

A neural blackboard architecture of sentence structure

Frank van der Velde¹

Marc de Kamps²

*¹Cognitive Psychology Unit, Leiden University
Wassenaarseweg 52, 2333 AK Leiden, The Netherlands
Tel: (31) (0)715273637, Fax: (31) (0)715273783
vdvelde@fsw.leidenuniv.nl*

*²Robotics and Embedded Systems
Department of Informatics, Technische Universität München
Boltzmannstr. 3, D-85748 Garching bei München, Germany
kamps@in.tum.de*

Abstract

We present a neural architecture for sentence representation. Sentences are represented in terms of word representations as constituents. A word representation consists of a neural assembly distributed over the brain. Sentence representation does not result from associations between neural word assemblies. Instead, word assemblies are embedded in a neural architecture, in which the structural (thematic) relations between words can be represented. Arbitrary thematic relations between arguments and verbs can be represented. Arguments can consist of nouns and phrases, as in sentences with relative clauses. A number of sentences can be stored simultaneously in this architecture. We simulate how probe questions about thematic relations can be answered. We discuss how differences in sentence complexity, such as the difference between subject-extracted versus object-extracted relative clauses and the difference between right-branching versus center-embedded structures, can be related to the underlying neural dynamics of the model. Finally, we illustrate how memory capacity for sentence representation can be related to the nature of reverberating neural activity, which is used to store information temporarily in this architecture.

Introduction

To understand how the brain enables the mind, processes at the neural level have to be related with processes at the cognitive level. This entails an implementation of cognitive processes in terms of neural computations. Successful implementations of this kind have been produced for processes in visual perception (e.g., Grossberg, 2000), working memory (e.g., Amit & Brunel, 1997; Wang, 2001), and visual attention (e.g., Usher & Niebur, 1996; Itty & Koch, 2001; Van der Velde & de Kamps, 2001). However, neural implementations of language processes have been hard to come by. It is not difficult to see why. On the one hand, linguistic expressions are highly structured and language processes depend on complex and often recursive forms of information processing (e.g., Jackendoff, 1999). On the other hand, a direct animal model of language processing is lacking, which precludes a systematic analysis of language processes at the neural level.

Yet, the overall structure of the cortex is highly uniform (e.g., Calvin, 1995; Mountcastle, 1998), which suggests that forms of neural representation and processing found in perception or attention could play an important role in other cognitive processes as well. This notion can be combined with the detailed knowledge about language representation and processing obtained by linguistics and psycholinguistics over the last decades. Thus, knowledge about language representation and processing can be used as a guiding principle in an implementation of (aspects of) language processing in terms of established forms of neural representation and processing. In this article, we will explore the possibility of a neural implementation of language processing. In particular, we will focus on three fundamental aspects of such an implementation: combinatorial productivity, retrieval of information and performance effects.

First, a neural implementation of language processing should satisfy the combinatorial productivity of language. Words can be combined arbitrarily to form sentences, in such a way that the relations between the words are determined by the syntactic structure of the sentence. For instance, the sentence *The mouse chases the cat* expresses a relation between the words *mouse*, *chases* and *cat*, determined by the syntactic structure of the sentence. In this case it is clear that the mouse initiates an action (chasing), which is directed at the cat. Relations of this kind can be described in terms of the argument structure of a verb, which is determined by the thematic roles that the verb permits or requires. In this example, *mouse* is the agent of the verb *chases* and *cat* is the theme (or patient) of this verb. Thus, in the representation of this sentence, the arguments *mouse* and *cat* have to be related (or bound) correctly to the thematic roles of agent and theme of the verb *chases*. The combinatorial productivity of language entails that a neural implementation of language processing must be able to represent the binding of arbitrary arguments (e.g., nouns and clauses) to the thematic roles of arbitrary verbs. We will describe a model that implements arbitrary verb-argument binding in terms of neural assemblies embedded in a neural architecture.

