February 1991

Dynamic Binding Communication Mechanism

Jeffrey S. Aaronson
University of Pennsylvania

Dynamic Binding Communication Mechanism

Abstract

Shastri & Ajjanagadde have proposed a biologically plausible connectionist rule-based reasoning system (hereafter referred to as a knowledge base, or KB), that represents a dynamic binding as the simultaneous, or in-phase, activity of the appropriate nodes [9]. This paper makes the first attempt at designing a biologically plausible connectionist interface mechanism between 2 distinct phase-based KB, as the next step toward providing a computational account of common-sense reasoning. The Dynamic Binding Communication Mechanism (DBCM) extracts a dynamic binding from a source KB and incorporates the binding into a destination KB so that it is consistent with the knowledge already represented in the latter. DBCM consists of several distinct, special-purpose modules. The Binding Memory (BM) is made up of several identical banks of nodes. Each time a temporally-encoded dynamic binding is extracted from the source KB, it is transferred into one of the banks, where the binding is converted to a spatially-encoded representation. The Phase Database (PD) monitors the target KB and produces a phased output that is out of phase with all other nodes in the target KB. The Phase Allocator (PA) synthesizes information from the Phase Database and from the target KB to determine the phase in which to introduce the new dynamic binding into the target KB. In turn, the PA extracts a single binding from one of the banks in the BM and introduces it into the target KB. The interface also utilizes 2 searchlight mechanisms: the first governs which bank in the BM receives bindings; the second mediates between the active banks (those which are currently representing bindings), and the Phase Allocator.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-91-16.

Dynamic Binding Communication Mechanism

MS-CIS-91-16
LINC LAB 196

Jeffrey S. Aaronson

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

February 1991

Dynamic Binding Communication Mechanism

Jeffrey S. Aaronson
Computer and Information Science Department
University of Pennsylvania
Philadelphia, PA 19104

February 1991

Abstract

Shastri & Ajjanagadde have proposed a biologically plausible connectionist rule-based reasoning system (hereafter referred to as a knowledge base, or KB), that represents a dynamic binding as the simultaneous, or in-phase, activity of the appropriate nodes [9]. This paper makes the first attempt at designing a biologically plausible connectionist interface mechanism between 2 distinct phase-based KB, as the next step toward providing a computational account of common-sense reasoning. The Dynamic Binding Communication Mechanism (DBCM) extracts a dynamic binding from a source KB and incorporates the binding into a destination KB so that it is consistent with the knowledge already represented in the latter. DBCM consists of several distinct, special-purpose modules. The Binding Memory (BM) is made up of several identical banks of nodes. Each time a temporally-encoded dynamic binding is extracted from the source KB, it is transferred into one of the banks, where the binding is converted to a spatially-encoded representation. The Phase Database (PD) monitors the target KB and produces a phased output that is out of phase with all other nodes in the target KB. The Phase Allocator (PA) synthesizes information from the Phase Database and from the target KB to determine the phase in which to introduce the new dynamic binding into the target KB. In turn, the PA extracts a single binding from one of the banks in the BM and introduces it into the target KB. The interface also utilizes 2 searchlight mechanisms: the first governs which bank in the BM receives bindings; the second mediates between the active banks (those which are currently representing bindings), and the Phase Allocator.

1 Introduction

Shastri & Ajjanagadde have suggested a computational account of how a human agent, seemingly without conscious intervention, can perform a broad class of inference with remarkable efficiency (within a few hundred milliseconds) [9]. They describe this type of reasoning as *reflexive* reasoning, since the inferences seem to be drawn automatically and effortlessly [8]. Reflexive reasoning (sometimes referred to as common-sense reasoning) proficiency is independant of the size of the body of knowledge upon which the reflexive inferences are drawn. If this were not the case, then it would take you longer to understand this sentence today than it did 5 years ago! As argued in [9], this strongly suggests that reflexive reasoning is the manisfestation of a massively parallel reasoning system in which response time is governed by the length of the chain of inference, and that multiple potentially relevant inference paths are explored simultaneously. In light of results from complexity theory, this implies that human agents are not general-purpose reasoners, as *all* valid inferences cannot be derived proportional to the length of the proof. The connectionist knowledge base described by Shastri & Ajjanagadde not only offers a biologically plausible, computational account of reflexive reasoning, but suggests a sufficient, limited class of inference by which human agents *do* reason.

A brief description of their system follows. Each constant and predicate argument in the KB is represented by a unit. Long-term, or static, facts, are physically encoded as an ensemble of units with structured interconnections that bind together the appropriate filler/role pairs. Rules are represented as a mapping between arguments of the antecedent predicate(s) and arguments of the consequent predicate. As such, there is a physical link reflecting each argument mapping for every rule. Inferences are drawn as a result of activation spreading through the system in parallel, by way of the links between predicate arguments. During each episode of reasoning, many transient facts are created, propagated and subsequently discarded at the end of the episode. Any attempt at modelling reflexive reasoning must have a method of creating and propagating these dynamic bindings (see [9] for for an in-depth discussion) that allows the inference process to perform in real-time, without succumbing to a complex form of cross-talk called the *variable binding problem.* A hard-wired solution to this problem is adopted in order to represent *static*, *long*-term facts. However, this method is not a feasible means of representing *dynamic* bindings in *short*-term memory. Consider that an episode of reasoning takes on the order of a few hundred milliseconds. This suggests that the creation of the new bindings introduced by each rule application can take only on the order of tens of milliseconds. It has yet to be shown that synaptic modification occurs anywhere near this quickly. Hence, reflexive reasoning does not permit us the luxury of allowing synaptic modification to recruit new units and build up the necessary connectivity "on the fly".

The solution chosen by Shastri & Ajjanagadde represents a dynamic binding as the simultaneous, or in-phase, activity of the appropriate nodes [9]. Note that a system in which nodes characteristically fire in short slices at a given frequency does not require a controlling global clock. Rather, the only requirement is that we have phase-sensitive units that fire at a fixed frequency. Thus the inference mechanism consists of a rhythmic spread of activation through the network, each set of bindings "lighting up" during its distinct phase within a fixed length period, dynamically creating a set of bindings with more member units as the inference process continues. The analogous *spatial* solution would be realized by creating ρ copies of each unit in the KB, one for each of the ρ phases. If a unit were active in the former system in phase i, then the i^{th} copy of the unit would be active in the latter system. While this method would increase the size of the KB by a factor of ρ, the speed of the system presumably would increase by a factor of ρ as well. Were we not concerned with a biologically plausible system, this might well be an attractive tradeoff. However, the firing characteristics of neurons are such that they fire in short bursts followed by a longer period of latency. Thus, by adopting the temporal synchrony approach we reduce the size of the KB without sacrificing speed. Furthermore, there is mounting biological evidence that neurons fire at a fixed frequency and that suggests that we encode the unity of objects via the proposed temporal method [4, 5, 2, 3].

