From neural dynamics to true combinatorial structures.

Frank van der Velde[a]
Marc de Kamps[b]
[a]Cognitive Psychology, Leiden University
Wassenaarseweg 52, 2333 AK Leiden
The Netherlands
vdvelde@fsw.leidenuniv.nl

[b]Robotics and Embedded Systems
Department of Informatics, Technische Universität München
Boltzmannstr. 3, D-85748 Garching bei München
Germany
kamps@in.tum.de

Abstract: Various issues concerning the neural blackboard architectures for combinatorial structures are discussed and clarified. They range from issues related with neural dynamics, the structure of the architectures for language and vision, alternative architectures, to linguistic issues concerning the language architecture. Particular attention is given to the nature of true combinatorial structures and the way in which information can be retrieved from them in a productive and systematic manner.

To begin, we would like to express our appreciation for the work done by the commentators. They have presented us with a wide array of comments on the target article, ranging from dynamics and neural structure to intricate lexical issues. The topic of the target article was to solve the four problems described by Jackendoff (2002), and to illustrate that the solutions offered have the potential for further development. The problems described by Jackendoff concern the issue of the neural instantiation of combinatorial structures. Although it is true that language provides the most complex and hierarchical examples of combinatorial structures, combinatorial structures are also found in other domains of cognition, such as vision. Therefore, we discussed sentence structure and visual binding as examples of combinatorial structures. We argued and illustrated that these structures can be instantiated in neural terms by means of neural blackboard architectures. We aimed to discuss as many topics as possible within the framework of a target article. As a result, we had to glance over many details and related issues. The commentaries received offer us a possibility to discus some of these in more detail, and to rectify some of the misunderstandings about the nature of the architectures that might have resulted from this approach.

R1. Goal

The goal of our article as outlined above was understood very well by the commentators, with the exception of **Bod et al.**, who complain that we have not solved “the binding problem”. From the remainder of their commentary we get the impression that with “the binding problem” they refer to the issue of how all the potential representations can be derived form a single sentence input. In the case of a multidisciplinary discussion it is always important to clarify terms, because they could have a different meaning in different fields. For this reason, we explained the goal of our article by a rather detailed discussion of the problems addressed by Jackendoff. The binding problem in this respect refers to the notion of binding in neurocomputational terms. Jackendoff introduces it as follows (2002, p. 59): “The need for combining independent bits into a single coherent percept has been recognized in the theory of vision under the name of the *binding problem* (not to be confused with linguists' Binding Theory …). In the discussions I have encountered, the binding problem is usually stated in this way: we have found that the shape and the color of an object are encoded in different regions of the brain, an they can be differentially impaired by brain damage. How is it, then, that we sense a particular shape and color as attributes of the same object?". Jackendoff goes on to explain that this binding problem also occurs in language, and even more massively than in vision. So, the issue is how information, encoded in different regions in the brain, can be combined (bound) in a coherent representation, both in language and in vision. We argue that this can be done with blackboard architectures. The relation with vision is clear from the fact that we have also discussed a blackboard architecture for vision. It is difficult to see how this topic can be confused with the issue of finding all the syntactic structures of a given sentence (which is basically a virtual problem, see below)

Bod et al. also complain that we have concentrated on the issue of representing combinatorial structures in neural terms. This remark reflects a fundamental lack of understanding on the difference between symbolic computation and neural computation. Briefly, what is easy in symbolic computation is difficult in neural computation and vice versa. Representation of symbolic structures is easy in symbolic computation. Basically, anything goes. Hence, there are no restrictions on the kind of structures and processes one can have in symbolic computation. This is one of the reasons why we have so many different theories in theoretical linguistics today. (For example, **Bod et al.** urge us to take more note of Chomsky, 1957. But why this Chomsky? Why not the Chomsky of Government and Binding or the Chomsky of the Minimalist Program?)

The importance of the issue of representation can be seen very clearly in the need to copy symbols in the symbolic approach. Newell et al. (1989, p. 105) defended copying symbols as follows: "The need for symbols arises because it is not possible for all of the structure involved in a computation to be assembled ahead of time at the physical site of the computation. Thus it is necessary to travel out to other (distal) parts of the memory to obtain the additional structure". Hence, it is necessary to copy the information over there and to bring it over here. But nodes in a connectionist network seem to be fixed in their web of connections. This was a main argument for Fodor and Pylyshyn (1988) to dismiss connectionism as a basis for human cognition. In the words of Dennett (1991, p. 270): "This problem of fixed versus movable semantic elements is one way of looking at a fundamental unsolved problem of cognitive science." In our target article, we have provided a solution for this problem. Of course, a solution for this problem does not directly result in a complete theory of language and the brain. But, with a solution for this problem "the dialogue between linguistic theory and neural network modeling will begin to be more productive" (Jackendoff 2002, p. 65).

Bod et al. dismiss the architecture we present because it uses pre-wired connections. This remark again reflects a fundamental lack of understanding on neural computation. For example, to establish a relation between two words, there has to be a connection path between the neural structures that represent these words in the brain. Dramatic evidence comes from split brain patients, in which both hemispheres operate separately because the connection between the two is missing. For example, when the word *hammer* is presented in the left visual field, a split brain patient will perceive the word with the right hemisphere. So, the patient can use the left arm to point at an object representing a hammer. But he cannot name the word, because verbalization is (usually) produced by the left hemisphere, and this hemisphere is oblivious of the word perceived by the right hemisphere. Similarly, the left hemisphere can name the word *nail* when it is presented in the right visual field, but the right hemisphere is unaware of this word. So, when both *hammer* and *nail* are presented together in this way, the split brain patient is unaware of the relation between these two words, precisely because a connection path between their neural instantiations in both hemispheres is missing. Therefore, unless you want to assume that connections can grow on the fly, a pre-wired connection path has to be available to relate any two words with each other. The architecture we present provides a very parsimonious solution for this problem (see R3).

Bod et al. argue that we should have concentrated on the issue of finding the appropriate neural representations of all the syntactic structures that can be assigned to a given sentence. This remark illustrates that they are out of touch with the literature on human parsing. The human parser does not operate like a theoretical linguist. That is, the human parser does not wait until the sentence has been presented and then tries to make as many representations as possible. Instead, the parsing process starts straight away, and operates in an incremental manner, almost word by word. The circuit we illustrate in fig. 20 operates in such an incremental manner. As a result of this process, humans are quite often oblivious of the ambiguities that can be found in a sentence (e.g., Pinker 1994). Evidence comes from garden path sentences (e.g., Meng and Bader 2000), such as *The man who hunts ducks out on weekends* (Pinker 1994). Typically, humans interpret *duck* as a noun instead of a verb, which results in a misrepresentation. Furthermore, recent research is beginning to show that human sentence comprehension can produce underspecified representations of a given sentence (Sanford and Sturt 2002), which is quite the opposite from analyzing all possible interpretations of a sentence.

Bod et al. also argue that connectionism should be about learning. This idea is wrong for several reasons. First, the notion of connectionism derives from Cajal, who described the basic structure of the brain in terms of neurons as the functional units, which are related with connections. Nothing in this description associates connectionism exclusively with learning. Second, even in the PDP tradition one can find influential models which are not about learning. An example is the interactive activation model of letter and word recognition by McClelland and Rumelhart (1981). Third, it is a hazardous idea. There is no guarantee that connectionism, based on an initially unstructured architecture and learning principles, will eventually result in a cognitive system comparable to that of humans (or even primates). By far most species that developed a neural system in the course of evolution did not develop a neural system on the level of that of humans or primates. The lesson from this is that simple architectures with simple adaptive processes do not in general result in human-like cognition. In this respect, we fully accept the possibility that an architecture like ours will not develop on the basis of simple learning principles and an initially unstructured architecture. But that does not mean that it is incorrect. Fourth, it does not agree with brain development. A good example is given by the orientation maps in the primary visual cortex. The basis structure of these maps is available at birth, including individual cells that respond to specific line orientations (e.g., Chapman et al. 1996). So, here one can already find an example that contradicts the notion that connectionism should be only about learning.

Other remarks of **Bod et al**. can be easily dismissed as well. For example, they argue that the architecture cannot not detect the similarity between *The man frightens the child* and *The child fears the man*. It is amazing to read this statement, because we discussed this issue in section 6.6, where we discussed how the architecture can relate *John gives Mary a book* with *Mary owns a book*. We did this in terms of the long-term memory structures of these sentences, but the same mechanism works for structures in the blackboard, because the relation between both sentences depends on the associations between the word assemblies for *give* and *own*. These assemblies are the same for structures in the blackboard and for structures in long-term memory. We discussed this issue in terms of a single example, because we wanted to discuss as many different examples as possible, trusting the reader to understand that if the architecture can instantiate the relation between *gives (John, Mary, book)* and *own(Mary, book)*, the architecture can also instantiate the relation between *frighten(man, child)* and *fear(child, man)* or any other example of this kind. Similar remarks can be made for issues like agreement and long-distance binding. Examples of these are found in the article as well.

R2. Related work

Baars argued that one can find many other examples of blackboard architectures in cognition. We agree with that observation. For example, Jackendoff (2002) describes symbolic blackboard architectures for phonological structures, sentence structures and semantic structures. Each of these blackboards has its own specific structure, dependent on the nature of the combinatorial structures they are involved in. In fact, we took the importance of blackboard architectures for cognition as granted, and concentrated on the issue of their neural instantiation.

We took language as our prime target because of the obvious combinatorial structure of language. But we decided to include vision as well, because combinatorial structures are also found in vision, and because, at face value, they are different from language. In this way, we could raise the issue of whether neural blackboard architectures can be described for these different cognitive domains. If so, what makes them different, what makes them similar, and how can they be related (e.g., provide grounding of language in the visual domain)? The differences between the neural architectures we discussed follow from the differences between their cognitive domains (simply stated, sequential for language and spatial for vision). Yet, there are also similarities. In particular, they both support the process view of binding, that is, binding within the frame of reference of the system itself. Furthermore, they are related in terms of the grounding of language in perception, which could be the basis for a further development of semantics.

Similar questions can be discussed in case of the blackboard architectures mentioned by **Baars**. A neural instantiation of these architectures would have to be developed, with an emphasis, in our view, on the ability to answer "binding questions" in these architectures without relying on conjunctive forms of representation. These binding questions can be explicitly stated, but they could also be implicit. In fact, it would be interesting to explore if conscious cognition and voluntary control are related with the process of answering self-generated binding questions. For example, if it is the case that we are conscious of a color of an object when (or because?) we (implicitly) answer the question: "What is the color of this object?".

Clancey describes an interesting relation between our blackboard architecture for language and his Conceptual Coordination (CC). The relation between our blackboard architecture and architectures such as CC indeed indicate that the integration between neural and symbolic analyses of cognition is progressing.

As **Clancey** notes, comprehension is a sequential process and not some kind of state that is hold or “moved” in memory. This is what we have tried to illustrate in terms of the process of answering binding questions, both in the sentence blackboard and in the visual blackboard. It contrasts with the notion, as expressed by **Doumas et al.**, that “downstream” neurons should have an explicit representation of the binding structure instantiated with “upstream” neurons.

Clancey discusses some refinements, related with binding problems involved in sentence comprehension, that could be introduced in the architecture. The examples given are interesting and need to be investigated in more detail. For the moment, we note that some of these refinements are already implemented in the architecture. They concern the structure assemblies that are simultaneously active in memory. Their number is indeed related to memory span. **Clancey** suggest that the multiple activations that constitute the memory span are not given by the structure assemblies themselves, but by the categorizations given by the different delay assemblies that connect the structure assemblies. Indeed, this is what happens in the blackboard. The activation of the structure assemblies guides the binding process, but once binding has been achieved, the memory is given by the active delay assemblies. The idea that only one delay assembly can be active at the time at a specific binding site is reflected in the connection structure illustrated in fig. 5. The competition within the “rows” of columns and within the “columns” of columns also ensures that a delay assembly can only be active in one way at a time.

