It is likely that there are multiple attentional processes in the brain subserving various forms of attention. Many such processes can be incorporated into CN using reciprocal links between layers of units.

Variegated Controls: Social Organizations

Social organizations have a great variety of sizes and structures, from the smallest 2-individual groups of completely independent organisms, through families, herds, schools, and societies, to the total ecosystem. Some (e.g., armies; assemblylines) have central controllers (e.g., commanding generals, dictators, directors; the motors that drive the line; and the accountants who determine its speed). Others have hierachies of controllers (e.g., bureaucracies and large companies), sometimes with specialties (e.g., foreman, business manager, office manager). In human and animal organizations there are usually additional informal controllers, and central control is not nearly so strong as it appears. And often of great importance are links between friends and acquaintances widely separated in terms of formal control.

Human society has developed the concept of several major different types of organization - from autocractic dictatorship to participatory democracy and anarchy, moving from complete control (in theory - in fact most aspects of people's lives are controlled by many other local forces, or by themselves) to equally shared control, to no control (in theory, since even in the most egalitarian societies individuals always seem to strive to control).

It may well be that greater central control gives greater productivity - if everyone is assigned appropriate complementary work and individuals are in agreement and fully motivated. On the other hand, anarchy in theory maximizes each individual's freedom - including freedom to be productive and creative - although in fact it appears to need completely unselfish members who do not try to pressure, control, or take advantage of one another. But when cooperation is crucial (as when many people row a large boat or work in a factory) things appear to work best when a clear-cut statement is made as to what each person should do, everyone is motivated to do well and further the groups goals, and there are supervisors who try to make sure things run well and decide how to modify things if they do not.

A productive social group typically exercises the following kinds of interlocking functions: Some members work - e.g., gather, grow, and process raw materials; or gather, process and digest information; or make, sell, and distribute products; or think and work out new things to do, and methods for doing them. Other members oversee, organize, and decide what to do, and direct other members. All members exert a great deal of control over what they do themselves, and varying amounts of control over members close to them (both physically and in the group's structure).

Occasionally decisions are made that trigger small or major changes - analogous to shifts of attention in the individual brain. For example, an overseer decides to direct that certain new or different types of information be gathered. Or a manager might decide to change the layout of an assemblyline, or build a new factory. Thus individual units can initiate processes that serve to synchronize and control other units - sometimes only a few, sometimes many. Of course these human units are all highly intelligent, and their processes and the information that they take into account are usually extremely complex.

Each individual is a member of many different interlocking groups. The total society is a complex function of their interactions. The nominal highest-level structures - e.g., the individual nations, treaty blocks, United Nations - are in many ways much less important than much smaller structures like city, workplace, or family.

Control-and-coordination structures akin to those found in social organizations can be useful between distinct but interacting modules of a CN, as well as collections of CN. A framework for coordination and communication among a large number of agents of differing degrees of intelligence has been explored in the *Society of Mind* paradigm for the study of intelligent systems by Minsky (1986), and using the *Ant Colony Metaphor* by Hofstadter (1979). Systematic studies of organizations might suggest mechanisms that might be fruitfully incorporated into CN.

A Summary of the General Types and Loci of Possible Coordination and Control Structures and Processes for CN

There are many different potential types and sources of complete or partial control, including the following:

* Controls can be (and often are) introduced into the user programs or operating systems that the computers that simulate the CN execute. Almost all of today's CN are completely synchronized by a global clock that fires all units at exactly the same time, and are switched regularly between behaving and learning.

* Control can be introduced by building in particular structures and processes as needed - as in specifying particular micro-circuits like winner-take-all nets or decision-trees that make choices and selecctively transmit the appropriate information.

* The simultaneous sensory input of whole fields of closely related or even identical information can initiate a simultaneous flow through the CN that tends to control its processes, and at least to partially synchronize them.

* There can be a complete global controller built into the hardware or/and software - as in the conventional stored-program computers of today - although this appears to violate the spirit of CN (i.e., the distributed nature of control).

* Partial control can be built in globally - e.g., a global clock that synchronizes time - so that all processors execute one step at the same time, then send information to one another, then execute the next step, and so on.