Second, a neural implementation of language processing must allow the retrieval of information (e.g., the thematic relations) expressed in a sentence. Given that the main purpose of language is to provide information about 'who does what to whom' (e.g., Pinker, 1994; Calvin & Bickerton, 2000), a neural implementation of language processing should be able to produce answers to 'who does what to whom' questions. These probe questions can be called 'binding' questions, because their answers depend on the correct representation of the thematic relations (verb-argument binding) expressed in

a sentence. The ability to reproduce or recognize the thematic relations expressed in a sentence is a crucial aspect of language comprehension. As such it has been used (in a non-verbal manner) as a test for language comprehension in aphasic stroke patients (e.g., Caplan, Baker & Dehaut, 1985; Grodzinsky, 2000). Thus, in case of the sentence *The mouse chases the cat*, it should be possible to retrieve information that answers questions like "Who chases the cat" or "Whom does the mouse chase?". We will discuss and simulate how information related to verb-argument binding can be retrieved in the model presented here.

Third, a neural implementation of language processing should account for the performance effects observed in human sentence processing. We will discuss performance effects related with sentence complexity in terms of the structure and dynamics of the model presented here. In particular, we will discuss the difference between subject-extracted relative clauses (*The mouse that sees the dog chases the cat*) versus object-extracted relative clauses (*The mouse that the dog sees chases the cat*), and the difference between right-branching and center-embedded structures. Finally, we will discuss memory capacity for sentence representation in terms of the dynamics of the model presented here.

Representation and architecture

The model presented here is based on the assumption that information in the brain is represented by means of neural cell assemblies, as proposed by Hebb (1949). A neural assembly consists of an interconnected group of neurons, which is generally distributed over the brain. In the case of language, Hebb's proposal suggests that words are represented by means of neural assemblies (or word assemblies, for short). Evidence for the existence of word assemblies is presented by Pulvermüller (1999, 2001). One example concerns the difference in neurophysiological responses (ERP and MEG) generated by action verbs versus visually related nouns. In terms of these measures, a difference in activation between fronto-central action-related areas (resulting from action verbs) and occipital visual areas (resulting from visually related nouns) was found. Differences in brain activation were also found between action-related nouns and visually related nouns, and between action verbs related with leg actions ('walking') versus action verbs related with face actions ('talking'). Furthermore, an fMRI study showed a difference in location of activation between arm-related action verbs and leg-related action verbs, in line with the difference in location of activation found with arm movements versus leg movements. On the basis of such evidence, Pulvermüller (1999, 2001) argued that word representations consist of neural assemblies, distributed over different parts in the brain. The word assemblies will develop as a result of associations with representations (such as action representations or visual object representations) that constitute the referential meaning of the words, as illustrated in figure 1.

Figure 2 (left) illustrates the word assemblies for *chases*, *mouse* and *cat* that would be activated with the sentence *The mouse chases the cat*. Figure 2 (right) illustrates that the same word assemblies would be activated with the sentence *The cat chases the mouse*. The fact that two different sentences can result in the activation of the same word assemblies raises the question of how sentences are represented in the brain. A sentence representation would have to consist of a form of binding between the arguments (e.g., nouns) and the verb in a manner that satisfies the structure of the sentence (e.g., the

thematic relations expressed in the sentence). However, the binding of arguments to a verb cannot consist of (temporary) associations between word assemblies. For instance, as illustrated in figure 2, associations between *chases*, *mouse* and *cat* do not distinguish between *The mouse chases the cat* and *The cat chases the mouse*, because these word assemblies are active in each of these two sentences.

To further illustrate the issues involved, consider the sentence *The mouse that sees the dog chases the cat*. In the sentence *The mouse chases the cat*, the agent argument of *chases* is *the mouse*, but in this sentence the whole phrase *the mouse that sees the dog* is the agent of the verb. In linguistic terms this entails that a representation of *the mouse that sees the dog* is copied into the open (agent) argument slot of the verb *chases* (Pinker, 1989). The fact that representations can be copied in linguistic expressions is also clear in a simultaneous representation of the sentences *The mouse chases the cat* and *The cat chases the mouse*, which consists of two copies of the verb *chases*, each with different arguments (given by copies of *mouse* and *cat*) in the argument slots. Copying representations is a natural operation in digital computers, but it is questionable whether this occurs in the brain. Instead, if words are represented in the brain by means of neural assemblies distributed over different parts of the brain, as illustrated in figure 1, it is difficult to see how such an assembly could be copied and represented elsewhere. Furthermore, an attempt to copy a part of an assembly would disrupt its connection structure. For instance, if the lexical entry of a word is represented by a part of the overall assembly, then the associations in the overall assembly would be broken when that part of the word assembly is copied and represented elsewhere. In this way, (part of) the meaning of the word would be lost in the copied assembly. For these reasons, the word assemblies in the model presented here are not copied. Instead, the word assemblies are embedded in a neural architecture in which they are bound temporarily in a manner that preserves the relations between the words expressed in the sentence.