R3. Structure of the architecture

Hadley argued that we have presented a conjunctive form of binding, which would require in the order of 60 million distinct binding circuits to succeed. If this was true, the architecture we presented would hardly be worth a discussion. But **Hadley**’s point, and his calculation, are based on a misinterpretation of our architecture. It turns out that **Hadley** confuses words (e.g., nouns and verbs) with structure assemblies (e.g., NP and VP assemblies). With 5000 nouns and 3000 verbs, **Hadley** calculates that the “agent matrix” (fig. 5) would consist of 15 million memory circuits. However, the agent matrix does not represent binding between nouns and verbs, but between NP assemblies and VP assemblies. The binding of nouns and verbs is indirect in the architecture. First, nouns bind with NP assemblies and verbs bind with VP assemblies. Then, the NP and VP assemblies bind with each other. This intermediary layer of binding makes all the difference. This is what makes the architecture a blackboard architecture. NP and VP assemblies are only needed for the temporal and online representation and processing of a sentence structure.

We noted in the target article that in the order of 100 NP assemblies and 100 VP assemblies would suffice for this purpose. In that case, one could have in the order of 100 nouns and 100 verbs involved in online representation and processing, which is more than sufficient to account for working memory capacity in language comprehension (e.g., Just and Carpenter 1992), that is, before transfer to the HC. With 100 NP and 100 VP assemblies, the agent matrix consists of 10,000 memory circuits, a far cry from the 15 million memory circuits calculated by **Hadley**. But, of course, we also have to count the bindings between words and structure assemblies. Using **Hadley**’s example, we would have 500,000 memory circuits to bind the nouns with the NP assemblies and 300,000 memory circuits to bind the verbs with the VP assemblies, that is, in the order of 10^6 bindings between words and structure assemblies. Again, this is significantly less than the number calculated by **Hadley**, but it is a substantial number. **Choe**, **Durstewitz** and **Grüning & Treves** also raised doubts about this number, and about the fact that novel words could hardly bind with structure assemblies in this way.

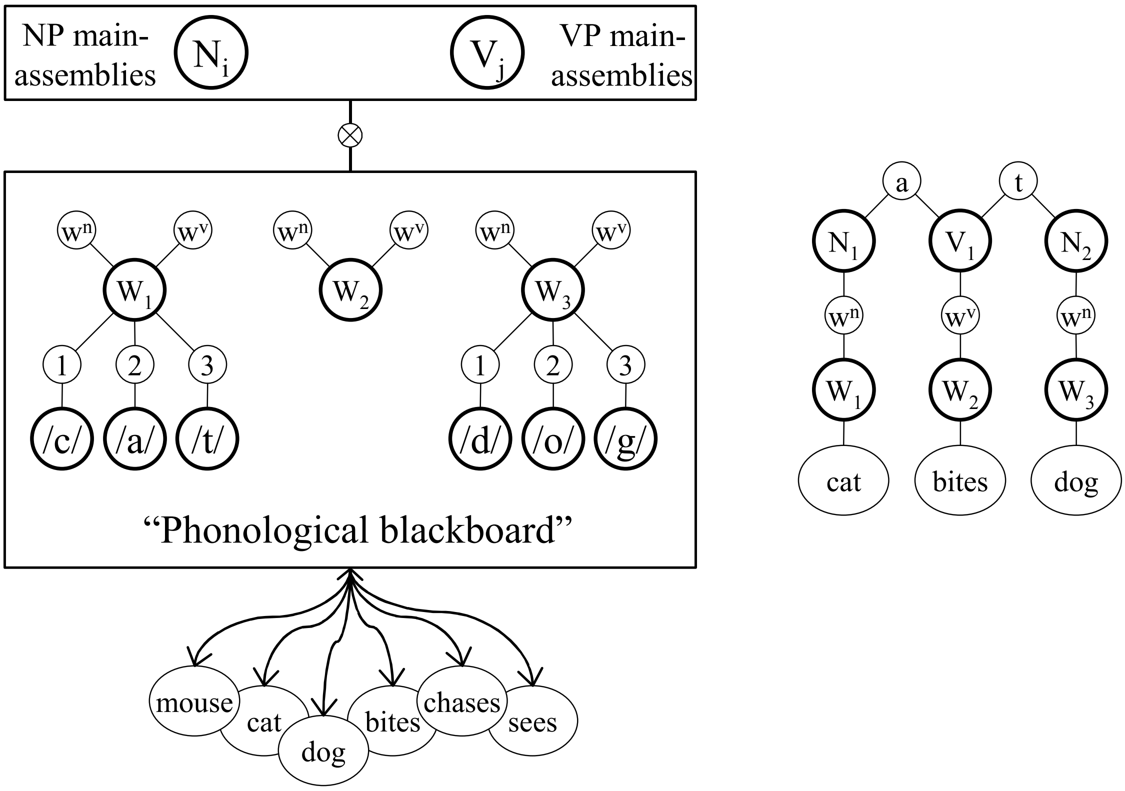

Figure R1. Illustration of how a "phonological blackboard" would bind words to structure assemblies in the sentence blackboard.

In section 6.9 we discussed that, in a future development, the sentence blackboard architecture should be combined with a phonological blackboard architecture (as in Jackendoff 2002). We will illustrate how such a phonological blackboard architecture will significantly reduce the number of bindings between words and structure assemblies, and how it can account for the binding of novel words. Figure R1 illustrates a quasi "phonological blackboard" that connects with the sentence blackboard (e.g., to the NP and VP assemblies). The structures in the "phonological blackboard" are not intended as genuine phonological structures, but they do illustrate the point that a word is also a combinatorial structure based on phonological constituents. These constituents will bind with W_x assemblies similar to the way in which words (structure assemblies) bind with the S_i assemblies in the sentence blackboard. The binding circuits will be the same as in the sentence blackboard. The word assemblies, representing the long-term information related to words, are located outside of the phonological blackboard, but they interact with it in an interactive way. The word assemblies can be as distributed as they have to be, but each word assembly will be connected to specific phonological constituents in the phonological blackboard. So, the word assembly for *cat* is connected to the quasi phonological constituents /c/, /a/ and /t/ of the word *cat*.

So, when a word is presented, it will activate its word assembly and its phonological constituents in the phonological blackboard. The phonological constituents in turn will bind with an arbitrary (but "free") W_x assembly. Like the structure assemblies, the W_x assemblies are involved only in online representation and processing. So, one can assume that the number of W_x assemblies will be in the order of 100 as well. Because the W_x assemblies are arbitrary, one needs a connection matrix to bind W_x assemblies with structure (e.g., NP and VP) assemblies. In turn, this means that the NP and VP assemblies have to have subassemblies for this binding, as illustrated in figure R1 (which makes all bindings of structure assemblies of the same kind, i.e., via subassemblies). The W_x assemblies have to have subassemblies as well. We assume that they have specific subassemblies that allow selective binding with NP assemblies (with w^n subassemblies) and VP assemblies (with w^v subassemblies). For each of these, one needs a connection matrix in the order of 10,000 memory circuits (like the agent matrix in fig. 5). So, the 10^6 bindings between words and structure assemblies reduce to (in the order of) 10^4 bindings between the phonological blackboard and the sentence blackboard. Hence, all bindings in the architecture (i.e., between subassemblies) are in the order of 10^4. The number of different subassemblies, reflecting different categories (e.g., "agent", "theme", "clause", "preposition") will be between 10 and 100. Therefore, the overall number of bindings (memory circuits) in the architecture will be between 10^5 and 10^6. This number is a lot more feasible than the 60 million calculated by **Hadley**.

The use of a phonological blackboard also explains how novel words can bind with the sentence structure. A novel word is a new combination of familiar phonological constituents, which can bind with a W_x assembly. The phonological nature of the new word could be used to decide if the word is a noun or verb, so that it can bind directly in the available sentence structure.

A similar confusion between words (e.g., nouns, verbs) and structure assemblies (e.g., NP and VP) seems to be the basis of **Dyer**'s comment. The connection matrix in fig. 5 refers to binding between the NP and VP structure assemblies, not to the binding between words. As we argued above (R1), some kind of connection path between any pair of words (i.e., their neural instantiation) is necessary to account for the fact that these words can be related to one another. As we calculated here (R3), the binding structure in the blackboard is a very parsimonious way to provide for all possible connection paths that language would require.

The connection matrix between structure assemblies also accounts for the binding process itself. Binding occurs between active structure assemblies (i.e., their corresponding subassemblies). Usually, only one NP subassembly and one VP subassembly are simultaneously active. So, binding in the connection matrix will be immediate, and no winner-take all process is needed. In fact, the conflicts in binding that do sometimes occur are a major source of complexity in this architecture, as illustrated in fig. 18 and fig. R2 (section R8.1).

The connection matrix also accounts for the way in which the binding process is initiated. When sentence information indicates that a noun should bind with a verb as its "agent", the gating circuits between the main assemblies and the subassemblies are opened. Indeed, for all NP and VP assemblies (but not for all nouns and verbs). But only one NP (main) assembly and only one VP (main) assembly is active at a given moment. So, only one subassembly for NP assemblies and only one subassembly for VP assemblies will be activated, which will result in the binding process described in the previous paragraph.

Dyer raises an interesting point about whether the architecture would scale up for semantics. We used the labels "agent" and "theme" to describe the binding structure in the architecture.

In passing, we noted that we took these labels in terms of semantic macroroles (Van Valin 2001, see note 6). We could also have used the notation of Pinker (1989), who refers to the argument slots of verbs in terms of *X*, *Y* and *Z*. So, *X chases Y*, *X bites Y*, and *X gives Y to Z*. A substantial number of different semantic roles can be distinguished, but nevertheless verbs can have one, two or three arguments, and these argument positions seem to have something in common. The connection matrices we described would be used for these argument positions, i.e., for the *X*, *Y* and *Z* arguments of verbs. The more precise classification of the semantic roles will depend on the (lexical) information associated with a particular verb. There is no need to instantiate all these different semantic roles directly within the architecture. Notice that there is only one instance of any given predicate in the architecture (e.g., *give* or *own*), that is, there is only one word assembly for *give* or *own*. Long-term associations can be formed between these predicates, that is, between their neural assemblies. In the same way, information can be associated with a word assembly of a verb to express the fact that, say, the *X* argument of that verb has a particular semantic role (e.g, agent, experiencer, cognizer, believer, etc.).

R3.1. Holistic representation. In their commentary, **Dresp & Barthaud** argue for holistic representations instead of combinatorial ones that can be implemented in a blackboard architecture. In particular, because this would fit more with the need for generating behavioral success. We agree that if a problem can be handled with conjunctive or holistic forms of representation it pays of to do so. Processes that use these forms of representation typically operate very fast, which increases the chance of survival. However, the success of this approach depends on the size of the problem space. When the problem space increases, the success of this approach declines rapidly.

The argument of behavioral adaptation is moot here, because human language did evolve, and it provides the ability to go beyond the here and now that is typical for behavioral adaptation. The argument that one does no find these processes in animals is of little value, considering that one does not find language in animals either. The same applies to those aspects of human cognition that have produced the environment that most of us live in, which is substantially different from the environments that animals live in. Just think of all the aspects of cognition that are involved in creating and sustaining a research institute or university, or in organizing a discussion like this one. One cannot explain that away by placing human cognition on the Procrustean bed of animal cognition and immediate behavioral success. Something has to be different to account for the uniqueness of human cognition.