- The topology of the network, sub-net, and local micro-circuits can exercise major control functions - as when a tree of processors successively transforms, combines, and reduces information, or a pipeline of arrays sequentially applies a series of transformations to the input.
- Subtle control can be built into either the simulation or into particular microcircuits that instantiate desired constraints in any of the processes executed by the CN (e.g., pattern recognition networks are sometimes constrained to learn a single set of weights for a local transform irrespective of where the transform is applied).
- Complete (and, if desired, rigid) control of the sort found in conventional computers and multi-computers (e.g., execution of instructions in order; controlled sequences of state transitions) can be built in either centrally or locally at each node or small sets of nodes in the CN.
- A variety of coordination and control structures (e.g., message passing, blackboard structures for messages, instruction broadcasting, multiplexing, conflict resolution) used in multicomputer networks can be built into the CN.
- A host of control mechanisms of the sort found in conventional programming language constructs (conditional execution, loops, etc.) can be built into local or global microcircuits embedded in the CN.
- Completely dispersed control can be attempted, as in evolution. But to make this work it would appear that a number of other (control) mechanisms are needed, like mutations, sexual reproduction, and incentives tending toward cooperation.
- CNs may be provided with compact (gene-like) encodings of their structural and functional properties (e.g., the sizes of receptive fields, general topological constraints on connectivity, etc.). Such encodings may be transformed through genetic operators (e.g., crossover, mutation) to yield variant CN specifications. Environmental rewards and punishments may be used as means of guiding whole CN populations to evolve so as to perform better at tasks presented to them by the environment.
- DNA-like encodings can be incorporated, along with the capability to make copies, linking networks over which these copies can be sent, and decoders to transform these encodings into specifications for network structures and processes to be realized.
- Gene-like information can be used to dynamically specify different types of functional units in CN (analogous to the mechanisms of cellular differentiation). Controls can activate or suppress the expression of different functional properties in CN nodes or node ensembles.
- Local interactions of the sort found on the surface of cells that are bound by cell adhesion molecules might be used to determine how to build single units and larger micro-circuits into successively larger structures (e.g., through a specification of how the microcircuits are to be assembled together).
- The immune system’s rapid, evolution-like adaptations suggest the possibility of triggering massive proliferations of a range of mutation-like variants, and then as a function of local feedback, to trigger processes (that might probably de-generate and discard the node(s) or microcircuits deemed useless for the system’s function)

to converge on a small subset of the population.

* Independent entities like chemical markers and pilot cells can - guided by information of the sort found in chemical gradients and lock-in-key-type templates - combine to build well-organized wholes.

* The complex interactions between chains of enzymes suggest the possibility of mechanisms that can modify, build, and recycle network structures in a self-sustaining manner.

* Multi-messenger pathways analogous to those supported by intra- as well as inter-cellular communication (e.g., axonal transport mechanisms, membrane proteins) can be built into and among individual CN units and microcircuits of units.

* Several different types of global sub-nets can be used, tailored to and serving different purposes - much like the brain's neurotransmitters, global electrical waves, and chemical messenger systems. Such systems can be embedded in the CN using token-passing networks or codebook-like structures.

* Neuromodulator-like influences can be achieved by incorporating linking sub-networks that contain links with different amounts of delays, transmit information to changes thresholds, to alter network plasticity, and to switch between alternate possible functions.

* Modulatory networks analogous to hormones can now have relatively diffuse, slow-acting effects on such global processes as memory storage, memory modification or selective recall of stored memories.

* Regulatory subnetworks can also be used to initiate, modulate, and terminate plasticity of specific CN modules in a controlled fashion during learning.

* Specific control subnetworks can be embedded into the CN to instantiate processes like attention - selectively enhancing or attenuating the relative contributions of different aspects of the environmental stimuli.

* A variety of simple controls that regulate various aspects of learning (including the form and content of the learned representations), alter vigilance, and choose between different learning strategies can be incorporated into CN

* A variety of contextually driven switching mechanisms can be built into the CN to alter, in a dynamic fashion, the functions of different CN modules or the interactions among modules.

* Pacemaker units that emit pulses regularly can serve to drive relatively local clocks and build oscillators, which in turn trigger and synchronize larger sub-structures. Several local clocking networks can overlap, and interact in such a way as to partially or totally re-synchronize one another.

* Oscillators can be used to handle a variety of important problems - e.g., to switch between processes; to decide when to initiate, or to terminate, a process; to sample the environmental input at a desired rate.

* Clocks or networks of clocks can execute the equivalent of a *while-loop* in conventional programming languages in CN. For example, information can cycle through several interior layers until a decision net is triggered - which in turn fires into nodes, switching them on so that they in turn fire out in synchrony to other regions

of the net.