Association versus structural representation

As discussed above, the structural relations between the words in a sentence cannot be represented with direct associations between word assemblies, as illustrated in figure 2. Therefore, in the model presented here, word assemblies are embedded in a neural architecture in which structural relations can be formed between the word assemblies. Information that is sensitive to the structural relations between the words in a sentence can be represented and retrieved in this way. The neural architecture is implemented by means of ‘structure’ assemblies that interact with the word assemblies. The structure assemblies provide the possibility to represent different tokens of the same word assembly, and they are used to represent elements of syntactic structures. For instance, there are structure assemblies used in the representation of syntactic structures such as Noun Phrases (NPs) and Verb Phrases (VPs).

Figure 3 presents the representation of the sentence *The mouse chases the cat* in the architecture discussed here. The sentence is represented by means of assemblies that represent words (word assemblies, see figure 2), assemblies that are used to represent the structure of the sentence (structure assemblies), gating circuits that are used to control the process of sentence representation, and memory circuits that are used to bind different word and structure assemblies into a (temporal) representation of the overall sentence.

The figure illustrates how the word assemblies for *mouse*, *chases* and *cat* are bound to different structure assemblies, which in turn are bound to represent the overall sentence.

The structure assemblies possess an internal structure, composed of a main assembly (N_i for NP assemblies and V_i for VP assemblies) and an unspecified number of subassemblies. Figure 3 shows the subassemblies for the thematic roles of agent (a) and theme (t). The subassemblies are connected to the main assembly by gating circuits, which can be activated when certain structural control conditions are met. During syntactic processing, word and structure assemblies are bound to one another by activating memory circuits that connect the assemblies. The intermediary binding to VP and NP assemblies is necessary to avoid the ‘binding problems’ that often occur in forms of neural representation (Van der Velde, 2001). Assemblies like VPs and NPs also play an important role in the representation of the structural relations expressed in the sentence. That is, they can bind word assemblies in a manner that preserves the relations between the words in the sentence. Before describing this architecture further, we will first describe the gating and memory circuits.

Gating and Memory Circuits

Figure 4 illustrates the gating circuit. The overall circuit is in fact a combination of two gating circuits, one for each direction. Each gating circuit is a disinhibition circuit that controls the flow of activation between two assemblies (X and Y in figure 4) by means of an external control signal. Disinhibition circuits have been found in the visual cortex (Gonchar & Burkhalter, 1999), and they have been used to model object-based attention in the visual cortex (Van der Velde, 1997; Van der Velde & de Kamps, 2001). The gating circuit that controls the flow of activation from X to Y operates in the following manner. If the assembly X is active, it activates an inhibition neuron (or group of neurons) i_x , which inhibits the flow of activation from X to X_{out} . When i_x is inhibited by another inhibition neuron (I_x) that is activated by an external ‘control signal’, X activates X_{out} . In turn, X_{out} activates Y . The gating circuit from Y to X operates in a similar manner. In figure 3, the combination of both gating circuits between X and Y is represented with one symbol, also illustrated in figure 4. Notice, however, that the flow of activation in each gating circuit can be controlled with a separate control signal.

The memory circuit is presented in figure 5 (left). It also consists of two gating circuits that control the flow of activation from X to Y and vice versa, as in figure 4. In this case, however, the control signal in both gating circuits results from a ‘delay assembly’. The delay assembly is activated when X and Y are active simultaneously (figure 5, right). The delay assembly then remains active due to the reverberating activity in this assembly. Reverberating activity in the cortex has been found with memory tasks, such as delayed response tasks in which a response can only be given after a waiting period (e.g., Fuster, 1973). The reverberating activity retains the response related information during the memory period. Thus, reverberating activity constitutes a form of working memory (e.g., Amit, 1995, Wang, 2001). Here, the delay activity in a memory circuit constitutes a memory of the fact that the two assemblies connected by the circuit have been simultaneously active at a certain time, e.g., in the course of syntactic processing. When the memory circuit is active, it allows activation to flow between the assemblies it connects. In this way, the memory circuit produces a binding between these

assemblies. As a result, the memory (gating) circuit can be in two different states, inactive and active, as illustrated with the symbols presented in figure 5.