This being said, we do in fact believe that, in particular, the blackboard architecture for vision is related to behavioral success, and that the relation between the language blackboard and the vision blackboard we proposed (the "grounding" of language) is sound. **Dresp & Barthaud** argue that the ART theory of Grossberg (1999) is an alternative theory which fits with their holistic view of cognition. Binding between "what" and "where" would correspond with resonant activation states in ART, formed by the activation patterns in the "what" and "where" processing steams. So, binding could be achieve without any language-like combinatorial process.

However, the binding process in the visual architecture is combinatorial but not language-like. The reason why we discussed the blackboards for language and vision is to show that one can have different blackboard architectures that are tailor-made for the modality in which they operate. Contrary to the claim of **Dresp & Barthaud**, they are not at all general purpose machines. For example, the purpose of the discussion on language complexity was to show that the language architecture is selective to the kind of sentence structures it can encode, contrary what one finds in a general purpose machine.

Furthermore, the idea that resonant activation states in ART bind activation patterns in the "what" and "where" processing steams is based on a complete misinterpretation of Grossberg (1999). Grossberg argued that ART concerns the process of learning and categorization in the "what" (ventral) stream, and that an entirely different (in fact, complementary) model accounts for learning and processing in the "where" (dorsal) stream. The difference between these two complementary models is a crucial point in Grossberg (1999). In short, ART is related to conscious memories, and the "dorsal" model is related to procedural (unconscious) memory.

ART is about learning and categorization in the "what" stream, whereas our vision architecture is related to the interaction between the "what" and "where" stream. There are a number of animal experiments that are in agreement with the interaction in the vision blackboard illustrated in figs 22 to 24. A detailed description can be found in Van der Velde and de Kamps (2001). Furthermore, the blackboard architecture is in agreement with aspects of visual agnosia. Farah (1990) described a dissociation between dorsal simultanagnosia and ventral simultanagnosia. The first results from damage in the dorsal pathway. Object recognition is still possible with these patients, but they cannot integrate that information with spatial information. Ventral simultanagnosia results from a damage in the ventral pathway, and it consists of a loss of identity information of objects. The double dissociation between these two forms of agnosia, the first can occur without the second and vice versa, is a classical argument against a holistic form of representation. Damaged holistic representations are always damaged holistically. On the basis of these and other forms of visual agnosia, Farah (1990, p. 151) presents a global model that is in agreement with the visual architecture we presented.

The visual architecture we presented here is in fact motivated by speed of processing. Initial processing in the visual cortex proceeds with feedforward networks (e.g., Oram & Perrett 1994), because processing in feedforward networks is very fast. But feedforward networks are prone to catastrophic interference (e.g., McCloskey and Cohen 1989). This results when the same feedforward network is used to learn different tasks. The best way to avoid catastrophic interference with feedforward networks is to use different feedforward networks for different tasks. This solution seems to be the basis for the structure of the visual cortex, which consists of different pathways that initially operate in a feedforward manner. But with different feedforward networks the integration of the information they process becomes a problem. The solution of that problem is the basis for the visual blackboard we have presented here. So, our visual blackboard architecture directly derives from issues related to speed of processing, and so from issues related to fast behavioral adaptation to changes occurring in the physical word. Moreover, these fast occurring changes are often related to new compositions of familiar entities (trees in a forest, houses and roads in a city). Handling novel compositions of familiar entities is the prime target of the visual blackboard architecture. It can do it straight away. No new learning, as in forming new holistic representations for these new combinations, is required. So, even in this respect, the architecture is faster than models based on holistic representation.

The grounding of language in the visual blackboard concerns the interaction between these two different types of blackboards. The fact that they interact is obvious. Visual information can be transformed into linguistic information, and linguistic information can be used to guide the visual system. The latter is obvious given the ability to direct visual attention by means of verbal instructions. This is not a form of priming, as **Dresp & Barthaud** mistakenly assume. Instead, it concerns the selection if visual information on the basis of linguistic information. The interaction between the visual and language blackboard derives from the fact that word assemblies are not copied or moved around, as in a symbolic architecture, but always remain "in situ". In this way, associations can develop between the word assemblies and other representations in the brain, such as representations in the visual cortex.

In a similar vein, **Powers** seems to deny the importance of language for human cognition, distinctive from other forms of cognition such as vision. However, this discussion itself illustrates the importance of language in cognition. We can have a discussion on a topic like the importance of language without any direct (visual) contact. Visual perception is always bound to the here and now, but language can make us go beyond the here and now. This is what makes language unique: people can exchange information without direct visual contact, without living in the same place, or even without living in the same time. In this way, knowledge can accumulate in a manner that is simply not possible if knowledge depended exclusively on direct visual contact with the world. Language can inform you about events that occurred thousands of years ago, or thousands of miles away. It can do so precisely because it can capture the actor-patient situation of a given event in the structure of a sentence. Readers can then understand the actor-patient situation of that event because they can understand the "Who does what to whom" relation expressed in the sentence. In other words, because they can answer (implicit or explicit) binding questions in the manner as we analyzed in our target article.

R3.2. Architecture versus association. In their commentary, **Doumas et al.** argue that Von der Malsburg (1986) has shown that connections between assemblies cannot carry information about binding relations. This analysis of processing in cell assemblies does not relate to the architecture we have presented. Cell assemblies are associative, but our architecture is not. Due to the effect of the gating circuits in the architecture, the flow of activation can be controlled in a manner that is not possible in an associative structure like a cell assembly. We illustrated this difference in figs 6 and 15.

Furthermore, **Doumas et al.** note that the architecture can answer a question like "Who is the cat chasing?", in the case of the proposition *chase*(*cat, mouse*), but that downstream neurons only "see" the neural activation of the assemblies involved, but not their binding relations. But why would these binding relations have to be represented explicitly by the downstream neurons, that is, why would they have to be burdened with representing that information themselves? There is a limit on the information that can be stored at a given site. So, when the need to store more information arises, e.g., when cognitive capabilities increase, information has to be stored in different sites. This is, of course, the reason why the cortex of humans is about four times larger than the cortex of apes (Calvin 1995). Distribution of information over different sites is not a problem as long as relevant information can be retrieved from distant sites. So, if the downstream neurons of **Doumas et al**. want to "know" whether *chase*(*cat, mouse*) or *chase*(*mouse*, *cat*) is stored by the neurons upstream, they can initiate a sequence of questions like "Who is chasing?" and "Who is being chased?". If the sequence of answers is *cat*, *mouse* (instead of *mouse*, *cat*), the downstream neurons "know" that *chase*(*cat, mouse*) is stored by the neurons upstream, instead of *chase*(*mouse*, *cat*). It is necessary that downstream and upstream neurons can influence each other. It is not necessary for each of them to do the task of the other. It would, in fact, be highly detrimental for the system as a whole if such a duplication of tasks was always needed.

For example, in the visual cortex the neurons in the "What" stream do not have to know where the object is and the neurons in the "Where" stream do not have to know what the object is. In this way, each stream can restrict learning to the specific task it has to perform. Learning would be much more difficult if each stream had to learn all tasks simultaneously. Yet, by influencing each other, the system as a whole knows where the object is and what the object is, as the interaction process in our visual architecture illustrates.

R4. Binding with synchrony

A number of commentators, **Dyer**, **Doumas et al.**, **Durstewitz**, **Müller**, **Shastri** and **Powers**, have argued against our dismissal of synchrony of activation as a binding mechanism in combinatorial structures. It is important to understand the topic of this debate. The topic is not so much whether or not one will find cross-talk in hierarchical representations. We are willing to grant that this can be avoided to a certain level (see Van der Velde and de Kamps 2002). The crucial issue concerns the "read-out" (i.e., answering binding questions) in the case of true forms of combinatorial structures, that is, structures for which no conjunctive (coincidence) representations exist. The commentaries that advocate synchrony as a binding mechanism do not really address this issue. There are two questions that need to be answered for synchrony as a binding mechanism in combinatorial structures (e.g., as found in language). How can binding questions be answered without relying on conjunction (coincidence) detectors needed to detect synchronous activation, and, if not, how can one have conjunction (coincidence) detectors for all possible binding relations that can exist in language? We are not aware of a model that has answered these questions.

R4.1. SHRUTI. In his commentary, **Shastri** provides further details of the reasoning process in SHRUTI. They are much appreciated, but they do not touch upon the topic of the discussion we have with this model. The topic is this: human cognition, in particular in higher levels of cognition, is characterized by productivity, compositionality and systematicity. These characteristics are related in the sense that one goes with the others (Fodor & Pylyshyn 1988). For example, the productivity of language requires linguistic structures to be compositional. If the 'performance set' of natural language (i.e., the set of sentences that human language users can comprehend) is at least in the order of 10^{20}, it is excluded to have a specific conjunctive forms of representation for each of these sentences. If only because the time to develop or learn these representations is not available. So, neural models of these forms of human cognition have to have the ability to represent and process cognitive structures in a productive, compositional and systematic manner. This, in short, is the topic of our discussion with models like SHRUTI.

SHRUTI is a model about reasoning in long-term memory. For instance, it can relate the proposition *John gives Mary a book*, stored in long-term memory, with the proposition *Mary owns a book*. However, the systematicity of reasoning entails that the relation between *give(X,Y,Z)* and *own(X,Y)* does not just hold for the specific tokens of this relation stored in long-term memory, but holds for all possible tokens of this relation. Viewed in this way, it does not make much sense to have a model that deals only with reasoning in long-term memory. It should be a model that can handle all tokens of the same kind, regardless of whether these tokens are stored in long-term memory or are represented on the fly (i.e., novel tokens of the same relation).

With this in mind we can have a closer look at SHRUTI. Figure 12 in Shastri and Ajjanagadde (1993) illustrates how SHRUTI can relate the proposition *John gives Mary a book* with the proposition *Mary owns a book*. This figure shows that the proposition *John gives Mary a book* is represented with a specific “fact node” (or, indeed, a “fact circuit”). On its own, it is acceptable to represent a proposition in long-term memory with a designated form of representation like a fact node or fact circuit. Long-term memory consists only of those propositions that we have actually memorized, so the problem of productivity that one finds with the performance set of language does not occur here. However, figure 12 in S&A (1993) shows that the fact node for *John gives Mary a book* plays a crucial role in the inference process that establishes the relation between *John gives Mary a book* with the proposition *Mary owns a book*. For example, activation of the fact node is needed to activate the “collector-node” of the predicate *give*. In turn, the activation of a collector node is needed to answer the queries in SHRUTI. So, if *John gives Mary a book* is stored in long-term memory, the collector node of *give* has to be activated to answer the query “Does Mary own a book?”. In turn, the collector node of *give* is activated by the fact node for *John gives Mary a book*, which shows that fact nodes play a crucial role in the reasoning process itself, instead of just representing a specific proposition.

The role of fact nodes in the reasoning process in SHRUTI raises the issue of the true systematicity of this model. Consider a novel proposition *John gives Dumbledore a broom* and the query “Does Dumbledore own a broom?”. **Shastri** asserts that SHRUTI can handle a novel event when it is told that “Dumbledore is an instance of an existing type”. We do not quite understand what this means and how it works, but we can argue that reasoning on the basis of type information in the case of specific propositions is a hazardous affair. *Dumbledore* is of the type *person* or *noun*, so it would seem that this solution entails that one can answer the query “Does Dumbledore own a book?” on the basis of the representation *John gives noun/person a book.* However, it is quite possible that *Dumbledore owns a book* is not true, even though *John gives noun/person a book* is true, e.g., because *John gives Mary a book*.

The reasoning process we discussed in section 6.6 does not depend on specific tokens of the kind *John gives Mary a book*. Instead, it depends only on the relations between the predicates *give* and *own*, and between the argument slots of these predicates. We used a long-term memory encoding of *John gives Mary a book* to illustrate how combinatorial structures can be encoded in a more explicit form in long-term memory. But the reasoning process described in section 6.6 works in the same way for a novel proposition like *Dumbledore gives Harry a broom* that is temporarily stored in the blackboard. The only difference between a long-term memory encoding and a temporal encoding of a proposition is the nature of binding of the arguments. In the temporal encoding, binding results from the (temporal) activity in the memory circuits, whereas in long-term memory binding results from direct associations between arguments and predicate slots. This difference in binding does not affect the reasoning process because that depends only on the associations between the predicates.