The bottleneck in such a *phased* system is ρ, the ratio between period duration and phase duration. This ratio determines the number of distinct individuals that that can be attuned to during any episode of reasoning. Furthermore, this ratio is KB dependant; different KB, with different characteristic ratios, will be able to support a different number of distinct individuals at one time during an episode of reasoning. It is important to note that the number of predicate arguments, or roles, to which an individual can be dynamically bound is unlimited. Hereafter, a binding will refer only to a single entity filling a single role. A set of bindings involving the same entity, will be referred to as a Binding Set (BSet) Thus, the characteristic ratio determines the number of temporal BSets that may be supported during an episode of reasoning. Each Binding Set may include an arbitrary (unlimited) number of bindings.

2 Dynamic Binding Communication Mechanism (DBCM)

The animal brain is not an amorphous glob of wetware. Rather it is a highly-structured, modular architecture in which specific areas are delegated toward processing one type, or one form, of information. For example, consider the sense of sight and the sense of hearing. Each of these different forms of perception has its own unique, special-purpose region of the brain where sensory input of only that type is processed. The visual system solely processes visual data, and the auditory system solely processes auditory data. Despite this clear seperation, intelligent agents have the ability to draw reflexive inferences that can only be reached as a result of an integration of data from distinct sensory processing "stations". An interface mechanism that is able to communicate between 2 distinct repositories of information (i.e. the "visual" KB and the "auditory" KB) must be a component in not only the reasoning process routinely performed by an intelligent agent, but must be a component in the *reflexive* reasoning process routinely performed by an intelligent agent. A typist glancing down at his keyboard is reflexively aware of the fact that the repetitive clacks he hears are the result of the depression of keys on his keyboard. This conclusion could not have been reached unless there was some mechanism in place that allowed his visual station to confer with his auditory station, with the requisite efficiency that enables him to draw such inferences reflexively. The work reported here is an attempt to begin to provide a biologically plausible, computationally efficient, explanation of this phenomenon.

This paper details a biologically plausible communication mechanism, operating between distinct knowledge bases, each with their own phase structure, that temporally encode BSets as described at the beginning of the introduction, as the next step towards a computational account of reflexive reasoning. As the result of an internally generated, phased *extractor* signal within the source KB, the Dynamic Binding Communication Mechanism (DBCM) extracts the dynamic bindings represented in that phase from the source KB and introduces this BSet into a destination KB. The phase in which the new BSet is encoded is consistent with the existing phases already assigned in the destination KB. DBCM consists of several distinct, special-purpose modules. See figure 1. The input buffer is made up of several identical banks of nodes. These banks collectively make up the Binding Memory (BM). Each time a temporally-encoded BSet is extracted from the source KB, it is transferred into the BM, where it is converted to a spatially-encoded representation in one of its banks. The Phase Database (PD) monitors the target KB and produces a phased output that indicates a currently unused phase within the target KB. The Phase Allocator (PA) synthesizes information from the PD and from the target KB to decide in which phase to introduce a new

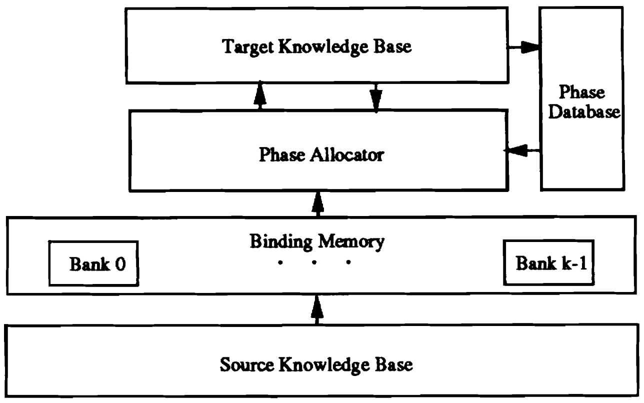

Figure 1: Global View of Dynamic Binding Communication Mechasnism

BSet into the target KB. It then extracts the BSet from the BM, temporally encodes the bindings using the chosen phase, and introduces the resulting temporal BSet into the target KB.

The interface also utilizes 2 searchlight mechanisms: the first searchlight enables only a single bank in the BM to be a candidate repository of a temporal BSet at a time; the second mediates between the active banks (those which are currently representing bindings) and the PA.

2.1 Terminology

All links in DBCM are unidirectional. Unless otherwise specified, links should be considered excitatory in nature. There are several types of units that make up DBCM:

ρ-btu ρ-btu nodes take the weighted sum of their inputs. If during a given phase, this total meets or exceeds the threshold of the node, then the node responds in that phase during the next period with a pulse. It continues to fire in that phase as long as the driving input persists. ρ-btu nodes are represented diagramatically by a circle.

τ-or τ-or nodes take the weighted sum of their inputs. If during a given phase, this total meets or exceeds the threshold of the node, then the node responds for the complete duration of the next period with a pulse. τ-or nodes are represented diagramatically by a triangle.

τ-or latch τ-or latch nodes are identical to τ-or nodes, except that once active, they fire steadily for several cycles (periods). τ-or latches are represented diagramatically by a bold triangle.

2.2 Searchlight Mechanisms

The searchlight mechanisms implemented here is slightly different in intent that that spoken of by Crick [1] and Treistman & Gelade [10]. They speak of an attentional searchlight mechanism in the brain that is used to highlight some small subsection of an input field. The hypothesis is that at any given point in time, while we are aware of the image portrayed over the entire input field, we are focusing our attention on the small section that is lit up by the beam of the searchlight. After some period of time, the beam of the searchlight moves and highlights another portion of the input space.

Both searchlight mechanisms used by DBCM differ from the above description, as well as from each other. Most importantly, the DBCM searchlights do not (necessarily) have any overt attentional significance, but rather, an internal attentional function. Despite their differences, they all share the idea of highlighting some subsection of a field, which becomes the focus of attention as processing is concentrated within the area lit up by the beam. After the processing is complete, the beam moves on to the next region within the field. The first searchlight mechanism directs the destination of each temporal BSet that is being extracted from the source KB into one of k banks in the Binding Memory. The second searchlight mechanism restricts the flow of bindings from the BM into the Phase Allocator, enforcing that the PA process information from only a single bank at a time (i.e. the quantum of data on which the PA operates is a Binding Set).

3 Architecture

As mentioned earlier, the source KB is responsible for initiating a communications request by triggering a special *extractor* unit, which indicates by its phased firing which BSet is to be extracted from the KB. DBCM will not infer any particular scheme for encoding rules and facts other than that constants and predicate arguments are represented by distinct, phased units. As mentioned in the introduction, the phase structure of the source KB and the target KB are independant of each other. We will say that the source KB has p_s phases and that the target KB has p_t phases.

It is assumed that the single entity participating in each BSet is composed of a single constant.

3.1 Binding Memory (BM)

The first step in the communication process involves extracting a temporal BSet out of the source KB, in response to an internally generated phased *extractor* signal from within the KB. There are several reasons why a KB might request an extraction. It may be that the KB has completed its episode of reasoning, or perhaps it has reached some intermediate conclusion that it wishes to pass along to the target KB as quickly as possible, with more definitive information to follow. It may also be the case that the source KB may desire to free up some of its phases, as new input, possibly requiring several individuals, may be waiting for entry. Thus, there is a sense of urgency that DBCM not only be able to perform an extraction quickly, but be able to perform multiple extractions quickly. In particular, quicker than the processing speed of the bottleneck of the system, the Phase Allocator, which is able to process only a single BSet at a time (see section 3.3).