Overview of the Architecture

Figure 6 presents an overview of a neural architecture for sentence representation (in particular for verb-argument binding). Each assembly that represents a noun is connected to the main assembly of each NP assembly by means of a memory circuit, which is initially inactive. In the same manner, each assembly that represents a verb is connected to the main assembly of each VP assembly by means of an (initially inactive) memory circuit. The main assembly of each NP or VP assembly is connected to an (unspecified) number of subassemblies by means of gating circuits (i.e., each NP or VP assembly has its own set of subassemblies, as illustrated with V_1 in figure 3). Main assemblies are also delay assemblies, in the sense that they can remain active on their own. Subassemblies are used to represent thematic roles, such as agent or theme, as shown in figure 6. They can also be used to represent syntactic structures such as complements or relative clauses (as discussed later on). Subassemblies can be used to represent thematic roles or syntactic structures, because they are used to connect the NP and VP assemblies. Thus, all agent subassemblies of the NP assemblies are connected to all agent subassemblies of the VP assemblies, by means of (initially inactive) memory circuits. Likewise for the other kinds of subassemblies.

There is also an interaction between the VP assemblies, as illustrated in Figure 7. The VP assemblies activate a population of inhibitory neurons, which in turn inhibits each of the VP main assemblies. In this way, the VP assemblies mutually interact in an inhibitory manner, which results in a competition between the VP assemblies, as indicated in figure 6. However, the population of inhibitory neurons itself can also be inhibited. This provides a dynamic control over the competition between the VP assemblies. The ability to retrieve information from this architecture critically depends on this competition and the possibility to control it.

Figure 3 shows the memory circuits that are active in the representation of the sentence *The mouse chases the cat*. It is assumed that, when a sentence is processed, one of the NP assemblies is activated whenever a word assembly representing a noun is activated. It is arbitrary which NP assembly is activated, provided it is 'free', that is, not already bound to a noun. The distinction between free and 'bound' NP assemblies can be made in terms of the activity in the memory circuits connected to the bound NP assemblies. On the basis of this activity, the activation of the bound NP assemblies can be suppressed during the processing of a sentence (a form of 'inhibition of return' between structure assemblies). The active NP assembly will remain active until a new NP assembly is activated by the occurrence of a new noun in the sentence. (E.g., the occurrence of a new noun could result in the inhibition of the active NP assembly before a new NP assembly is generated.) The selection of a VP assembly proceeds in the same manner.

Thus when the assembly for *mouse* is activated, a NP assembly is activated as well. As a result, the assembly for *mouse* is bound to the main assembly of this NP assembly, because the memory circuit between these assemblies is activated (see figure 5). In the same manner, the assembly for *chases* is bound to a VP assembly. To achieve the binding of *mouse* and *chases*, a binding has to occur between the NP and VP

assemblies to which *mouse* and *chases* are bound. Figure 6 shows that a binding between NP and VP assemblies can only occur by means of the subassemblies of the same kind. In this case, a binding should occur between the agent subassembly of the NP assembly for *mouse* and the agent subassembly of the VP assembly for *chases* (figure 3). This binding does indeed occur because the gating circuits between the main assemblies of the structure assemblies and their agent subassemblies are activated in a selective manner by neural control circuits. For instance, a neural control circuit can identify the noun as the agent of the verb in the sequence noun-verb (given by *mouse chases*). It can then produce a control signal that activates the gating circuits for the agent subassemblies. This will result in the activation of the agent subassemblies that belong to the NP assembly for *mouse* and the VP assembly for *chases*, because they are the only NP and VP assemblies that are active at that moment. As a consequence, these assemblies will be bound in the manner illustrated in figure 5. The binding of *chases* and *cat* proceeds in a similar manner.