Furthermore , the reasoning process described in section 6.6 can answer specific binding questions like “Who owns the book?” instead of just yes/no queries like “Does Mary own the book?” in SHRUTI. This is a consequence of the fact that SHRUTI answers a query on the basis of the activation of a collector node, which is a yes or no event. In turn, the yes or no activation of a collector node reflects the yes or no activation of a fact node, needed to detect the respective synchrony between arguments and predicate slots.

The lack of systematicity in SHRUTI is also reflected in the problem of 2. **Shastri** acknowledges that the solution in SHRUTI for this problem, i.e., the duplication (multiplication) of predicate nodes, results in a loss of systematicity. Instead of having one relation between the predicates *give* and *own*, SHRUTI has distinct relations between $give_1$ and own_1 and between $give_2$ and own_2. **Shastri** asserts that our blackboard architecture is faced with a similar problem, because different tokens of a proposition are represented with different structure assemblies. This is correct, but that has no consequence for learning the relations that produce systematic reasoning, because these relations depend only on the associations between predicates, e.g., between the word assemblies for *give* and *own*, and predicates (word assemblies) are not duplicated in the blackboard architecture. The predicate *give* is always represented with one and the same word assembly in any proposition in which *give* occurs. Indeed, the very reason for the existence of structure assemblies next to word assemblies is that structure assemblies can be used to handle binding problems and the problem of two, whereas the (single) word assemblies can be used to account for relationships between predicates and issues like grounding as illustrated in figure 27.

Finally, **Shastri** asserts that it is odd that we did not discuss the neural parsing model of Henderson (1994), based on SHRUTI, as an alternative to our own. There is a simple reason why we did not do this: there is no neural parsing model to be found in Henderson (1994). Henderson introduces a parser in which computational constraints derived from SHRUTI are used. But the parser itself operates entirely in a symbolic way. For example, Henderson (1994, p. 367) describes how syntactic nodes can be closed for further parsing: “Closed nodes can be removed from the phrase structure representation, thereby reducing the number of nodes [the parser] needs to store information about”. Removing nodes from a structure is one of those things that one can do easily in a symbolic implementation, but it turns out to be very hard (if not impossible) in a genuine neural implementation. None of these issues about real neural implementation are discussed, let alone answered, in this parsing model. None of the problems that derive from these implementation issues, like answering specific binding questions without using conjunctive representations, are discussed, let alone answered, in this parsing model.

R4.2. LISA. The fundamental problem with any model using synchrony of activation, as we stated in section 3.2, is the dependence on conjunctive forms of encoding to get things done. Dependence on conjunctive encoding destroys the productivity and systematicity of a model. We analyzed this in case of SHRUTI, but we can give the same analysis in the case of the model advocated by **Doumas et al.**, known as LISA. For example, LISA encodes *Bill loves Mary* by having conjunctive (coincidence) nodes for the entire proposition, for *Bill-lover* and for *Mary-beloved* (see figure 3, Hummel and Holyoak 2003). So, the question is how LISA can represent, and operate on, a novel proposition like *Dumbledore loves Harry*? How could one have a conjunctive node for this proposition if one has never seen or heard it before? By the same token, how could one have a conjunctive representation of *Dumbledore-lover* and *Harry-beloved*? Furthermore, it is possible to have the conjunctive representations for one relation, but not for the reverse relation. So, one could have conjunctive nodes for *Harry loves Dumbledore*, *Harry-lover* and *Dumbledore-beloved*, but not for *Dumbledore loves Harry*, *Dumbledore-lover* and *Harry-beloved*. Consequently, LISA could make inferences about *Harry loves Dumbledore* but not for *Dumbledore loves Harry*. This kind of behavior is characteristic of a system that lacks systematicity.

The inference process in LISA critically depends on its conjunctive form of representation. An example can be found in figure 4 in Hummel and Holyoak (1997). The figure illustrates a mapping from *John loves Mary* onto *Bill likes Susan* versus *Peter fears Beth* (i.e., the inference that *John loves Mary* is more resemblant of *Bill likes Susan* than of *Peter fears Beth*). The process starts with the activation of the node for *John loves Mary*. This node, in turn activates conjunction nodes for *John-lover* and *Mary-beloved*, which then activate a set of nodes for semantic primitives, which in turn activate conjunction nodes for conjunctions like *Bill-liker* and *Bill likes Susan*. The need for synchrony is obvious: there are so many conjunctive nodes, both for entire propositions and for partial propositions, that without synchrony a superposition catastrophe would easily occur. But this is not what systematic reasoning is about. The purpose of systematic reasoning is not to avoid a superposition catastrophe in a selected set of conjunctive representations, but to establish relations between arbitrary tokens of familiar predicates. This is where LISA fails. LISA cannot conclude that a novel proposition like *Dumbledore loves Harry* is more resemblant of *Bill likes Susan* than of *Peter fears Beth*, because the conjunctive representation of *Dumbledore loves Harry* and its relations with representations for *Dumbledore-lover* and *Harry-beloved* are missing in LISA. Yet, the resemblance between *Dumbledore loves Harry* and *Bill likes Susan* is just as great as the resemblance between *John loves Mary* and *Bill likes Susan*. LISA, however, is completely blind to the systematicity between these two tokens of the same resemblance. This is why synchrony of activation in a model like LISA fails to encode productive and systematic cognitive structures and relations.

R4.3. Role of synchrony. In his commentary, **Durstewitz** argues that synchronization could be mechanism for binding without the use of specific coincidence detectors, because our architecture also operates with assemblies and subassemblies that are synchronously active. However, the word “synchronous” is used with two different meanings here: simultaneous and true synchrony. The members of a band can play in the same period (simultaneous), even if they do not play in (true) synchrony. The first meaning of “synchronous” as used by **Durstewitz** refers to simultaneous activation in our architecture. The populations in the architecture are simultaneously active in a given time period, in the sense that they generate enough activity within that period to influence each other (just as band members can produce sound in the same period). The second meaning of the word “synchronous” as used by **Durstewitz** refers to phase coherence. This requires that populations are not just simultaneously active in a given period, but that they also play in the same rhythm, so to speak. It is not required that the activations of the populations in the architecture show any phase coherence. In contrast, phase coherence is essential in models that use synchrony as a binding mechanism, such as SHRUTI and LISA. If one would take away the phase coherence in these models, the units/neurons in these models would still be simultaneously active. But the ability for binding is lost in these models, because that ability depends critically on phase coherence, which is detected by the coincidence detectors in these models.

It is important to note, however, that our rejection of synchrony as a binding mechanism in combinatorial structures does not entail a rejection of any role of synchronous activation in the brain. Synchrony of activation can play a role in processes that depend on conjunctive forms of representation. As we argued in R3.1, if an organism can solve a problem with conjunctive representations, it should do so. As **Durstewitz** noted, time is a serious constraint in biology, and conjunctive forms of processing a much faster than combinatorial forms of processing. However, for this solution to succeed, the problem space at hand should not be too large. Otherwise, the conjunctive representations that the organism can learn in its life time will not sufficiently cover the problem space. In that case, there is a reasonable chance that the organism can be confronted with a problem for which it has not (yet) developed a conjunctive representation.

The basis for combinatorial structures in human cognition is the observation that problem spaces arise in human cognition that are too large to be covered in any reasonable way with conjunctive forms of representation. Here, the need for combinatorial processing arises. We used the example of the performance set of natural language, which consists of 10^{20} sentences or more. In any lifetime, one can form only a minute set of conjunctive representations for this set (in the order of one sentence for every 10^{10} sentences or more). In visual cognition, an example would be the number of different patterns that can be created on a chess board.

But for problem spaces that are not too large, conjunctive representation would be the preferred choice. Many examples can be found in the visual cortex (e.g., the orientation columns in the primary visual cortex). In the case of conjunctive representation, synchrony can help to avoid the superposition catastrophe that can easily occur with conjunctive representations, as suggested by Von der Malsburg (1987). So, historically, the use of synchrony arose out of a problem with conjunctive representations. Synchrony solved the superposition catastrophe with these representations by using coincidence detectors as conjunctive units. It is important toe realize, that this is a refinement of the use of conjunctive representation, not an introduction of combinatorial representation.

The empirical evidence for synchrony of activation is in line with this interpretation. In a characteristic experiment (e.g., Singer & Gray, 1995), one or more light bars are moved across the receptive fields of two groups of neurons. A synchrony of activation was found between both groups of neurons when a single light bar (i.e., a coherent object) was moved across the receptive fields of both groups of neurons. But when two light bars were moved in opposite directions, each one across the receptive field of one group of neurons, synchrony of activation did not occur. As we argued in section 2.1, problems such as motion detection are most likely solved in conjunctive manner in the visual cortex. The same could be true for the detection of motion coherence of a moving object, as in the experiment of Singer and Gray. Given the occurrence of conjunctive coding in the visual cortex, it is not a surprise that most of the empirical evidence related with synchrony of activation has been observed in the visual cortex.

Besides a mechanism for binding in these forms of conjunctive processing, synchrony of activation as observed in the brain could also be a “dynamical fingerprint” of an interaction that occurs between brain areas. When brain areas exchange information in functional manner, in the context of the cognitive process they instantiate, they will also interact as coupled dynamical systems. Coupled dynamical systems have a tendency to synchronize their behavior, because of the (oscillating) force they exert on each other. In the same way, brain areas that interact in a cognitive process could have a tendency to synchronize their dynamics as well. This form of synchronization does not have to play a functional role in the cognitive process, as in answering binding questions. Instead, it could just be a dynamical by-product of the interaction between brain areas. But it provides important information about which areas interact (communicate) in a given cognitive process. That is, it serves as a “dynamical fingerprint” of an interaction between brain areas. The fact that these synchronization effects are sometimes detected only with sophisticated mathematical techniques, of the kind that intelligence services use to eavesdrop though windows or thin walls, corroborates this view. It is difficult to see how such a faint effect could play a functional role in a cognitive process. But the effect does not have to be strong to be a dynamical fingerprint of an interaction between brain areas.

R5. Recurrent networks and language

Dyer, Durstewitz and **Müller** argue that we have discarded RNNs (of the kind used by Elman 1991) too easily as potential neural models of language processing. For example, **Durstewitz** notes that the fact that RNNs could not (in our simulations) learn generative properties of language does not entail that they could not represent these properties. In a similar vein, **Müller** notes that RNNs with more than one hidden layer (e.g., 100) could represent much more than the RNNs we have tested. Indeed, it would not be justified for us to criticize RNNs on their learning (in)abilities, considering the fact that we have not yet shown how learning could take place in our architecture.

However, that is not the basis of our argument. Instead, we do indeed seriously question the abilities of RNNs (Elman style) to represent the generative capabilities of language, whether they have one or a 100 hidden layers. To understand why we make this claim, it is important to realize that these models arose out of an attempt to model language performance without using the distinction between rules and a lexicon. Linguistics has (typically) assumed that the distinction between rules and a lexicon is necessary to account for the productivity of language. For example, with the rules one could create a sentence (syntactic) “skeleton”, which in combination with the lexicon could generate a huge number of sentences. In contrast with this approach, RNNs directly operate on word strings, and they learn the words and the sentence structures at the same time. In other words, the aim is to represent the generative capabilities of natural language by deriving representations of all possible sentence on the basis of a set of sentences actually learned by the network.