- Synchronized sampling of environmental stimuli in different sensory modalities can be used as a means of multi-sensory integration.
- Network structures that initiate, transmit and terminate global wave-like signals to large regions of the network can instantiate arousal-like processes or help prepare entire network modules to better process the incoming signals.
- Multi-level shifter-like structures embedded in the CN can be used for a variety of control functions (e.g., to register inputs from two visual sensors, to compensate for the motion of objects in the scene, to dynamically control the degree of smoothing to suppress the noise in the input).
- Feedback pathways can be built into CN to subserve a number of subtle control functions (e.g., selective modulation of the sensory signals as a function of some assessments made at the higher levels, selective attention to specific aspects of the environmental stimuli ordered by their salience).
- Control structures that dynamically link subnetworks that are activated by different features in the environmental stimuli into transient network assemblies can serve a host of useful functions (e.g., dynamic variable binding necessary for complex inferences) in CN.
- Specific subnetworks can be built into the CN that ensure a proper balance in the allocation of different network resources (e.g., nodes, links, long-range communication networks) among different functions as the network learns and evolves.
- Network controls can dynamically alter the incentives for CN units, microcircuits or functional modules to cooperate (as opposed to compete) with each other.
- Several simple mechanisms can be incorporated into CN to tune the various control structures in response to the changing needs imposed by the environment.
- Networks of partial and overlapping controls in a CN can be made to evolve over time by having the sphere of influence of each control structure expand with success and contract with failure (analogous to what takes place in societies). Then when these inevitably overlapping controls conflict, additional local controllers might resolve their differences.

Summary and Concluding Remarks

CN built from sufficiently large numbers of relatively simple - and independent - units are general-purpose, hence potentially capable of doing anything that any possible artificial intelligence program can do. But without adequate control structures, CN - just as a haphazardly-put-together sequence of instructions for a conventional computer - will accomplish little.

There are several potential sources of control that can be given to the CN: the processes carried out by the individual units, microcircuits of units, or entire subnetworks dedicated to control; the larger structures that link and organize the units; and learning plus evolution under environmental influence. We have examined some of the ways that these can fruitfully be used to improve performance by adding the appropriate control structures as necessary. In doing so, we have relied upon a variety of natural and artificial systems for suggestions.

A priori, it is difficult to determine exactly which subsets of the suggested control structures are appropriate for a given system, or which of them work well together in practice. These questions can only be answered through extensive empirical investigations on a large number of increasingly challenging perceptual and cognitive tasks. To do this effectively, appropriate network simulation environments and high-level network specification languages need to be developed.

We have tried to give a picture of the great variety of global and local control mechanisms that exist in nature and can be incorporated into CN. This picture is necessarily incomplete - because brains and other natural systems are far from being completely understood. But - at least until our understanding of CN has progressed to the point where we can achieve nets that begin to mirror the brain in its complexity - only a very small subset of the controls outlined here should probably be used in any particular CN. The range of possibilities presented here should be considered as a source of suggestions for potentially useful mechanisms. Even though brains and other natural systems make simultaneous use of many of these, today's CN are not yet in a position to do so.

It is important to emphasize that any and all of these mechanisms can be realized as sub-nets in CN, since CN are general-purpose. The added control structures at worst increase the size of the CN by a small constant factor. In practice, the potential benefits in terms of power, flexibility, and adaptability may well far outweigh the costs.

Even the least plausible type of control - the conventional computer's completely dominant and dictatorial central controller that handles everything, including fetching and interpreting a stored program - should be considered. In fact a good experimental tactic might be to get the CN to adequately handle some non-trivial problem of interest - that is, to exhibit as much power as possible - and then replace such excessive control by successively less rigid and more plausible mechanisms, getting rid of undesirable built-in black boxes. For some problems, extremely large, elaborate, and meticulously crafted networks appear to be needed; whereas other problems can be handled with only minor changes and small additions to today's systems. When it is not known how to realize a particular mechanism with a plausible net, it can be treated as a black box, and the task of compiling it into a net might be attacked using techniques of linear and nonlinear control system synthesis (Kailath, 1982; Nagrath & Gopal, 1982; Narendra & Annaswamy, 1988).

Whether the net is "plausible" or not is a subtle issue, tantamount to assessing the goodness of a model. Nature offers an abundance of control structures and processes everywhere, in many guises - in contrast to today's relatively impowerished CN. There is little reason to not embed the kinds of structures and processes found in the brain, or in a host of natural and man-made systems in CN - especially if doing so adds to their flexibility, versatility, and power at modest additional cost.

References

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., & Watson, J. D. (1983). *Molecular Biology of the Cell.* New York: Garland.

Almasi, G. S., & Gottlieb, A. (1989). *Highly Parallel Computing.* New York:

Benjamin/Cummings.

Anderson, C. H., & Van Essen, D. C. (1987). Shifter Circuits: A Computational Strategy for Dynamic Aspects of Visual Processing. *Proceedings of the National Academy of Sciences, USA.* 84, 6297-6301.