Multiple instantiation and compositional representation

Figure 8 shows the simultaneous representation of the sentences *The mouse chases the cat*, *The cat chases the mouse* and *The mouse sees the dog* in the architecture presented in figure 6. The neural assembly representation of *The mouse chases the cat* in figure 8 is the same as in figure 3. However, sentence presentation is simplified in figure 8. In particular, the gating and memory circuits are omitted in figure 8 (but they are implied). Thus, *mouse* is still connected to N_1 by means of an active memory circuit (likewise for the other word assemblies). Furthermore, a subassembly in figure 8 now represents the two corresponding subassemblies of a NP and a VP assembly and the active memory circuit that connects them (as in figure 3).

The words *mouse* and *chases* occur in more than one sentence in figure 8, and, in the case of *mouse*, in more than one thematic role. This creates the problem of the multiple instantiation of the representations for *mouse* and *chases*. Multiple instantiation of representations is a difficult problem for neural or connectionist systems (e.g., Sougné, 1998). Figure 8 illustrates how the problem of multiple instantiation is solved in the architecture presented in figure 6. Each word in a sentence is represented by binding its word assembly to a unique structure assembly. For instance, the word assembly for *mouse* is bound the NP assemblies N_1 , N_4 and N_5 in figure 8. These different NP assemblies represent *mouse* in the different sentences involved. In this way, *mouse* can be represented as agent in one sentence (by N_1 or N_5) and as theme in another (by N_4). Thus, the different NP assemblies represent *mouse* as different tokens of the same type. Similarly, the different VP assemblies (V_1 and V_2) represent *chases* as different tokens of the same type. Token representation is important for the generation of a compositional form of representation (e.g., Fodor & Pylyshyn, 1988). In turn, a compositional form of representation is important to provide for the productivity of language, as illustrated in Figure 8.

As noted above, the sentences presented in figure 8 cannot be represented in terms of direct associations between the word (noun and verb) assemblies. For instance, the association of *mouse-chases-cat* does not distinguish between the sentences *The mouse chases the cat* and *The cat chases the mouse*, because *mouse* and *cat* are not represented as agent or theme in these associations. Even with separate representations for *noun-*

agent and *noun-theme* (e.g., *mouse-agent* and *mouse-theme*) confusions would arise if sentences were represented in terms of direct associations between these representations. For instance, in the simultaneous representation of *The mouse chases the cat* and *The cat chases the mouse*, the verb *chases* would be associated with *mouse-agent*, *cat-theme*, *cat-agent* and *mouse-theme*. But the same associations would be formed with the sentences *The mouse chases the mouse* and *The cat chases the cat*. In contrast, in the architecture illustrated in figure 6, the sentences in figure 8 can be represented using the representations for *mouse*, *cat*, *dog*, *chases* and *sees* as constituent representations. In this case, the sentences *The mouse chases the cat* and *The cat chases the mouse* can be distinguished because they are represented with different NP and VP assemblies. As a result, *mouse-N₁* and *cat-N₂* are the agent and theme of *chases-V₁*, whereas *cat-N₃* and *mouse-N₄* are the agent and theme of *chases-V₂*.

The internal structure of the NP and VP assemblies, given by the gating circuits, is of crucial importance in this respect. Without this internal structure, the representations presented in figure 8 would also consist of direct associations between neural assemblies, which would create the same problems as described above, such as the failure to distinguish between *The mouse chases the cat* and *The cat chases the mouse*. With the control of activation provided by gating circuits, the representations of these two sentences can be selectively (re)activated. We will illustrate this in the next section. In particular, we will investigate how information can be retrieved (i.e., answers to binding questions can be produced) in the architecture presented in figure 6, even with multiple instantiation of representations as illustrated in figure 8.