The problem with this approach does not reside in the number of hidden layers, or in the specific learning algorithm used. The problem resides in the contrast between the number of sentences that can be learned by the network (in any reasonable amount of time) and the number of sentences that can be formed on the basis of the combinatorial productivity of language. This performance set is in the order of 10^{20} or more. In contrast, even if a child learns a sentence every second for the first 20 years of life, the set of learned sentences is only in the order of 10^{9}. This is just one sentence for every 10^{11} sentences in the performance set. In other words, between the sentences learned by the network there are gaps of 10^{11} sentences (or more) about which the network has had no direct information.

The significance of these gaps can be appreciated by looking at the mathematical background in these RNNs (Van der Velde, 2001). In performance and learning the RNNs operate as feedforward networks (the feedback connections are frozen in these moments). In fact, the feedback connections determine only the input given to the network in performance or learning (which consists of a word in the sentence and the activation state in the hidden layer produced with the previous input). The reason why it is possible to process a sequence of words with RNNs results from the input presentation: the activation of the hidden layer with input n is part of the input n+1. In this way (and only in this way) does the previous processing in the network influence the current processing.

In mathematical terms a feedforward network implements a function between an input vector space and an output vector space. Hornik et al. (1989) showed that feedforward networks are “universal approximators” for a wide class of functions. This means that for any function in this class one can find a feedforward network that approximates that function on a subset of the input space within any desired level of accuracy. The proof by Hornik et al. is based on the Stone-Weierstrass theorem, which is itself a generalization of a theorem originally discovered by Weierstrass (e.g., see Rudin 1976). The Weierstrass theorem states that a continuous function f(x) can be approximated on a finite interval of its input domain by a sequence of polynomials. This theorem is the mathematical foundation of important approximation methods such as Fourier series and Legendre polynomials (e.g., see Byron & Fuller 1992).

The limitation of the approximation to a finite interval is the basis for the fact that feedforward networks can approximate functions only on limited subsets of the domain. Furthermore, the ability to approximate a function derives from the use of a set of input values and their corresponding function values to determine the coefficients of the polynomial, or the weights of the feedforward network (e.g., by using the backpropagation algorithm). For this to succeed, the Weierstrass theorem demands that the function is known sufficiently ("globally") over the entire interval on which the function is approximated. That is, the gaps between the input values used to set the coefficients or weights cannot be too large (Byron & Fuller 1992).

As stated, RNNs are feedforward networks during learning, that is, when the weights of the network are set to determine the approximation of the function that the network computes. Thus, as demanded by the Weierstrass theorem, the gaps between the input values used to set the coefficients or weights cannot be too large. This restriction is clearly violated in the case of language, because the set of sentences that can be learned in a lifetime is minute compared to the performance set of language. As we illustrated above, gaps of 10^{11} sentences or more inevitably occur.

Furthermore, it should also be mentioned that the tasks that RNNs perform are only remotely related with language performance. All they can do is predict the lexical category of a word that can follow after a given input string. They cannot, however, answer specific binding questions ("Who does what to whom") for the sentences they process. As Jackendoff (2002, p 163) noted about language performance with RNNs: "virtually everything of linguistic relevance has been bled out of the task the network is designed to perform".

The reference of **Durstewitz** to Hanson and Negishi (2002) is not relevant here. First of all, this paper is about learning finite-state languages with neural networks. It is well known that neural networks can handle languages of this kind. Furthermore, the language is an artificial language, that is, the set of words in these languages is very small. The crucial problem for natural language is in dealing with languages that are more complex than a finite state language, and which have a very large lexicon.

R6. Reduced vector coding

Gayler and **Sommer & Kanerva** argue for an alternative solution of combinatorial structures in terms of reduced representations in high dimensional vector spaces. In the target article, we briefly referred to problems associated with tensor networks as models for combinatorial structures, in particular the problem that tensor networks increase in size with levels of binding. Both **Gayler** and **Sommer & Kanerva** acknowledge this problem, but they note that the problem can be solved with reduced forms of representation, as given by Vector Symbolic Architectures (VSAs). In a VSA, the binding between two (or more) vectors does not result in an increase in size. Instead, the constituent vectors and the combinatorial vectors have the same dimensionality. An example of a VSA is the Holographic Reduced Representation (HRR) proposed by Plate (1997; 2003). So, with VSAs the problem of tensor networks is avoided, and one would have distributed vector processing as the basis of neurocognition.

It is true that VSAs such as HRR solve the dimensionality problem of tensor networks. However, the solution itself prevents their use in truly combinatorial (novel) structures. We will use HRR to illustrate why this is the case. Consider the proposition *John eats fish*. A HRR representation of this proposition proceeds as follows (e.g., Plate 1997; 2003). First, *John*, *eats*, and *fish* are represented as n-dimensional vectors. The exact value of "n" is not crucial, but it is crucial that all vectors have the same dimensionality. So, the n-dimensional vectors **John**, **eat**, and **fish** represent the words *John*, *eat*, and *fish*. Furthermore, the argument roles of *eat* are also represented with n-dimensional vectors, giving $\mathbf{eat}_{\mathbf{agent}}$ and $\mathbf{eat}_{\mathbf{theme}}$. The binding of **John** with $\mathbf{eat}_{\mathbf{agent}}$ is achieved with an intricate mathematical operation called circular convolution. Fortunately, we do not have to consider the details of this operation, it suffices to know that it produces an n-dimensional vector $\mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{John}$ that represents the binding in *John eats* (⊗ represents the binding operation). The fact that $\mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{John}$ has the same dimensionality as **John**, **eat** and $\mathbf{eat}_{\mathbf{agent}}$ shows that the size problem of binding in tensor networks is solved. In the same way, $\mathbf{eat}_{\mathbf{theme}}\otimes\mathbf{fish}$ represents the binding in *eats fish*.

The entire proposition *John eats fish* is represented with the vector $\mathbf{P_1} = \langle \mathbf{eat} + \mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{John} + \mathbf{eat}_{\mathbf{theme}}\otimes\mathbf{fish}\rangle$. $\mathbf{P_1}$ is just an arbitrary label for this vector. It is also an n-dimensional vector. The symbol + denotes the operation of superposition, which basically consists of adding the component values of the vectors involved. The brackets ⟨ ⟩ denote that $\mathbf{P_1}$ is normalized, i.e., its component values are kept with a given range. The proposition *fish eats John* can also be represented in this way, with $\mathbf{P_2} = \langle \mathbf{eat} + \mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{fish} + \mathbf{eat}_{\mathbf{theme}}\otimes\mathbf{John}\rangle$. The vectors $\mathbf{P_1}$ and $\mathbf{P_2}$ are different, so the role reversal of *John* and *fish* is adequately represented with HRR. Because a proposition is itself an n-dimensional vector, it can be used as an argument in another proposition. For example, consider *Susan knows John eats fish*. Here, $\mathbf{P_1}$ is the "theme" argument of **know**. The entire vector for this proposition is $\mathbf{P_3} = \langle \mathbf{know} + \mathbf{know}_{\mathbf{agent}}\otimes\mathbf{Susan} + \mathbf{know}_{\mathbf{theme}}\otimes\mathbf{P_1}\rangle$.

Because vectors are normalized, some loss of information occurs. However, it is remarkable to see how much information can be encoded in this kind of representation (e.g., see Plate 2003). Information retrieval with HRR proceeds by decoding bindings. For example, *John* as the argument of *John eats* can be found by convolving the vector $\mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{John}$ with the "approximate inverse" $\mathbf{eat^*}_{\mathbf{agent}}$ of $\mathbf{eat}_{\mathbf{agent}}$. The result $\mathbf{eat^*}_{\mathbf{agent}}\mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{John}$ is a noisy version of **John**. This noisy version of **John** has to be "cleaned-up" by finding its closest-match in a clean-up memory. This process is a form of pattern recognition that produces **John** as its outcome when the input is $\mathbf{eat^*}_{\mathbf{agent}}\mathbf{eat}_{\mathbf{agent}}\otimes\mathbf{John}$. The clean-up memory must contain all patterns of both objects and structures which can result from decoding a HHR in the system.

These examples illustrate that HRR (and VSAs) can encode combinatorial structures. However, encoding is just one part of the story. It is also important that information can be retrieved in a combinatorial manner, that is, without relying on conjunctive forms of representation. Moreover, the system itself has to be able to retrieve that information. With this in mind we can have a closer look at the retrieval process in HRR.

Consider the proposition *John eats fish* and *Susan knows John eats fish*. Suppose we ask the binding question "What does John eat?" to an HRR system. When the proposition *John eats fish* is stored in its memory, the system would have to answer the question by first decomposing the vector $\mathbf{P_1} = \langle \mathbf{eat} + \mathbf{eat_{agent}} \otimes \mathbf{John} + \mathbf{eat_{theme}} \otimes \mathbf{fish} \rangle$ into **eat**, $\mathbf{eat_{agent}} \otimes \mathbf{John}$, and $\mathbf{eat_{theme}} \otimes \mathbf{fish}$. Then it would have to produce $\mathbf{eat^*_{theme} eat_{theme}} \otimes \mathbf{fish}$, and present this vector to the clean-up memory to produce **fish** as the answer. However, when the proposition *Susan knows John eats fish* is stored in its memory, the system would first have to decompose the vector $\mathbf{P_3} = \langle \mathbf{know} + \mathbf{know_{agent}} \otimes \mathbf{Susan} + \mathbf{know_{theme}} \otimes \mathbf{P_1} \rangle$ into **know**, $\mathbf{know_{agent}} \otimes \mathbf{Susan}$, and $\mathbf{know_{theme}} \otimes \mathbf{P_1}$. Then, it would have to compute $\mathbf{know^*_{theme} know_{theme}} \otimes \mathbf{P_1}$, retrieve $\mathbf{P_1}$ from the clean-up memory, and execute the entire sequence of operations described for the proposition *John eats fish*.

So, to answer the question "What does John eat?" different sequences of operations have to be carried out by the system, depending on the vector (e.g., $\mathbf{P_1}$ or $\mathbf{P_3}$) stored in its memory. To make a selection of the sequence of operations to be carried out, the system has to know which of these two vectors is stored in it memory. However, both $\mathbf{P_1}$ and $\mathbf{P_3}$ are n-dimensional numerical vectors. On the surface they look alike. How is the system to know which of these is stored in its memory, and so which sequence of operations it has to carry out to produce the answer to the binding question? For example, assume that *John eats fish* and *Susan knows John eats fish* are both novel propositions (of familiar constituents), never seen before by the system. In that case, $\mathbf{P_1}$ and $\mathbf{P_3}$ are both novel n-dimensional numerical vectors, never seen before by the system. So, on what kind of information can the system decide that, e.g., $\mathbf{P_3}$ is stored in its memory, so that it has to execute the sequence of operations needed for $\mathbf{P_3}$? To argue that there will be some kind of representation of the proposition in the system that could guide the decoding processing is, of course, begging the question. The n-dimensional vector is the representation of the proposition in the system, and nothing else.

This problem illustrates an important aspect of combinatorial or compositional structures. Combinatorial structures are not only structures that are based on constituents, but the constituents have to be recognizable within the combinatorial structure itself. This is where reduced vector coding fails. As a result of the reduced representation, the constituents are no longer recognizable within the combinatorial structure. Instead, they are encapsulated within the combinatorial structure. Because the n-dimensional vector of a familiar constituent is not recognizable within the n-dimensional vector of a novel combinatorial structure, it cannot be used to guide the process of answering binding questions (as it does in the architecture we present here, e.g., see figure 7a).

Sommer & Kanerva argue that serious progress in cognitive modeling will be based on the understanding of the mathematical properties of high-dimensional representation spaces rather than on a specific solution to a relatively narrow set of challenge problems, i.e., the ones we discussed here. Of course, more knowledge of mathematics is always useful, but cognitive modeling is not just an exercise in mathematics. The aim is to capture cognitive and brain processing in a cognitive model in such a manner that "boundary conditions" related with cognition and the brain are incorporated in the model. One such boundary condition is the ability to answer binding questions related to novel combinatorial structures. The benefit of finding specific solutions to a relatively narrow set of challenge problems is that these boundary conditions are incorporated from the beginning.