The Binding Memory acts as the repository of each extracted BSet. In light of the fact that it would be deleterious to the source KB to operate down to the speed of the Phase Allocator, the BM, acting as a binding buffer, is able to support k banks of nodes, where each bank is able to spatially represent one BSet. The choice of k will most likely depend on the characteristic ratio (period width to phase width) of both the source KB as well as the target KB. Later we will see how the first searchlight mechanism enforces that exactly one bank is a candidate repository for each BSet being extracted. Still later, we will see how the second searchlight mechanism controls the flow of information from the BM such that the Phase Allocator only receives input from one bank at a time. This is necessary to avoid the cross-talk problem that would be the natural consequence of the PA processing several distinct BSets at one time.

As stated earlier, each bank represents a BSet via a spatial encoding. Thus, for every concept[1] in DBCM, there is a dedicated, non-phased τ-or latch unit (i.e. characteristically fires for an integer multiple of period length) within each bank that refers to that concept. Activation of said unit within a bank indicates that the bank is representing a binding in which that concept is involved. The physical mapping between the unit representing a concept in the source KB and its counterpart in each of the k banks in the BM is established by a link from the former to each of the latter. See figure 2.

There are several problems that have to be solved in order for the BM to operate as

[1]We will use *concept* to refer to any constant or predicate argument about which DBCM is concerned. The set of all such concepts that are part of DBCM will be a subset of those concepts (i.e. constants and predicate arguments) of the source KB and of the target KB.

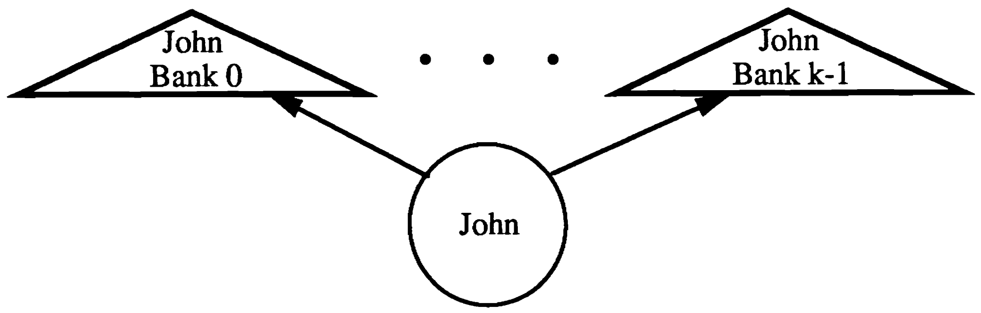

Figure 2: Connections between a sample concept in the source KB to the corresponding unit in each of the k banks of the Binding Memory. Each concept in the source KB is represented by a ρ-btu unit. Concepts in the BM are represented by τ-or latches.

planned. Let us first focus on how any given bank can come to represent a single BSet. As can be seen by looking at figure 2, there is nothing to prevent a bank from responding to all of the active concepts in the source KB. In addition to obviating the need for multiple banks, this would lead to the worst possible form of cross-talk, since we would have no way of distinguishing one BSet from another! What we need is a method by which a bank only responds to a single BSet at any given time. This effect is achieved by associating a "BSet grabber" unit with each bank in the BM and establishing a link from this unit to each concept unit in the bank. This new unit is a ρ-btu unit that "grabs" onto a phased extractor signal emanating from the source KB by firing in phased response. The weights on the links into each bank unit and the threshold of each bank unit are such that a bank unit must receive activation from both the grabber and from the source KB in order to fire in response. See figure 3. This ensures that a bank can only respond to input from one particular phase (chosen by its grabber) at any one time.

This architecture prevents a bank from succumbing to cross-talk, as a bank can only represent at most one BSet at any given time. However, as it stand now we still have the problem that all the banks will respond identically; in response to the first extractor signal, *all* the BSet grabbers will fire, and each bank will end up representing the same bindings. Not only is this redundancy unnecessary, it effectively reduces a multi-bank BM to a single bank. As discussed previously, it is important that DBCM is able to quickly extract multiple BSets from the source KB. Reducing a multi-bank BM to a single bank would eliminate this capability. What we need is a mechanism to guide the binding extractions so that only a single bank is a candidate repository for each temporal BSet being extracted. The effect is achieved by using a searchlight mechanism. The mechanism associates a "spotlight" unit with each bank and establishes a link from each spotlight to the corresponding BSet grabber. The strength of this connection, along with the strength of the connection from the extractor

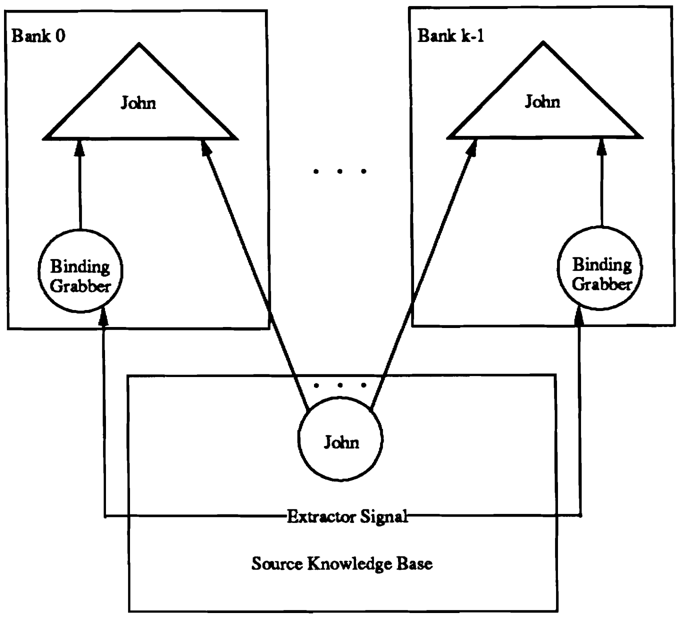

Figure 3: Binding grabbers make certain that only a single BSet is extracted into a bank at a time. A concept unit in a bank will respond only to simultaneous activity from the bank's binding grabber and the corresponding concept in source KB. Binding grabbers are activated by a phased extractor signal from the source KB.

signal into the grabber, along with the threshold of the grabber are such that the grabber will not fire unless it receives activation from an extractor signal as well as from its spotlight. Since the extractor signal will come in an unpredicatable phase, and will vary between extractions, it is necessary that the spotlight *not* be phased in nature, but rather, fire for full period length. Moreover, we cannot even predict the period in which the extractor signal will arrive. Thus, each spotlight should be implemented as a τ-or latch unit.