Retrieving information from the architecture

We will illustrate the ability to retrieve information from this architecture by analyzing and simulating the production of the answer to the question “Whom does the mouse chase?”, when the sentences presented in figure 8 are stored simultaneously. The assemblies were simulated as populations of spiking neurons, in terms of the average firing rate of the neurons in the population. Details of the dynamical equations are given in the Appendix. The simulations are illustrated in the figures 9 and 10. Figure 9 shows the activation of the word assemblies *mouse*, *chases* and *cat*, and the subassemblies for N₁-agent and V₁-theme. Figure 10 shows the activation of the NP main assemblies (left) and the VP main assemblies (right) used in the sentence representations in figure 8. Figure 10 (right) also shows two ‘free’ VP main assemblies (V4 and V5), to compare the activation of free assemblies with bound assemblies in this process. The vertical lines in the figures are used to compare the timing of events. The simulations start at t = 0 ms. Before that time, the only active assemblies are the delay assemblies in the memory circuits.

The question “Whom does the mouse chase?” provides information that *mouse* is the agent of *chases* and it asks for the theme of the sentence *mouse chases x*. The production of the answer consists of the selective activation of the word assembly for *cat* (figure 8). Backtracking, one can see (figures 3 and 8) that this requires the selective activation of the main assembly N₂, the theme subassemblies for N₂ and V₁, and the main assembly V₁ (in reversed order). This process proceeds as follows. First, we assume that the question temporarily activates the representations for *mouse* and *chases* and produces the control signal that activates the gating circuits for the agent subassemblies of the NP

assemblies. Figure 9 shows the activation of the assemblies for *mouse* and *chases* (beginning at $t = 0$ ms). To produce the selective activation of the word assembly for *cat* later on, other word assemblies cannot be active at that moment. Therefore, it is assumed that the word assemblies are inhibited after a certain time, and remain inhibited until *cat* is to be activated. The horizontal bar in figure 9 indicates the time interval in which the word assemblies (*mouse* and *chases*) are active. The end of the interval (at $t = 400$ ms) is marked by a vertical line.

As indicated in figure 8, the activation of *mouse* will result in the activation of the NP assemblies N_1 , N_4 , and N_5 , and the activation of *chases* will result in the activation of the VP assemblies V_1 and V_2 . Figure 10 shows that these assemblies are indeed activated as a result of the activation of *mouse* and *chases* in figure 9. As indicated with the vertical line in figure 10, the NP main assemblies N_1 , N_4 , and N_5 remain active when *mouse* is inhibited. This results from the reverberating ('delay') properties of main assemblies (see the Appendix for details). As long as V_1 and V_2 are both active, the question "Whom does the mouse chase?" cannot be answered. To produce the answer, the gating circuits for the theme VP subassemblies have to be activated, because the question asks for the theme of *mouse chases x*. However, when both V_1 and V_2 are active, this will result in the activation of the theme subassemblies for V_1 and V_2 , and, in turn, of *cat* and *mouse* (via N_2 and N_4). To prevent this, a competition between V_1 and V_2 has to occur, with V_1 as the winner.

The competition process between the VP assemblies proceeds as follows. Figure 7 shows that VP assemblies are connected to a population of inhibitory neurons. When this population is not inhibited (via 'dynamic control') it sends inhibitory activation to the VP assemblies. In figure 10 (right) the horizontal bar indicates the time interval in which the competition occurs (i.e., in which the inhibition population in figure 7 is not inhibited by dynamic control). The competition starts at $t = 0$ ms, thus at the moment when *chases* is activated (figure 9). In comparison with the NP assemblies activated by *mouse* (figure 10 left), the activity of V_1 and V_2 , initiated by *chases*, is reduced due to the competition between the VP assemblies. The competition can be decided by activating the gating circuits for the agent subassemblies (in the direction from NP to VP). The activation of the gating circuits for the agent subassemblies results in the activation of the agent subassemblies for N_1 , N_4 and N_5 , because they are the active NP assemblies (figure 10, left). The activation of the N_1 agent subassembly is illustrated in figure 9. The horizontal bar here indicates the time interval in which the gating circuits are activated (from $t = 150$ ms to $t = 400$ ms). The beginning of this interval is indicated by the asterisk in figure 10 (right). The active agent subassemblies N_1 and N_5 are bound to the VP assemblies V_1 and V_3 respectively (see figure 8). Thus, the VP assemblies V_1 and V_3 receive activation from the active NP assemblies when the 'agent' gating circuits are activated. (The agent subassembly of N_4 is not bound to a VP assembly, because N_4 is bound to a VP assembly with its theme subassembly, see figure 8). As a result, V_1 wins the competition between the VP assemblies, because V_1 receives activation from *chases* and N_1 , whereas V_2 only receives activation from *chases*, and V_3 only receives activation from N_5 . Figure 10 (right) shows that V_1 is the only active VP assembly after this competition process. The activation of V_2 and V_3 is reduced to the level of the 'free' assemblies V_4 and V_5 . When the competition has ended, the inhibition from the inhibitory population (figure 7) is not effective anymore (it can only result in a reduction of the activity of V_1). Therefore, this