R7. Neural basis

Choe addresses the issue of the neural instantiation of our blackboard architecture for sentence structure. It is interesting to see that a gating circuit can be identified in a cortico-thalamic loop. However, like **Choe**, we do not believe that the thalamus would be the site of the neural blackboard proposed here. In our view, the neural blackboard is most likely to be found in the cortex. This is one of the reasons that we included a discussion of the visual blackboard, to illustrate that processes in the blackboard (e.g., gating, memory by delay activity) can be found in the cortex. **Choe** describes a gating circuit, based on disinhibition, in the cortico-thalamic circuit. Gonchar and Burkhalter (1999) described a disinhibition circuit in the visual cortex of the rat. In terms of evolution, the visual cortex is one of the oldest parts of the cortex. So, if disinhibition circuits are found in the visual cortex, and in the cortex of animals like rats, it could be that they belong to the “building blocks” that determine the structure of the cortex (also given the structural similarity found within the cortex). Delay activity is know to be found in the cortex as well (e.g., Fuster 1995), so the building blocks for the gating circuits in our architecture could exist within the cortex.

Furthermore, the binding structure we illustrated in fig. 5 should be seen as a structure of cortical columns. Each column could be a microcolumn or perhaps hypercolumn in the cortex. They would all consist of the same kind of circuit, i.e., a gating circuit based on disinhibition, and the same kind of delay assembly. Regular column structures can be found in the cortex. An example is the “ice-cube” model of the primary visual cortex (e.g**.**, Coren et al. 1999) that consists of a very regular pattern of columns, each dedicated to a specific task, such as orientation detection of edges in a particular direction. In fact, each of these columns is a local representation of an elementary shape on a given retinal co-ordinate. So, the brain can develop regular cortical structures with local representation when it needs it for specific purposes. We do not see why such regular and even local structures could not have evolved for language processing as well.

Examples of regular structures and more or less local representation can be found in the language areas. For example, some stroke patients can name tools, but not animals. Other specific naming disabilities include plants, body parts and verbs (Pinker 1994, Calvin & Bickerton 2000). The temporal cortex seems to be organized in terms of categories of concepts, with different sites for different concepts. The temporal cortex, in turn, is connected with the prefrontal language areas with an important fiber bundle (the arcuate fasciculus), with axon branches that produce a very regular pattern of connectivity (Calvin & Bickerton 2000). In fact, the cortex has a much more regular structure than it is sometimes credited for (Hubel 1995).

A regular structure of columns also answers the question of **Cloe** about the duplication of structure assemblies. They are not duplicated on the fly, but they form an existing structure, (produced in a period of development). Again, the ice-cube model in the visual cortex comes to mind. Here, one also finds a regular structure, with repetitions (duplications) of the same representations (e.g., orientation columns).

R7.1. Dynamics. In their commentary, **Grüning & Treves** argue that we should combine our architecture with an effective use of cortical attractor dynamics. It is indeed our intention to do so. In fact, there is more attractor dynamics implied in the architecture than we were able to illustrate and describe. The word assemblies are indeed assumed to be attractor-like networks, and not just symbolic representations in neural terms. This impression may follow from the fact that we have connected word assemblies directly to the structure assemblies in the blackboard. However, as we discussed in R3, word assemblies will interact with the sentence blackboard through a phonological blackboard. The word assemblies are located outside the phonological (and sentence) blackboard. They would consist of attractors that are more or less distributed.

In terms of attractor dynamics, one could think of word assemblies as different attractors in the same network. That is, each word assembly would be an attractor state of the same set of neurons. This is most likely not the way in which word structures are instantiated in the brain. It does not agree, for example, with the fact that very specific loss of word use can be found in patients. With anomic patients (i.e., patients that have trouble using nouns) one can find patients that have difficulties in using concrete nouns but not abstract nouns, or vice versa. Similarly, one can find patients that have difficulties in using nouns for living things but not for non-living things, or vice versa. Other examples include specific deficits for naming animals, body parts, fruits and vegetables, colors, proper names, or objects that are typically found indoors (e.g., Pinker 1994). If al these words were instantiated as attractors in the same network, that would be hard to explain.

Therefore, we assume that word assemblies are partly overlapping, sharing network structures with some words in one domain and with other words in other domains. In each of these domains one will find attractor dynamics that selects an attractor state in these domains. Together, the domains interact to select the overall word assembly at a given moment. The dynamics that select a word assembly will consist of a combination of attractor dynamics and interactive activation. The word assemblies we have shown are just those parts of the overall word assemblies that are connected to the sentence blackboard (or better, to the phonological blackboard).

R7.2. Hippocampus. The role of the hippocampus in our architecture was discussed by **Shastri** and **Whitney**. The hippocampus (HC) model we used to explain one-trial learning is the "Hebb-Marr" model. The neuroscientific evidence for this model is well-documented (e.g., Rolls & Treves 1998). This model, as explained in section 6.5, provides a "snapshot" memory of about a second of an ongoing event. We showed how a sentence structure can be encoded in this way. Contrary to **Shastri**'s assumption, longer sentences can also be encoded in HC in terms of separate but partly overlapping events, as illustrated in fig. 12. The fact that a sentence is temporarily encoded in HC does not mean that every sentence will eventually be encoded in long-term memory, or that a sentence will always be encoded in the way in was presented. In fact, the role of the HC in memory encoding is to form a temporal storage of information that can be used in an elaborate process that incorporates aspects of that information into existing cortical memory structures. A discussion of this process, and of the role of the HC, can be found in O'Reilly and Rudy (2001).

Whitney notes that the sentence structure in the HC could interfere with the sentence structure in the blackboard, because of an overlap in the structure assemblies used. This is not a real issue. First of all, the HC does not play a direct role in sentence processing in our model. It's role is in the transfer of information to long-term memory. But the issue of using the same structure assemblies would not really be important even if HC would be involved (e.g., in longer sentences). It is likely that a kind of "inhibition of return" occurs that prevents recently used structure assemblies to be used again. So, the chances of an overlap, even for longer sentences, are small.

R7.3. Central pattern generator

The central pattern generator (CPG) does not regulate the flow of activation in a gating circuit, as assumed by **Choe**. It regulates the onset and offset of different stages of competition in the process of answering a binding question. In this way, it resembles motor control, in which the onset and offset of muscle activation has to be regulated. Motor control is a type related problem, because it consists of controlling a specific set of muscles. In a similar way, the competition in the blackboard is a type related problem as well, because it operates on the level of structure assemblies, not word assemblies. The control of the competition process in the blackboard is not programmed, as assumed by **Durstewitz**. Instead, as suggested by **Grüning & Treves**, it will have developed on the basis of self-organisation during language development. Because the control of the blackboard as given by a CPG does not depend on the specific content in the blackboard, it is possible to learn over time how control of specific sentence types (or clause types) proceeds in the blackboard. Further understanding of how motor control proceeds in the brain will be useful for understanding dynamic control in the blackboard as well. But, in addition to motor control, linguistic processing is faced with the combinatorial productivity of language. The blackboard architecture was introduced to solve this issue

R7.4. Development and evolution. The issue of how an architecture like the one we have presented can develop and evolve was raised by **Durstewitz**, **Grüning & Treves**, **Hadley** and **Müller**. This is an important topic for further research. To study these issues, we have chosen a "backtrack" approach (Van der Velde 2005). The aim of this approach is first to develop an architecture that can handle a number of linguistic issues (including human processing) reasonably well. Then, this architecture can be used as a "target" architecture, to investigate how this target architecture can develop on the basis of a more simplified version of it, in combination with learning and development procedures. The benefit of this approach is that the information available on language and processing can be used to guide the process of development. Furthermore, the functionality of the target architecture is known, so there is no risk of getting stuck halfway. That risk is a serious problem of the procedure in which language development and evolution is modelled on the basis of arbitrary (i.e., unstructured) architectures and learning procedures. There is no guarantee that language can develop on the basis of any given neural structure. In fact, most species have not developed brains that can handle cognition of the complexity comparable to language. With this procedure, one could get stuck halfway, that is, even after some initial success, the initial architecture might not develop further to a more complete architecture for language (Van der Velde 2005).

The fact that, for example, orientation maps and single cell selectivity to orientation are already available at birth (e.g., Chapman 1996) is in line with this approach. Apparently, basic elements of structure in the brain are available before learning. The disinhibition circuits in the cortex (e.g., Gonchar & Burkhalter 1999), the circuits for delay activity in the cortex (e.g., Fuster 1995), and the regular pattern of connectivity in the superficial layers of the cortex (e.g., Calvin 1995) could also belong to the building blocks of brain structure that are available before learning occurs. In this way, the development of the blackboard could indeed be the result of a process of self-organization as suggested by **Grüning & Treves**.

R8. Linguistic issues.

A number of commentators have raised interesting linguistic issues concerning the blackboard for sentence structure, and the processes related with it. Further development of the language blackboard is a topic for future research. But, as Jackendoff (2002) noted, a solution of the problems he described will produce a more productive dialogue between linguistic theory and neural network modeling. The issues raised by the commentators illustrate the beginning of such a dialogue.

R8.1. Complexity. We have argued that aspects of the neural dynamics in the architecture (e.g., dynamic interactions) can account for some complexity effects observed in linguistic processing. A more substantial account of these effects will depend on a further development of the architecture. However, the effect of dynamics can be used to illustrate the complexity difference between complement clause within relative clause (RC/CC) versus relative clause within complement clause (CC/RC) referred to by **Whitney**. The examples provided by Gibson (1998) are:

CC/RC: *The fact that the employee who the manager hired stole office supplies worried the executive.*

RC/CC: *The executive who the fact that the employee stole office supplies worried hired the manager.*

The RC/CC sentence is more complex than the CC/RC sentence. Fig. R2 provides the basic structure of the sentences in terms of the blackboard architecture (ignoring *the* and *office*). On the face of it, they are similar and of equal complexity. However, in terms of the dynamics in the architecture, the situation is different. Delay activity, as used in our model, is based on stochastic neural activity (e.g., Amit 1989). This will have at least two effects on complexity. First, the longer it takes for a binding to occur, the greater the chance that delay activity (of a subassembly) has disappeared before a binding needs to be produced. Secondly, any binding conflict between subassemblies (even of an unequal nature) will affect the complexity of a sentence. In this respect, the sentences in fig. R2 are different. In the RC/CC sentence the binding of *executive* (with *worried*) spans a greater interval of time compared to the CC/RC sentence. Furthermore, there is a conflict with the binding of *stole* with *supplies*, which occurs within this interval. In the CC/RC sentence, the interval of binding *employee* with *hired* is shorter, and there is no conflict in between. An additional aspect of complexity may result from the ambiguity associated with the word *fact*. This word can introduce a complement clause, but it can also be a theme of a next verb. The human language processor, operating in an incremental manner, often makes a choice between these ambiguities. It could be that when *fact* is found at the beginning of sentence, it is interpreted more often as a noun that introduces a complement clause (based on a familiarity with this sentence type). Instead, when *fact* occurs within the sentence, it could also be interpreted as a noun that can be the theme of the next verb. In that case, the RC/CC sentence in fig. R2 is even more complex than the RC/RC sentence in fig. 18.