What remains for us now is to control the operation of the spotlights. Clearly, only the spotlight for one bank should fire at any given point in time, otherwise, the BM will be redundant. As pointed out previously, the source KB would like the BM to be able to extract bindings as quickly as possible. Hence, we would like to minimize the lag time between the turning off of one spotlight and the subsequent onset of another, since the BSet extraction process may not begin in the absence of spotlight activity. In searchlight terminology, we want a single beam searchlight that is able to move it's beam as quickly as possible. Let us first address the issue of turning on a dormant spotlight.

As soon as a grabber latches onto a BSet, we wish to turn on the spotlight in the next bank[2]. Thus, we add a link from the grabber for bank i to the spotlight for bank $i + 1$[3].

[2]Assume some arbitrary linear ordering of the banks.

[3]Let us number the banks $0, \ldots, k-1$, and for convenience, assume all addition is modulo k.

Activation along this link should be sufficient to turn on this next spotlight, thus enabling the next bank to be a candidate recipient for a BSet as soon as possible (2 periods following the previous extraction).

What remains is for us to describe how an active spotlight is turned off. The benefit of the activation of the spotlight ends as soon as the corresponding grabber fires, thereby latching onto a BSet. Thus, we add a strong inhibitory link from each grabber to the corresponding spotlight. When the grabber fires, activation along this link is sufficient to turn off the spotlight. This unit will not become active again until it receives an onset pulse from the grabber in the previous bank, as discussed above. See figure 4. Thus, candidate banks are

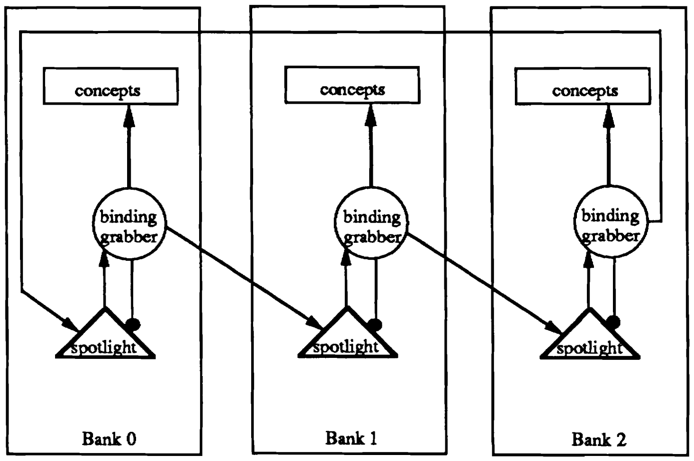

Figure 4: Here, $k = 3$. Spotlights fire in a mutual exclusive manner.

chosen in round-robin fashion to store BSets.

Up until now we have not elaborated on the extractor signals emanating from the source KB. This paper does not address how the extractor signals are generated, other than that they are internally-generated by the source KB. It is assumed that the signals are phased, and that a signal in phase i is a message to DBCM to extract the BSet encoded in phase i from the source KB. Consistent with the KB form, DBCM assumes that there are a set of ρ-btu-like extractor units within the source KB, and that there are links from each of these units to each grabber. See figure 5 . As such, any bank has the potential of representing the BSet indicated by the firing of any of the extractor units. The responsibility for turning off a firing extractor unit is that of the source KB, although perhaps it should do so in response to an appropriate phased pulse from DBCM that signals the source KB that the request is being processed. An extractor grabbed by bank i should be turned off before the onset of spotlight $i+1$. Failure to do this may result in redundancy within the BM, as more than one bank may come to represent the same BSet. Furthermore, this redundant extraction may

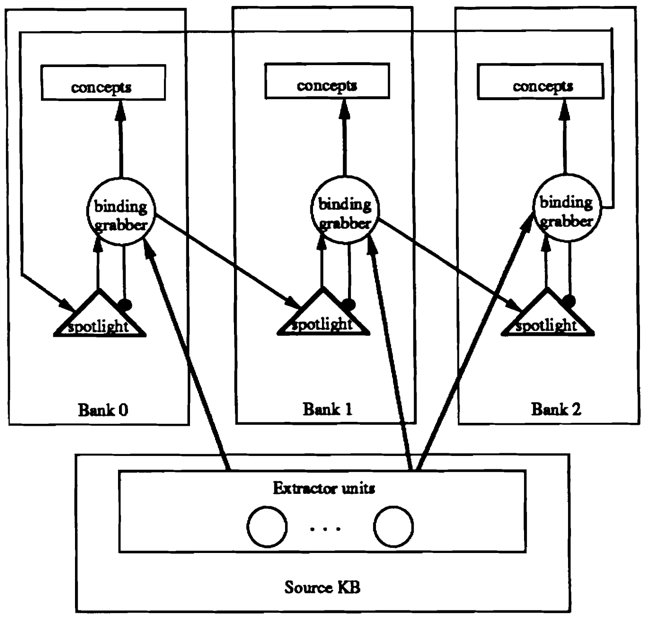

Figure 5: Here, $k = 3$. There is a connection from each ρ-btu-like unit in the source KB to the binding grabber in each bank of the BM.

be delaying the extraction process of another BSet that the source KB wishes to transfer.

In light of the above, it seems prudent to provide pairwise inhibition between the BSet grabbers in order to be more robust in the presence of such failures. The idea is for a grabber, once it starts to fire, to inhibit the other grabbers from responding in that same phase. Ideally, the blocking pulse should last until the triggering extractor unit has been turned off. Such a mechanism might even be required when dealing with a multi-beam searchlight that allows for multiple BSet extractions to occur simultaneously. It is likely that future versions of DBCM will incorporate such an inhibitory technique.

3.2 Phase Database

The function of the Phase Database is to monitor the target KB and find an unused, candidate phase in which to place the next incoming BSet. See figure 6 for a diagram of the PD. There are 2 sets of units that compose the PD. First, there are a set of p_t ρ-btu units that oscillates within each period, *according to the phase structure of the target KB*. The i^{th} unit fires during the i^{th} phase of each period. The behaviour of these units never varies. They are present merely to communicate the phase structure of the target KB to the second part of the PD, the "free-phase-selector" (FPS). The FPS is a single ρ-btu unit that has the responsibility of communicating to the Phase Allocator, by virtue of a phased pulse, an available phase for the incoming BSet.

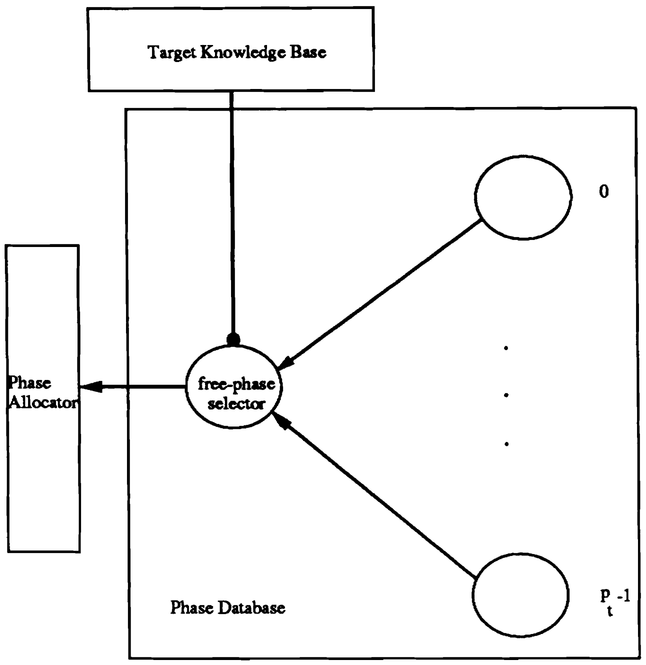

Figure 6: The Phase Database. The free-phase-selector transmits a currently unused phase within the target KB to the PA. The units on the right oscillate according to the phase structure of the target KB, one per each phase. The FPS receives inhibition from the concept units within the target KB.