inhibition is ended by means of 'dynamic control' (figure 7), as indicated by the horizontal bar in figure 10 (right).

When V_1 remains as the only active VP assembly, the answer *cat* can be produced by activating the theme subassemblies in the direction from VP to NP. This will produce the selective activation of N_2 , which is the NP assembly bound to *cat* in figure 8, provided that the active NP main assemblies (N_1 , N_4 and N_5 in figure 10) are inhibited first. The horizontal bar in figure 10 (left) illustrates the time interval of this inhibition (from $t = 600$ ms to $t = 650$ ms). After the inhibition of the active NP assemblies, the theme subassemblies in the direction from VP to NP can be activated. The horizontal bar in figure 9 (V1-theme) illustrates the time interval in which the gating circuits for the theme subassemblies are activated (from $t = 700$ ms to $t = 800$ ms). The onset of this event is also illustrated by the dashed vertical line in figures 9 and 10. Figure 9 shows that, as a result, the theme subassembly of V_1 is activated. Figure 10 (left) shows that N_2 is now selectively activated as well. As a result, the word assembly for *cat* can be activated. Thus, the answer to the question "Whom does the mouse chase?" is produced because the information given in the question was used to bias the competition between the VP assemblies. V_1 wins the competition between the VP assemblies, because V_1 was bound to *mouse* (via N_1) during the processing of *The mouse chases the cat*.

The effect of event timing

The competition between the VP assemblies, illustrated in figure 10, produced V_1 as the only active VP assembly. However, the process described above shows that the relative timing of the events that determine the competition process is very important. We will illustrate this in more detail with the relative timing between the inhibition of the word assemblies (like *chases*) and the ending of the competition process, initiated by the 'dynamic control' in figure 7. For example, if the word assembly for *chases* is still active when the competition between the VP assemblies has ended, the assembly V_2 will be reactivated, because it is (also) bound to *chases* (figure 8). Thus, the selective activation of V_1 , needed to produce the answer *cat*, depends on the fact that the assembly for *chases* is inhibited before the end of the VP competition, as indicated by the vertical (solid) line in figure 10 (right).

However, even when the competition between the VP assemblies ends after the inhibition of *chases*, there is still a possibility for interference, as illustrated in figure 11. Figure 11 (right) shows what happens if the competition between the VP assemblies is ended too soon after the inhibition of the word assemblies. Initially, the competition between the VP assemblies has resulted in the selective activation of V_1 , as in figure 10 (right). But when the competition ends, V_2 and V_3 are reactivated. This results from the gradual decay of the word assembly for *chases* and the 'delay' properties of the VP main assemblies. A 'delay' population can maintain an elevated activation without external activation, due to the reverberating activity within the population. The elevated activation of a delay population is in fact an 'attractor' state (Amit, 1989). This means that the population can reproduce the elevated activation when a fluctuation in activation has occurred (as long as the fluctuation remain within the 'attractor' limits). Thus, when the activation of a delay population is reduced due to inhibition, it will reproduce the elevated activation when the inhibition stops, provided the level of activation of the population is still within the 'attractor' limits. This is what happens with the V_2 and V_3

assemblies in figure 11 (right). V_2 was activated by *chases* and V_3 was activated by N_5 (through the activation of the agent subassemblies described above). Due to the competition between the VP assemblies, the activation in V_2 and V_3 is reduced, but when the competition ends, V_2 and V_3 are still active within their ‘attractor’ limits. As a result, the elevated activation in V_2 and V_3 recovers after the end of the competition between the VP assemblies (see the Appendix).