The fact that delay activity, in particular in binding conflicts, can be disrupted (i.e., disappear) could account for the observation that RC/RC sentences can be perceived as grammatical when the second verb has been omitted, as noted by **Whitney**. For example, if the activity of the theme subassembly of *cat* in fig. 18a is disrupted (due to the long binding interval, and binding conflict), the sentence without *bites* would indeed be perceived as grammatical (no

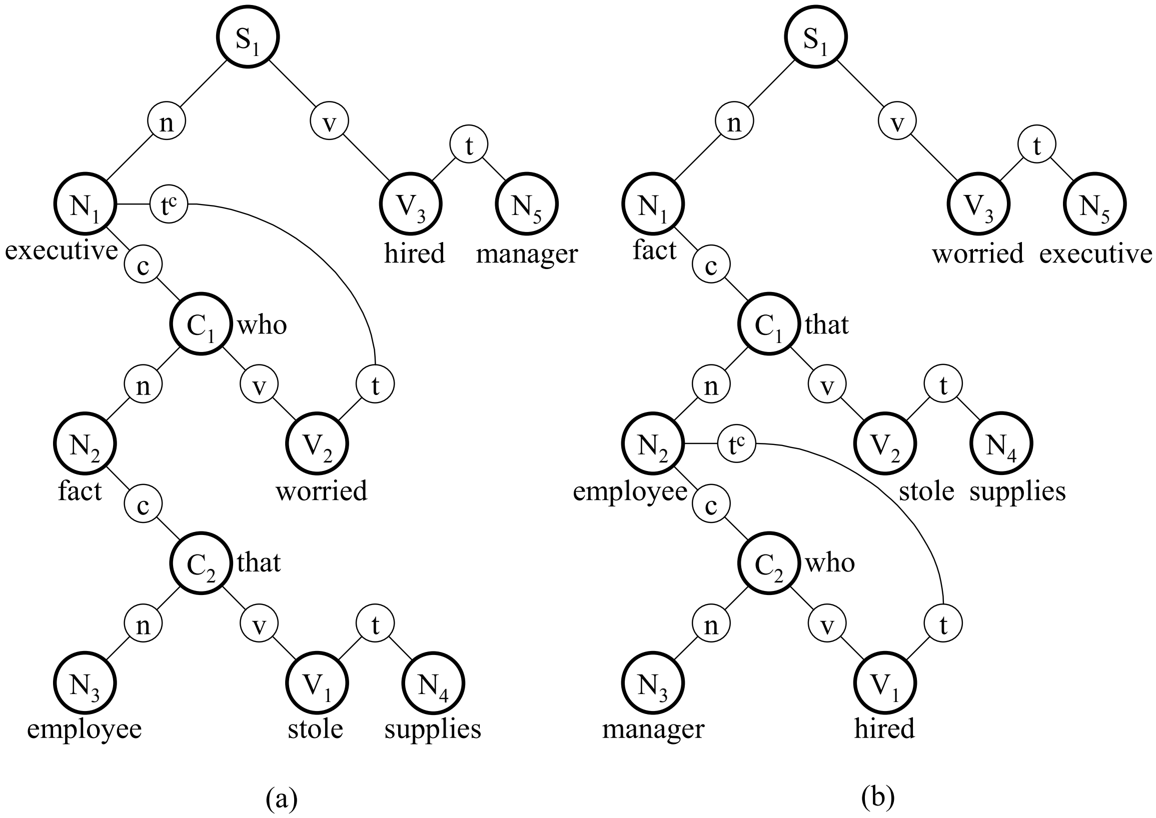

Figure R2. (a). Illustration of the structure of *The executive who the fact that the employee stole office supplies worried hired the manager* (without *the* and *office*). (b). Illustration of the structure of *The fact that the employee who the manager hired stole office supplies worried the executive* (without *the* and *office*).

open bindings or conflicts remain with this sentence).

Whitney argues that her TPARRSE model could provide an alternative to our architecture. Unfortunately, we are not familiar with the details of this model, so a more detailed comparison with our architecture will have to wait. However, there are a few aspects about this model, referred to by **Whitney**, that raise doubts about her model. The first is the aspect of vector coding and vector operations for binding in this model. In section R6 we discuss vector coding and binding in more detail. It seems that there are two possibilities. Either, the binding operations are those as found in tensor models. In that case, the need for an increase of structure with depth of binding will be a serious problem for **Whitney**'s model, as it is for tensor networks. The second option is the use of reduced vector representation to account for binding. However, we argue in R6 that there is a fundamental flaw with this form of binding. Constituent representations are encapsulated in reduced vector coding, so they cannot be used to guide the process of answering binding questions. This means that, certainly in novel binding structures, the system has no information to decide on its own what sequence of operations it has to execute to answer a given binding question.

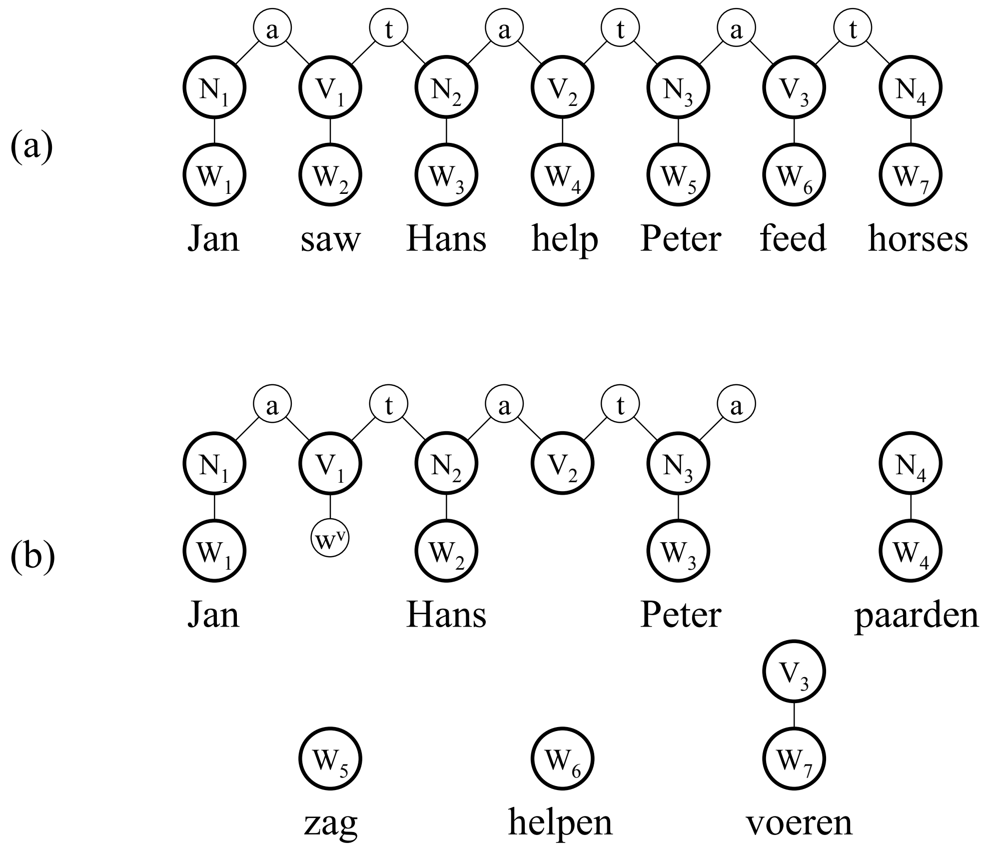

Figure R3. (a). Illustration of the clause *Jan saw Hans help Peter feed the horses*. (b). Illustration of the Dutch clause *Jan Hans Peter de paarden zag helpen voeren*.

The second aspect of **Whitney**'s model that raises doubts is the use of "firing order" to implement a stack memory. How can firing order distinguish between different occurrences of the same word? Consider, for example, *A spy to spy on our spy who searches for their spy* (Van der Velde 1999). The noun *spy* occurs in three positions in this sentence: beginning, middle and end. So, it has to have three firing orders at the same time. It would seem, though, that any new occurrence of a word will eliminate its previous firing order, leaving only the last firing order for *spy* in this sentence.

R8.2. Cross-serial dependency. In her commentary, **Whitney** also referred to the relative ease of cross-serial dependencies in Dutch. As an illustration, consider the difference between the following clauses (e.g., Steedman 2000):

… omdat Jan Hans Peter de paarden zag helpen voeren.
... because Jan Hans Peter the horses saw help feed.
... because Jan saw Hans help Peter feed the horses.

The clause *Jan Hans Peter de paarden zag helpen voeren* is an example of cross-serial dependency in Dutch. The English clause *Jan saw Hans help Peter feed the horses* illustrates the binding relations in the Dutch clause. The binding relations are cross-serial because the first "agent" (*Jan*) binds with the first verb (*zag/saw*). The binding relations between the other "agents" and verbs proceed in the same way. Fig. R3a illustrates the bare encoding of the English clause *Jan saw Hans help Peter feed the horses* in the blackboard architecture, using the W_x assemblies of the "phonological" blackboard illustrated in fig. R1. The temporal order in which the words bind to their W_x assemblies is the same as the temporal order in which the W_x assemblies bind to the structure assemblies (NP and VP), and the temporal order in which the structure assemblies bind to each other.

Fig. R3b illustrates the bare encoding of the Dutch structure *Jan Hans Peter de paarden zag helpen voeren*. The structure is the same as the English structure in fig. R3a. The difference is in the temporal order of binding. Because word assemblies now bind to W_x assemblies, VP assemblies can be activated ahead of verbs. So, V_1 and V_2 can be activated in anticipation of a verb. In the case of V_1 its "word" subassembly will be activated, and in the case of V_2 its main assembly. In this way, W5 (*zag*) will bind with V_1, W_6 (*helpen*) can then bind with V_2. After that, W_7 (*voeren*) can bind to V_3, which can bind to the "agent" subassembly of N_3 and to N_4 (with their "theme" subassemblies). Fig. R3b illustrates that this binding sequence is possible in the architecture, although it is of a more elaborate nature than the binding sequence for the equivalent English structure (fig. R3a). This might account for the relative minor occurrence of cross-serial dependencies in languages. The fact that the English and Dutch binding structures in the architecture are similar may relate to the fact that both languages are grammatically related, despite the difference in the surface structure of the clauses involved (e.g., Steedman 2000). The binding process in fig. R3b is easier than the binding process with center-embedding illustrated in fig. 18a.

R8.3. Processing. A number of questions were raised concerning processing in the model. **Powers** argued that we have not shown how words are recognized, including their semantics. Word recognition was not a topic in our article, because it has been a topic in neural network research for many years. Networks like feedforward networks, interactive activation networks (e.g., McClelland & Rumelhart 1981) or ART (e.g., Grossberg 1999) could, in our view, be used for this purpose. Our aim was to discuss how words, once recognized, can be combined in combinatorial structures. Fig. R1 illustrates that the word assemblies interact with the sentence blackboard through the phonological blackboard. As a result, the word assemblies can be as distributed as they need to be. Semantics will derive from the web of relations between the word assemblies and between word assemblies and other structures in the brain, such as the motor cortex or the visual cortex, as illustrated in fig. 27.

One aspect of word recognition is that it will provide information about the lexical nature of a word. In turn, this information will be used in the binding process in the architecture, as we illustrated in fig. 20. This is an aspect of the architecture that needs to be developed much further. Nevertheless, the circuit in fig. 20 illustrates some basic aspects of this process, in particular its incremental and interactive nature. What the circuit in fig. 20 in particular does not show is the stochastic nature of this process. Consider a sentence like *The man who hunts ducks out on weekend* (Pinker 1994). The word *duck* is ambiguous because it can be a noun and a verb. We assume that the neural structure of the word *duck* (e.g., its word assembly) contains both kinds of information, which can be activated in the course of word processing. However, the incremental nature of human processing suggest that a competition between these aspects occurs that results in one interpretation of *duck* (i.e., noun or verb), which is then used in a circuit for binding as illustrated in fig. 20. The garden path nature of this sentence suggest that *duck* is interpreted initially as a noun.