Ballard, D. H. (1984). Parameter Nets. *Artificial Intelligence,* 22, 235-267.

Bonner, J. T. (1988). *The Evolution of Complexity by Means of Natural Selection.* Princeton, NJ: Princeton University Press.

Brooks, D. R., & Wiley, E. O. (1988). *Evolution as Entropy: Toward a Unified Theory of Biology.* Chicago: The University of Chicago Press.

Changeux, J-P. (1984). *The Neuronal Man.* New York: Pantheon Books.

Crick, F. (1984). Function of the Thalamic Reticular Complex: the Searchlight Hypothesis. In: *Proceedings of the National Academy of Sciences, USA.* 81, 4586-4590.

Dolan, C. P., & Smolensky, P. (1989). Tensor Product Production System: A Modular Architecture and Representation. *Connection Science,* 1, 53-68.

Dulbecco, R. (1987). *The Design of Life.* New Haven, CT: Yale University Press.

Edelman, G. M. (1987). *Neural Darvinism.* New York: Basic Books.

Farmer, D. J., Packard, N. H., & Perelson, A. S. (1986). The Immune System, Adaptation, and Machine Learning. *Physica,* 22D 187-204.

Feldman, J. A., & Ballard, D. H. (1982). Connectionist Models and their Properties. *Cognitive Science,* 6, 205-254.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial Intelligence Through Simulated Evolution.* New York: Wiley.

Gilbert, S. F. (1985). *Developmental Biology.* Sunderland, MA: Sinauer Associates.

Glass, L., & Mackey, M. C. (1988). *From Clocks to Chaos: The Rhythms of Life.* Princeton, NJ: Princeton University Press.

Granger, R., Ambros-Ingerson, J., Staubli, U., & Lynch, G. (1989). Memorial Operation of Multiple, Interacting, Simulated Brain Structures. In M. Gluck & D. Rumelhart (Eds.) *Neuroscience and Connectionist Theory.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Hamacher, V. C., Vranesic, Z. G., & Zaky, S. G. (1984). *Computer Organization.* Singapore: McGraw-Hill.

Hampshire, J. B., & Waibel, A. H. (1989). *The Meta-Pi Network: Building Distributed Knowledge Representations for Robust Pattern Recognition.* Technical report CMU-CS-89-166. Pittsburgh, PA: Carnegie Mellon University, Computer Science Dept.

Hendler, J. (1989). The Design and Implementation of Marker-Passing Systems. *Connection Science,* 1, 17-40.

Hinton, G. E. (1987a). *Connectionist Learning Procedures.* Technical report CMU-CS-87-115. Pittsburgh, PA: Carnegie Mellon University, Computer Science Dept.

Hofstadter, D. R. (1979). *Godel, Escher, Bach: An Eternal Golden Braid.* New York: Basic Books.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Honavar, V., & Uhr, L. (1987). *Recognition Cones: A Neuronal Architecture for Perception and Learning.* Technical report 717. Madison, WI: University of Wisconsin, Computer Sciences Dept.

Honavar, V., & Uhr, L. (1988). A network of neuron-like units that learns to perceive by generation as well as reweighting of its links. In G. E. Hinton, T. J. Sejnowski, & D. S. Touretzky (Eds.) *Proceedings of the 1988 Connectionist Models Summer School.* San Mateo, CA: Morgan Kaufmann.

Honavar, V. (1989). *Perceptual Development and Learning: From Behavioral, Neurophysiological, and Morphological Evidence to Computational Models.* Technical report 818. Madison, WI: University of Wisconsin, Computer Sciences Dept.

Honavar, V., & Uhr, L. (1989). Generation, Local Receptive Fields and Global Convergence Improve Perceptual Learning in Connectionist Networks. In *Proceedings of the 1989 International Joint Conference on Artificial Intelligence.* San Mateo, CA: Morgan Kaufmann.

Honavar, V., & Uhr, L. (1990a). Brain-Structured Connectionist Networks That Perceive and Learn. *Connection Science,* 1, 139-159.

Honavar, V., & Uhr, L. (1990b). Analog and Hybrid Analog-Digital Microciruits for Connectionist Networks. Paper in preparation.

Honavar, V., & Uhr, L. (1990c). On the Role of Feedback Links in Connectionist Networks. Paper in preparation.

Honavar, V., & Uhr, L. (1990d). Generalized Connectionist Network Structures and Processes for Intelligent Systems. Paper in preparation.

Honavar, V., & Uhr, L. (1990e). Novelty-Driven Generative Learning in Connectionist Networks: The Search for Useful Internal Representations of the Environment. Submitted for publication.