In order for the FPS to perform its task, it must receive 2 types of information. Firstly, it must be cognizant of the phase structure of the target KB, so that it is aware of all of the potential phases in the target KB within which a BSet may be placed. Secondly, the unit must be cognizant as to which phases are currently in use within the target KB. The first effect is acheived by establishing an excitatory link to the FPS from each oscillating unit. The strength of each of these links is sufficient to incite the unit to fire. Thus, the FPS is capable of firing in concert with any phase in the target KB.

In order to understand how the second effect is achieved, it must be remembered that a constant participates in every binding. Hence, there is a target KB constant firing in each phase that is currently encoding a BSet. In order to inform the FPS that a phase is in use, we establish a strong inhibitory connection to the unit from each constant in the target KB[4].

Since each binding represented in the target KB includes a constant, the FPS will be strongly inhibited (and prevented from responding) during any target KB phase that is currently in use. During all other phases, however, the FPS only receives excitatory input from one of the oscillating units, and will respond with a phased pulse of its own according

[4]Such a huge fan-in to the FPS seems more plausible when we represent a concept not by a single unit, but by an ensemble of units. This has the added effect of rendering a phased system far less sensitive to noise [6].

to the firing behaviour of the ρ-btu unit. The FPS will continually fire in one phase until a new BSet is incorporated into the target KB. At this point, a constant that until now had been dormant must start firing (since each binding includes a constant), and the FPS will be strongly inhibited during that phase. At this point, the FPS will output another available phase. We say that the Phase Database resets itself when this occurs. When the target KB releases a BSet, the inhibition from the included constant will die, creating a new candidate phase for selection by the FPS.

3.3 Phase Allocator

The function of the Phase Allocator is to synthesize information from the Phase Database and from the target KB and decide in which phase to temporally encode the incoming BSet, currently spatially represented in one of the banks of the BM. It then must interpret these bindings into the phase selected and introduce them into the target KB.

The logic of the Phase Allocator is as follows. First, the second searchlight mechanism, discussed below, ensures that activity from at most one bank in the BM is currently affecting the PA. This eliminates the possibility of more than one BSet applying for entrance to the target KB simultaneously, a situation that would reak of cross-talk. When we subsequently refer to the PA receiving activation, we refer to this single, enabled bank. Since the PA must ultimately represent each BSet in a temporal code, we must provide the PA with a phased unit for each relevant concept, i.e. all concepts that are represented in a bank. As such, there is a ρ-btu unit in the PA for each constant and predicate argument that may participate in a binding. Again, since the ultimate objective involves transfering information from the BM into the target KB via temporally encoded bindings in the PA, there are connections to each of these PA units from each of the corresponding concepts in each of the k banks. Similarly, there is a link from each PA unit to the corresponding concept in the target KB. See figure 7.

While there is exactly one entity participating in each BSet, there could be an arbitrary number of predicate arguments participating in any one BSet. Thus, only 1 PA entity will receive activation during any transfer through the PA, while any number of PA predicate arguments may be activated. DBCM adopts an entity-driven approach to determine whether the member bindings of a BSet should be classified as either completely novel data with respect to the target KB (if the entity is inactive within the target KB), or as the (possible) augmentation of an existing DB already present in the target KB (if the entity is active within the target KB). If the included entity is already participating in a binding in the target KB,

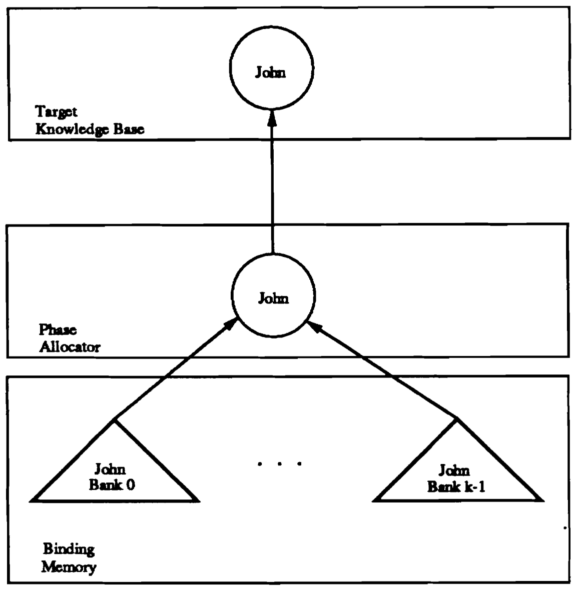

Figure 7: Pathway of a typical concept unit in DBCM.

the PA should augment this temporal BSet by incorporating the incoming bindings in the appropriate phase. If this entity is inactive in the target KB, the PA should introduce the bindings in the free phase selected by the FPS. It is the responsibility of the entity within the PA to decide between these 2 courses of action, and to communicate this decision (by means of a phased pulse) to the other members of the incoming BSet. These members will be some subset of the PA units representing predicate arguments (roles). In the current version of DBCM, each entity is composed of a single constant. Thus, the decision will be made at, and broadcast from, a single PA constant unit.

In order to be able to first make the decision, each constant unit needs to know the available phase selected by the FPS, as well as in which phase, if any, the constant is active in the target KB. Hence, we establish a connection from the FPS to each constant unit in the PA. Also, we establish a downward link from each constant in the target KB to the corresponding unit in the PA. In order to provide the PA constant a means of informing all the roles it fills of its decision, we must establish a connection within the PA from each constant to each predicate argument. See figure 8. For reasons that will become clear shortly, we need some more apparatus in order to have a properly designed PA. Associated with each constant unit in the PA is a τ-or unit, which receives a connection from both the PA constant and its corresponding unit in the target KB. There is a strong inhibitory signal from the τ-or unit that has the effect of disabling, or blocking, any activation being transmitted to the constant unit from the FPS. These τ-or units are incited by activation along any of its

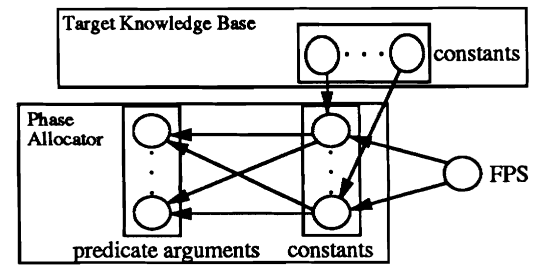

Figure 8: All constants in PA receive input from the FPS, and in turn, all constants feed all the predicate arguments within the PA. There are 1-1 downward connections from each constant in the target KB to the corresponding unit in the PA. Predicate argument units in the target KB are not shown.

incoming links. Finally, the connection strengths and unit thresholds are such that the PA ρ-btu units (i.e. the constant and predicate argument units) must receive activation from 2 sources in order to fire. Thus, a predicate argument unit must receive activation from the BM and from a constant unit in order to fire. A constant unit must receive activation from the BM and either from the PD or target KB in order to fire. See figure 9.