Several factors could influence the competition between ambiguous information related with a word. For example, the content of the sentence already processed, familiarity, or syntactic priming, that is, sentence structures of previously processed sentences (e.g., Pickering & Branigan 1999). Aspects of prosody could also be of influence. For example, the sentence *The man who hunts ducks...out on weekend* would be interpreted differently from the sentence *The man who hunts…ducks out on weekend*, because of the difference in the moment of the pause.

In terms of neural networks, these competition processes could be instantiated in interactive activation networks or attractor neural networks, as suggested by **Grüning & Treves**. In fact, we have already used attractor dynamics in the production of a context-free language with a production system (Van der Velde 1995). This is, of course, a toy model of language production, but it illustrates that attractor networks can be used to implement control-like structures as illustrated in fig. 20. Because the attractor dynamics used in the model was stochastic, the model already shows the noise tolerance, context-sensitivity, and analogue nature of processing referred to by **Durstewitz** and **Grüning & Treves**. It could also result in underspecified representations of a given sentence (Sanford and Sturt 2002).

So, the fact that the rules would operate in a stochastic manner, as noted by **Powers**, is not a real issue. Having statistical rules is not the same as having no rules. The difference is that the application of a rule is not deterministic, but depends on a stochastic process that can be influenced by a number of factors (e.g., context, syntactic priming). This kind of processing fits very well with the abilities of interactive activation networks, or attractor neural networks. The interaction between these networks and the blackboard for sentence structure can account for the resolution of ambiguity, which, as **Baars** noted, will consist of an expectation-driven process that selects the most plausible interpretation, disregarding the multitude of other options.

R8.4. Challenges. In their commentary, **Phillips & Wagers** raise a number of very interesting challenges for the blackboard architecture and suggestions for its further development. This is an illustration of how the interaction between linguistics and neural network modelling can begin to be more productive.

Phillips & Wagers argue that hierarchical constituent structure is missing in the architecture. In the target article we noted that binding in this architecture is not a state but a process (see also R2). The same it true for constituents. Assemblies such as S_i or C_i do not themselves represent constituents, as correctly noted by **Phillips & Wagers**. But they are crucial for the process in which constituents can be retrieved. For example, in fig. 16a *cat that bites the dog* is a hierarchical constituent, because it can be reactivated (retrieved) from S_1 downwards. That is, by opening all "downward" gating circuits in a hierarchical manner, the reactivation process will eventually reach N_2. But it will also stop there, which indicates that a constituent boundary has been reached. This illustrates that hierarchical constituent structure is given by a process in this architecture (i.e., the process that reactivates the constituent in a hierarchical manner). Issues related with constituents, as discussed by **Phillips & Wagers**, should be handled in the same way in the further development of the architecture.

Co-ordination rules, addressed by **Phillips & Wagers** , are an important topic for further development. One suggestion we have been exploring is the introduction of specific structure assemblies for co-ordination (e.g., "and" assemblies). This solution is in line with the suggestion made by **Phillips & Wagers** that NP and VP assemblies should bind recursively. (Furthermore, fig. R1 illustrates that they do not directly bind to words in the present version of the architecture.) A question that we explore is whether these co-ordination assemblies should be general (i.e., used for all co-ordinations) or specific (e.g., specific "and" assemblies for NPs). Co-ordination assemblies could also be used for structures such as *six big red India rubber*, which could be seen as an elliptic version of *six (and) big (and) red (and) India (and) rubber*. The word *and* is dropped, but the "and" assemblies could still be used.

Anaphoric relations are not yet included in the blackboard architecture. An interesting aspect of anaphoric relations is that they can reach beyond the boundaries of a sentence. For example, in *The boy sees ice cream. He wants it.*, the anaphoric relations bind information across sentences. In fig. R1 we illustrated a (quasi) architecture for structure within words. In the target article we discussed an architecture for structure within sentences and clauses. A direct extension would be a specific blackboard architecture for structure beyond sentence boundaries, as given by anaphoric relations.

R9. Final remarks

The purpose of the target article was to show how true combinatorial structures can be instantiated in neural terms. True combinatorial structures depend on constituent structures that can be combined on the fly. Furthermore, information can be retrieved from the combinatorial structures on the basis of their constituents. In this way (and, as we argue, only in this way) can information be processed and retrieved in a manner that is productive and systematic. Neural structure and dynamics seem to prevent constituents to be copied and moved elsewhere. Therefore, we suggest that neural representations of constituents remain "in situ". Combinatorial structures can be instantiated on the basis of these constituents by dynamic binding processes and structures (e.g., structure assemblies) of the kind illustrated in the target article.

However, the success of a particular approach cannot be determined on the basis of a single article. This is true for the architecture we have presented here, as well as for the alternative approaches that have been discussed here or presented elsewhere in the literature. In the end, the success of any approach is determined by the research program it initiates and the progress of that program. We are fully aware that we are just at the beginning of such a program, and that it is far too soon to determine whether it will be successful. The commentators have been very helpful in pointing out and discussing the problems that have to be solved and the difficulties that have to be surmounted. Most likely, may more will be lurking in the background, waiting to pop-up at the right (or wrong) moment. However, we have not yet seen a problem that seems to be unsolvable, or any difficulty that seems to be insurmountable. This raises our confidence in the further development of the architectures we presented here.

R10. References

Amit, D. J. (1989) *Modeling brain function.* Cambridge University Press.

Byron, F. W. & Fuller, R. W. (1992) *Mathematics of classical and quantum physics.* New York: Dover.

Calvin, W. H. (1995) Cortical columns, modules, and Hebbian assemblies. In: *The handbook of brain theory and neural networks,* ed. M.A. Arbib. Cambridge, MA: MIT Press.

Calvin, W. H. & Bickerton, D. (2000) *Lingua ex machina: Reconciling Darwin and Chomsky with the human brain.* Cambridge, MA: MIT Press.

Chapman, B., Stryker, M. P. & Bonhoeffer, T. (1996) Development of orientation preference maps in ferreft primary visual cortex. *Journal of Neuroscience* 16: 6443-6453.
Chomsky, N. (1957) *Syntactic structures.* The Hague: Mouton.
Coren, S., Ward, L. M. & Enns, J. T. (1999) *Sensation and perception*. Orlando FL: Harcourt Brace.
Dennett, D. C. (1991) *Consciousness explained.* London: Allen Lane.
Elman, J. L. (1991) Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning* 7:195-225.
Farah, M. J. (1990) *Visual agnosia*. Cambridge, MA: MIT Press.
Fodor, J. A., & Pylyshyn, Z. W. (1988) Connectionism and cognitive architecture: A critical analysis. *Cognition* 28:3-71.
Fuster, J. M. (1995) *Memory in the cerebral cortex.* Cambridge, MA: MIT Press.
Gibson, E. (1998) Linguistic complexity: Locality of syntactic dependencies. Cognition 68:1-76.
Gonchar, Y. & Burkhalter, A. (1999) Connectivity of GABAergic calretinin-immunoreactive neurons in rat primary visual cortex. *Cerebral Cortex* 9:683-696.
Grossberg, S. (1999) The link between brain learning, attention, and consciousness. *Consciousness & Cognition* 8:1-44.
Hanson, S. J. & Negishi, M. (2002) On the emergence of rules in neural networks. *Neural Computation* 14:2245-2268.
Henderson, J. (1994) Connectionist syntactic parsing using temporal variable binding. *Journal of Psycholinguistic Research* 23: 353-379.
Hornik, K., Stinchcombe, M. & White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* 2:359-366.
Hubel, D. (1995) *Eye, brain, and vision.* New York: Scientific American Library.
Hummel, J. E., Holyoak, K. J. (1997) Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review* 104:427-466.
Hummel, J. E., Holyoak, K. J. (2003) A symbolic-connectionist theory of relational inference and generalization. *Psychological Review* 110:220-264.
Jackendoff, R. (2002) *Foundations of language*. Oxford: Oxford University Press.
Just, M. A. & Carpenter, P. A. (1992) A capacity theory of comprehension: Individual differences in working memory. *Psychological Review* 99:122-149.
McClelland, J. L. & Rumelhart, D. E. (1981) An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review* 88:375-407.
McCloskey, M. & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In: *The Psychology of Learning and Motivation, Vol. 24*, ed. G. H. Bower. New York: Academic Press.
Meng, M. & Bader, M. (2000) Ungrammaticality detection and garden path strength: Evidence for serial parsing. *Language and Cognitive Processes* 15:615-666.
Newell, A., Rosenbloom, P. S. & Laird, J. E. (1989) Symbolic architectures for cognition. In: *Foundations of cognitive science*, ed. M. Posner. Cambridge, MA: MIT Press.
Oram, M. W. Perrett, D. I. (1994) Modeling visual recognition from neurobiological contraints. *Neural Networks* 7:945-972.
O'Reilly, R. C. & Rudy, J. W. (2001) Conjunctive representations in learning and memory: Principles of cortical and hippocampal function. *Psychological Review* 108:311-345.
Pickering, M. J. & Branigan, H. P. (1999) Syntactic priming in language production. *Trends in Cognitive Sciences* 3:136-141.
Pinker, S. (1994) *The language instinct*. London: Penguin.
Pinker, S. (1989) *Learnability and cognition*. Cambridge, MA: MIT Press.

Plate, T. A. (1997) A common framework for distributed representation schemes for compositional structure. In: *Connectionist systems for knowledge representation and deduction* eds. F. Maire, R. Hayward & J. Diederich. Brisbane, Australia: Queensland University of Technology Press.
Plate, T. A. (2003) *Holographic reduced representation: Distributed representation for cognitive structures*. Stanford, CA: CSLI Publications.
Rolls E. T. & Treves, A. (1998) *Neural networks and brain function*. Oxford: Oxford University Press.
Rudin, W. (1976) *Principles of mathematical analysis.* New York: McGraw-Hill.
Sanford, A. J. & Sturt, P. (2002) Depth of processing in language comprehension: not noticing the evidence. *Trends in Cognitive Sciences* 6:382-386.
Singer, W., Gray, C. M. (1995) Visual feature integration and the temporal correlation hypothesis. *Annual Reviews Neuroscience* 18:555-586.
Shastri L. & Ajjanagadde, V. (1993) From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16:417-494.
Steedman, M. (2000) *The syntactic process*. Cambridge MA: MIT Press.
Van der Velde, F. (1995) Symbol-manipulation with Neural Networks: Production of a context-free language using a modifiable working memory. *Connection Science* 7:247-280.
Van der Velde, F. (1999) A spy to spy on a spy: From type to token representation with cell assemblies. *Behavioral and Brain Sciences* 22:306-307.
Van der Velde, F. (2001) The study of brain function: Methodological considerations. In: *Frontiers of life, Vol III, part two: Building blocks for intelligence systems*, eds. D. J. Amit & G. Parisi. San Diego: Academic Press.
Van der Velde, F. (2005) Modelling language development and evolution with the benefit of hindsight. *Connection Science* (in press).
Van der Velde, F. & de Kamps, M. (2001) From knowing what to knowing where: Modeling object-based attention with feedback disinhibition of activation. *Journal of Cognitive Neuroscience* 13:479-491.
Van der Velde, F. & de Kamps, M. (2002) Synchrony in the eye of the beholder: An analysis of the role of neural synchronization in cognitive processes. *Brain and Mind* 3:291-312.
Van Valin, R. D. Jr. (2001) *An introduction to syntax*. Cambridge: Cambridge University Press.
Von der Malsburg, C. (1986) Am I thinking in assemblies? In: *Brain theory*, eds. G. Palm & A Aertsen. Berlin: Springer.
Von der Malsburg, C. (1987) Synaptic plasticity as basis of brain organization. In: *The Neural and Molecular Bases of Learning*, eds. J. P. Changeux and M. Konishi. Chichester: John Wiley.