Honavar, V., & Uhr, L. (1990f). Successive Refinement of Multi-resolution Representations in Connectionist Networks. Paper in preparation.

Jerne, N. K. (1967). Antibodies and Learning: Selection Versus Instruction. In G. C. Quarton, T. Melnechuk, & F. O. Schmitt (Eds.) *The Neurosciences: A Study Program.* New York, NY: Rockefeller University Press.

Julesz, B. (1984). Towards an Axiomatic Theory of Preattentive Vision. In G. M. Edelman, W. E. Gall, & W. M. Cowan (Eds.) *Dynamic Aspects of Neocortical Function.* New York: Wiley.

Kent, E. W., Shneier, M. O., & Lumia, R. (1985). Pipelined Image-Processing Engine. *Journal of Parallel and Distributed Computing.* 2, 50-78.

Kailath, T. (1982). *Linear Systems.* Englewood Cliffs, NJ: Prentice Hall.

Koch, C., & Ullman, S. (1985). Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. *Human Neurobiology*, 4, 219-227.

Kohavi, Z. (1970). *Switching and Finite Automata Theory.* New York: McGraw-Hill.

Kuffler, S. W., Nicholls, J. G., & Martin, A. R. (1984). *From Neuron to Brain.* Sunderland, MA: Sinaur Associates.

Lyapunov, A. A. (1963). Directional Systems of a Living Nature and the General Understanding of Living Processes. In *Problemy Kibernetiki (Problems of Cybernetics),* Moscow, USSR: Mir Publishers.

Lynch, G., & Larson, J. (1989). Long-term Potentiation and the Mechanism of Memory. (Preprint).

McCulloch, W. S., & Pitts, W. H. (1943). A logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 115-133.

Minsky, M. (1986). *Society of Mind.* New York: Basic Books.

Nagrath, I. J., & Gopal, M. (1982). *Control Systems Engineering.* New Delhi, India: Wiley.

Narendra, K. S., & Annaswamy, A. M. (1988). *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice Hall.

Pollack, J. (1987). Technical report MCCS-87-100. Las Cruces, NM: New Mexico State University Computing Research Laboratories.

Purves, D., & Lichtman, J. W. (1985). *Principles of Neural Development.* Sunderland, MA: Sinaur Associates.

Rashevsky, N. (1960). *Mathematical Biophysics: Physico-Mathematical Foundations of Biology. Vol. 2.* New York: Dover.

Rosenblatt, F. (1962). *Principles of Neurodynamics.* New York: Spartan Books.

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1986). A General Framework for Parallel Distributed Processing. In *Parallel Distributed Processing- Explorations in the Microstructure of Cognition. Vol. 1: Foundations.* Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing- Explorations in the Microstructure of Cognition. Vol. 1: Foundations.* Cambridge, MA: MIT Press.

Shepherd, G. M. (1988). *Neurobiology.* New York: Oxford University Press.

Shepherd, G. M. (1989). The Significance of Real Neuron Architectures for Neural Network Simulations. In E. Schwartz (Ed.) *Computational Neuroscience,* Cambridge, MA: MIT Press.

Squire, L. R. (1987). Memory: Neural Organization and Behavior. In V. Mountcastle, F. Plum, S. R. Geiger (Eds.) *Handbook of Physiology: The Nervous System. Vol. 5.* Bethesda, MD: American Physiological Society.

Touretzky, D. S., & Hinton, G. E. (1988). A Distributed Connectionist Production System. *Cognitive Science,* 12, 423-466.

Treisman, A., & Gelade, G. (1980). A Feature-Integration Theory of Attention. *Cognitive Psychology,* 12, 97-136.

Uhr, L. (1984). *Algorithm Structured Computer Arrays and Networks.* New York: Academic Press.

Uhr, L. (1987). Highly Parallel, Hierarchical, Recognition Cone Perceptual Structures. In L. Uhr (Ed.) *Parallel Computer Vision.* New York: Academic Press.

Uhr, L., (1989). Specifying Useful Sub-net and Micro-circuit Structures for Connectionist Networks. Paper in preparation.

Uhr, L., (1990a). Connectionist Networks Defined Generally, to Show Where Power Can be Increased *Connection Science,* (To appear).

Uhr, L., (1990b). Handling Lists and Structures of Lists in Connectionist Networks. Paper in preparation.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata.* A. W. Burks (ed). Urbana-Champaign, IL: University of Illinois Press.

Wiener, N. (1961). *Cybernetics: or Control and Communication in the Animal and the Machine.* Cambridge, MA: MIT Press.