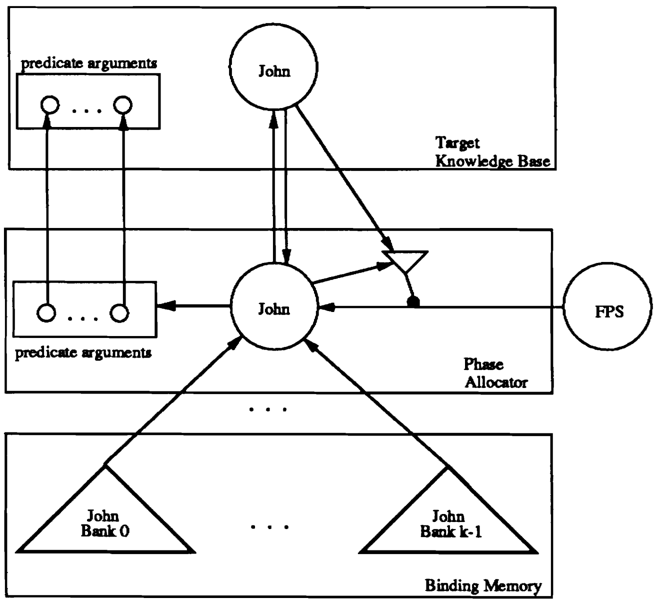

Figure 9: A constant unit in the PA must receive activation from the BM as well as from another source in order to fire. A predicate argument unit must receive activation from within the PA as well as from the BM (links from BM to PA predicate arguments not shown).

The PA operates as follows. Let's first consider the case where the constant in question is already participating in a BSet within the target KB. This constant then will have been activating the τ-or unit associated with the corresponding PA constant. This period-long pulse then has the effect of inhibiting activation from the FPS at that PA constant unit, regardless of the phase in which the activation occurs. At the same time, however, the constant in the target KB is sending a phased pulse through its downward link to the PA constant. Thus, this constant is receiving period-long activity from the BM and a phased pulse from the target KB. This activation is enough to trigger the ρ-btu constant, but *only during the phase in which it receives activation from the target KB.* The upward link from the PA constant to the target KB and the lateral link from the PA constant to the associated τ-or unit are not crucial in this case, since both units are already firing. However, the PA constant is connected to each PA predicate argument. This gives all of the PA predicate argument units that are participating in the incoming BSet (and thus are receiving input from the BM) enough impetus to fire in-phase with the constant. The corresponding predicate arguments in the target KB are then activated in-phase through the upward links from the PA. The Phase Database remains unchanged.

Now, consider the other alternative; the constant in question is not participating in any bindings within the target KB. In this case, the PA constant receives no activation from the target KB, and the unused phase selected by the FPS is transmitted, uninhibited, to the PA constant. Meanwhile, this constant is receiving period-long activity from the BM. Thus, the constant becomes active during this phase in the next period. The effect on the the rest of the PA and target KB is a superset of the other alternative. Here, in addition, the dormant constant in the target KB is activated in-phase through the upward link from the PA constant. This, in turn, has 2 important effects. One, the τ-or unit associated with the PA constant is activated, serving to inhibit any input to that constant from the FPS. This ensures that at a subsequent point in time, if another BSet containing this constant is being processed by the PA before the target KB has released this phase, that it will be processed as an instance of the previous case. Two, according to the logic of the Phase Database described in section 3.2, the FPS is turned off and is strongly inhibited from firing in this newly occupied phase . The FPS will then choose a new available phase that will be a candidate for the next DB processed. The link from the PA constant to the τ-or unit is neccessary to ensure that a PA constant does not receive activation from both the target KB and the FPS during the same period (we're actually concerned about the same *phase* during that period). During certain conditions, this event can activate the PA predicate arguments of the next incoming BSet prematurely and in the incorrect phase, leading to cross-talk in the target KB.

Note that in order to ensure the proper operation of the system, it is necessary that the spot-lit bank in the BM transmit its bindings for (at least) 2 consecutive periods. The first period enables the constant to make its decision, and the second allows the PA to correctly process the predicate arguments. As discussed earlier, the PA is the bottleneck of the system, and hence we wish to minimize the time spent by the PA in processing a BSet. In light of this, after these 2 periods, the second searchlight should disable the BM from transmitting the BSet. After a 1 period delay to allow the PD to reset itself (see section 3.2), the second searchlight should allow the next bank to transmit its BSet (if currently being represented). Thus the PA must be dedicated to a single transfer for 3 periods, which is the time it takes to move a set of bindings from a spot-lit bank into the target KB.

3.4 Controlling flow of bindings

In section 3.1 we discussed the benefits of a multi-bank BM, and described how a group of banks may all be active simultaneously, each representing a distinct BSet. The connectivity pictured in figure 7 indicates that activity in each bank is transmitted immediately into the Phase Allocator. Yet, allowing the PA to process more than one BSet simulateously will introduce cross-talk into the target KB! Clearly, we need a mechanism that will prohibit the BM from transfering data from more than one bank at a time to the PA, and at a rate no greater than the speed of the PA[5].

3.4.1 Searchlight Components

DBCM will utilize a searchlight mechanism to achieve this functionality. The implementation of this second searchlight requires 2 new units to be associated with each bank in the BM; Let us name the units "binding?" and "bank chooser".

Activity of a bank's binding? unit indicates that the bank is active. In effect, we wish this new unit to behave as if it were a member of whatever BSet the bank is currently representing. Thus, each binding? unit is τ-or latch with the same firing duration[6] as that of the banks' concept units. There is an intra-bank connection from the BSet grabber to this unit. See figure 10. The behaviour of this subnetwork is such that whenever the grabber fires, the binding? unit will begin firing during the following period. Looking back to section 3.1 we can see that the onset of this unit temporally coincides with the initial representation of an extracted BSet in a bank.

Activity of a bank's chooser unit indicates that it will be the next bank allowed to transmit its activity to the PA. Note that the activity of this unit does not imply that the

[5]Currently, the PA can process new BSets at the rate of 1 every 3 periods.

[6]This duration will be a function of the number of banks, the speed of the PA, and the speed and frequency of extracting a binding from the source KB. See section 4.

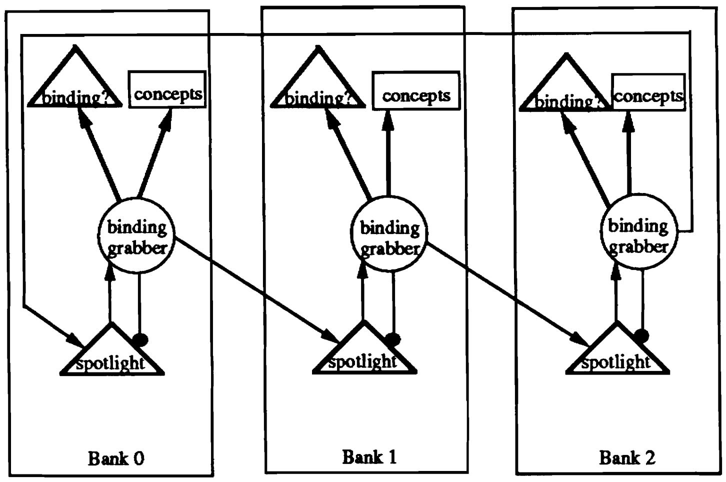

Figure 10: Here, $k = 3$. Activation from a binding grabber incites the corresponding binding? unit.

bank is active, only that the next set of bindings transferred into the PA may come from that bank. Since only one bank can activate the PA at a time without risking cross-talk, more than one of these units should not be active simultaneously. Remember that banks are chosen in a round-robin fashion to store BSets (section 3.1). In order to be certain that all extracted BSets get processed by the PA, the second searchlight mechanism gates active banks into the PA on a *first-active-first-gated basis*, mimicking the round-robin selection of the first searchlight. Thus, the chooser units much resemble the spotlight units in the first searchlight, firing in a mutual exclusive, round-robin manner, thereby selecting a single bank for processing.

3.4.2 Searchlight Control

The second searchlight should allow a bank to transmit its bindings to the PA only when it is at the head of the active bank queue. This occurs only upon the simultaneous activation of the binding? and chooser units within a single bank. The mechanism views this event as producing a **CABLE ACTIVATOR** (CA) signal for the bank, whose presence indicates that the cable of links from the single bank to the PA should immediately be activated, enabling the transfer of information along that cable to the PA. In the absence of such a signal, a cable is incapable of any transmission.

See figure 11 to see how the searchlight mechanism implements this control strategy locally at each concept unit in the PA. Each PA unit has k input sites (one for each cable) at which Binding Memory input is incident. The appropriate thread of the i^{th} cable will be incident at the i^{th} site of each destination unit. Also incident at the i^{th} site of each PA unit

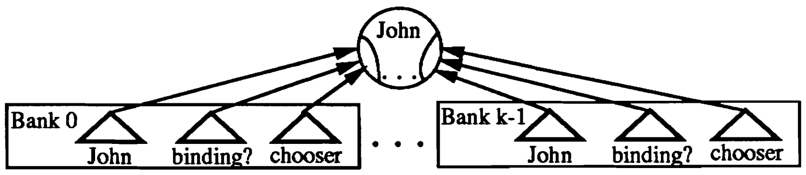

Figure 11: Second searchlight controls flow of activity from BM to PA. Each concept unit in the PA has k sites. The appropriate thread of cable i is incident at site i of the destination unit. Inter-bank connections are not shown. Simultaneous activation within a single bank of binding? and chooser units constitute a **CABLE ACTIVATOR** signal for that bank.

are connections from the binding? and chooser units associated with bank i. In order for one of these sites on a PA unit to signal activation, it must receive input from all 3 incident links. Thus, BM input is only capable of activating the PA in the presence of a **CA** signal. Since only one chooser can fire at a time, only the *same site* across the PA units may respond to BM input at any time.

The PA units as just described appear too complex. However, one may consider the simpler, yet functionally equivalent architecture where PA unit sites are replaced by new units that relay activation to the PA. This is illustrated in figure 12.

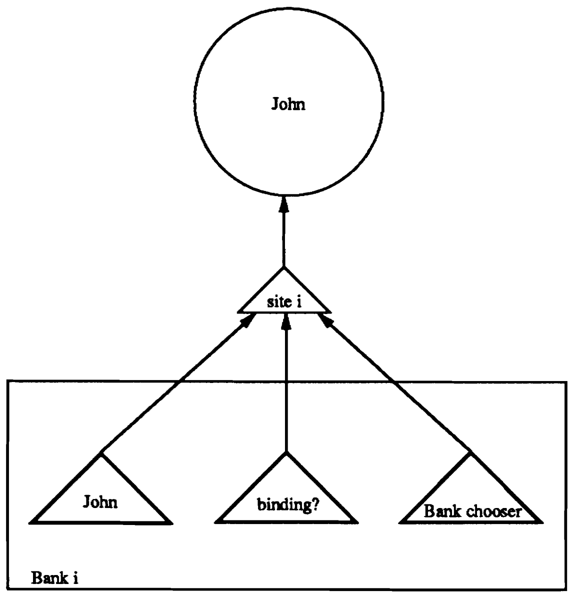

Figure 12: Each PA unit site in figure 11 is implemented as a seperate unit. Each of these k new units will have outgoing links incident at the corresponding PA unit. Only the units corresponding to a single bank are shown.

3.4.3 Searchlight Logic

We have shown how the searchlight mechanism eliminates the possibility of cross-talk. In this section we will see how the second searchlight mechanism not only operates according to the requirements specified in section 3.3, but does so optimally, as new BSets are allowed to enter to the PA as soon as permitted. Optimality is achieved by requiring that the **CA** signal be present for exactly 2 periods, and that there be an enforced delay of 1 period before the next bank is permitted to issue a **CA** signal.

The searchlight is able to dictate the behaviour of the **CA** signals by controlling the offset and onset of the chooser units for each bank. Turning off a **CA** signal is achieved by turning off the contributing bank chooser unit. The 1 period of cable silence is achieved by waiting 1 period following the offset of the preceeding **CA** unit before turning on the chooser of the next bank. In order to help us control the movement of the searchlight's beam, we associate a new, τ-or, "beam mover" unit with each bank.

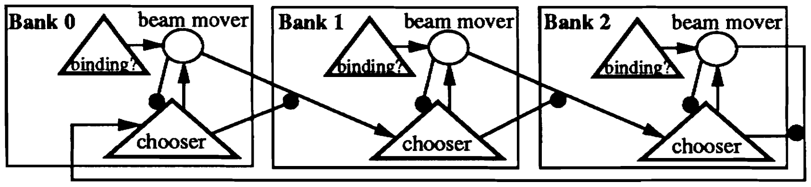

Figure 13: Here, $k = 3$. Logic of second searchlight enforces mutual exclusive activity among the chooser units.

Figure 13 depicts the logic of the second searchlight for k=3. There are intra-bank, excitatory connections to the beam mover unit from the binding? and chooser units. Activation along both these links is necessary to cause the beam mover unit to fire. Hence, this unit fires in response to the bank's **CA** signal. There is a strong inhibitory intra-bank connection from the beam mover to the bank chooser that stops the latter unit from firing, thereby removing the **CABLE ACTIVATOR** signal 2 periods following its inception. Note that the beam mover continues to fire for one period after offset of the chooser.

There is an inter-bank excitatory connection from the i^{th} beam mover to the $(i+1)^{th}$ chooser. Activation along this link provides sufficient impetus to incite the chooser to fire in the next period. However, a signal along this link is blocked by activation from the i^{th} chooser. Thus, the beam mover does not succeed in inciting the $(i+1)^{th}$ chooser until the i^{th} chooser is silent. The result is a one period lag between activation of succesive bank chooser units. During this lag, all the cables to the PA are disabled, allowing the Phase Database to reset itself so that the next BSet will be transferred properly. If the $(i+1)^{th}$ bank is active, then the bank will generate a **CA** signal immediately following this lag, enabling the bank to begin its 2 period broadcast to the PA as soon as is permitted.

4 Timing Considerations

Let us explicitly define the following timing constants that have arisen during the description of DBCM.

e the number of periods from a grabbed extractor pulse until the grabbed BSet is spatially represented in a bank (an extractor is grabbed in period i when a BSet grabber fires in response to that extractor in period i+1).

g minimum number of periods between succesive grabbed extractor signals.

p number of periods the Phase Allocator is dedicated to one BSet (including time needed to reset the Phase Database for the incoming BSet).

We have already seen that the values for these constants are 2, 2, and 3, respectively. Note that if we only concern ourselves with actual processing time, the number of periods needed to transfer a single BSet from the source KB to the target KB is given by $\mathbf{e} + \mathbf{p} = 5$. It is not always possible to acheive this lower bound for each BSet transferred, as we have seen how either searchlight might delay the processing of one BSet while concentrating on another (sections 3.1 and 3.4). However, the independence of the 2 searchlight mechanisms enables DBCM to overlap the processing of different BSets, more than compensating for these "local" delays. DBCM instantiated with k banks can completely process up to k transfers within any consecutive $((k * \mathbf{p}) + \mathbf{e})$ periods. Parameters that likely play a role in the optimal choice of k include p_s and p_t. The type of data being processed by both KBs, as well as the functional role that each plays in the reflexive reasoning model also will affect this value.

The activity of the τ-or latch units in each bank of the Binding Memory should endure until the second searchlight disables the outgoing cable to the Phase Allocator. A duration of $[(k - 1) * (\mathbf{p} - \mathbf{g})] + (\mathbf{p} - 1) = \mathbf{k} + 1$ periods satisfies this constraint. The first term reflects the maximal delay until onset of an active bank's **CABLE ACTIVATOR** signal. The second term reflects the duration of the **CA** signal.

5 Extensions

An extension to DBCM that enables communication between KBs that include an *isa*-heirarchy of concepts, as described in [9], is underway. A DB in such a system would typically include more than one constant, but all such constants will refer to a single entity.

While this will require a different implementation of the Phase Allocator, the logic of the PA will be unaffected, as we will still be keying on a single entity while deciding how to incorporate each BSet into the target KB. We are also looking into extending DBCM to allow a single target KB to receive bindings from multiple source KBs.

6 Discussion

We have described a communication mechanism between 2 distinct phased-based knowledge-bases that dynamically encode variable bindings by means of temporal synchrony. The quantum of data that is transmitted is a set of dynamic variable bindings, all involving the same entity. This BSet, temporally encoded in the source KB, is extracted and represented as a spatial encoding in the Binding Memory. Multiple BSets can be stored in the Binding Memory at one time. The ability to perform multiple extractions while avoiding cross-talk is the result of utilizing a searchlight mechanism to control the flow of information out of the source KB. The Phase Allocator transfers a spatially encoded BSet from the Binding Memory into the target KB, where variable bindings are again encoded via temporal synchrony. So as not to introduce cross-talk into the target KB, the Phase Allocator relies on feedback from the target KB as well as input from the Phase Database, which monitors the current allocation of phases in the target KB. Another potential source of cross-talk is prevented by a second searchlight mechanism, which ensures that only a single BSet is incorporated into the target KB at a time by controlling Binding Memory access to the PA.

As formulated, the Phase Allocator is the only means by which new bindings may be introduced into the target KB. We hope to show in future work how raw, sensory input might be introduced into a KB in much the same way that inter-KB data is processed (via a technique akin to that used by the Phase Allocator). We also hope to investigate the utility of DBCM by having the source KB act as a parser.

Lastly, we are currently developing a communication mechanism in which the unit of transfer is not a *phase* of activity, but rather is a set of facts (dynamically instantiated predicates). Typically, this would necessitate the transfer of bindings from multiple phases, involving multiple entities. Such a system could naturally operate between KBs that allow multiple instantiations of predicates [7]. Preliminary results indicate that arbitrary sets of facts can be transferred *between* distinct knowledge bases in this fashion on the same time scale needed to perform a single rule application *within* a single KB.

The Dynamic Binding Communication Mechanism is a computationally efficient, biologically plausible connectionist interface mechanism. We believe this system to be an important component in a general model of reflexive, i.e., rapid common sense reasoning.

References

[1] Crick, F. (1984). *Function of the thalamic reticular complex: The searchlight hypothesis*, PNAS, Vol. 81, pp. 4586-4590.

[2] Eckhorn, R., Bauer, R., Jordan, W., Brosch, M, Kruse, W. Munk, M., and Reitboeck, H.J. (1988). *Coherent Oscillations: A mechanism of feature linking in the visual cortex.* Biological Cybernetics, vol.60. 121-130.

[3] Eckhorn, R., Reitboeck, H.J., Arndt, M., and Dicke, P. (1989). *A Neural Network for feature linking via syncronous activity: Results from cat visual cortex and from simulations.* In Model of Brain Function, Cotterill RMJ (ed), Cambridge Univ. Press.

[4] Freeman, W.J. (1981). *A physiological basis of perception.* In Perspectives in Biology and Medicine. Univ. of Chicago Press.

[5] Gray, C.M. and Singer, W. (1989). *Stimulus-specific neuronal oscillations in orientation specific columns of cat visual cortex.* PNAS, Vol. 86, pp. 1698-1702.

[6] Mandelbaum, R. and Shastri, L. *A Robust Model for Temporal Synchronization of Distant Neurons.* Working Paper, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA. May, 1990.

[7] Mani, D.R. and Shastri, L. *Representing multiple dynamic instantiations of a predicate in a connectionist system.* Technical Report. Dept. of Computer Science, Univ. of Penssylvania. (To appear) May 1991.

[8] Shastri, L. (1990). *Connectionism and the Computational Effectiveness of Reasoning.* Theoretical Linguistics, Vol. 16, No. 1, 65-87, 1990.

[9] Shastri L, and Ajjanagadde, V. (1990). *From Simple Associations to Systematic Reasoning: A Connectionist Representation of Rules, Variables and Dynamic Bindings.* Technical Report MS-CIS-90-05, Dept. of Computer Science, Univ. of Pennsylvania.

[10] Treisman, A., and Gelade, G. (1980). *A feature integration theory of attention.* Cognitive Psychology, Vol. 12, 97-136.