The consequence of renewed activation of V_2 and V_3 is illustrated in figure 11 (left), which shows the activation of the NP assemblies in this case. After the inhibition of the NP assemblies, as in figure 10 (left), and the activation of the theme subassemblies (illustrated in figure 9, for V1-theme), the NP assemblies N_2 , N_4 , and N_6 are now activated, because they are connected by means of theme subassemblies to the VP assemblies V_1 , V_2 and V_3 respectively (see figure 8). In turn, this results in the incorrect activation of *cat*, *mouse* and *dog* as the answer to the question “Whom does the mouse chase?”.

Structural and dynamic control

The process of answering the question “Whom does the mouse chase?” described above was regulated by two forms of control: structural and dynamic. An example of structural control consists of the activation of the gating circuits for the agent subassemblies by which the competition between the VP assemblies is decided. This is a form of structural control because it depends on the structural information, given by the question, that *mouse* is the agent of *chases*. Likewise, the question asks for the theme of the relation *mouse chases x*, which results in the activation of the gating circuits for the theme subassemblies after the competition between the VP assemblies has ended. Dynamic control is found in the inhibition of the word assemblies and NP assemblies, which is needed to produce the activation of the correct NP assembly and word assembly to answer the question. This form of control does not depend on specific information provided by the question, but it is needed to regulate the dynamics of the neural assemblies in the production of the answer, as illustrated in figure 10. Likewise, the event timing discussed above is a form of dynamic control, as illustrated in figure 11. Dynamic control in this model in effect resembles motor control, which also depends on a sequential pattern of activation and inhibition of neurons and neural populations.

Structural and dynamic forms of control are also needed to regulate the process of binding word assemblies into the representation of a sentence, as illustrated in figures 3 and 6. Structural control is needed, for example, for a correct binding between *mouse*, *chases* and *cat* in the sentence representation illustrated in figure 3. To achieve this binding, *mouse* has to be interpreted as the agent and *cat* as the theme of *chases* in the sentence *The mouse chases the cat*. In this way, the gating circuits for the agent subassemblies can be activated so that *mouse* is bound as the agent of *chases* (i.e., N_1 and V_1 are bound by their agent subassemblies). Likewise, the gating circuits for the theme subassemblies have to be activated to bind *cat* as the theme of *chases* (i.e., binding V_1 and N_2 by their theme subassemblies). Again, dynamic control is needed to regulate the dynamics of this binding process. For example, to achieve binding between a VP and a NP assembly, both assemblies have to be active simultaneously, to allow the selective activation of their corresponding subassemblies (e.g., those for theme), selected by the activation of the corresponding gating circuit. This process will be disrupted if, for

instance, two NP assemblies are active at the same moment, because this will result in the binding of two NPs as the theme of a verb. Thus, when *cat* is bound as the theme of *chases* (figure 3), N_2 has to be active and N_1 has to be inhibited.

The combination of structural and dynamic control is a direct consequence of the fact that language processing in the brain depends on both linguistic and neurodynamic constraints. The linguistic constraints result from the linguistic structure of language. The dynamic constraints result from the neural dynamics in the underlying neural structures that produce language processing. The importance of structural and dynamic control raises the question of how these forms of control are implemented in the brain. At this point, we can only describe some general features of how this might occur. We assume that control of the binding process in the architecture presented here will result from 'neural control circuits' that represent particular conjunctions of features. When activated, these control circuits will in turn activate gating circuits or initiate the activation or inhibition of assemblies (e.g., the structure assemblies). For instance, a control circuit could activate the gating circuits for the agent subassemblies in figure 3, because it detected the conjunction *noun-verb* in the sentence *The mouse chases the cat*, and interpreted this conjunction in terms of the noun as the agent of the verb. Likewise, a control circuit could activate the theme subassemblies for *cat* and *chases*, after the detection of the conjunction *noun-verb-noun* in the sentence. These control circuits would thus form (partial) representations of abstract (syntactic) rules. Neurons that represent abstract rules (conjunctions) have been found in the (monkey) prefrontal cortex (Miller, 2000).

It is not difficult to implement a neural circuit that detects a specific conjunction like *noun-verb-noun* and activates agent and theme subassemblies. However, it is unlikely that there will be neural circuits that form conjunctive representations for each of the specific sentence types that can occur in language. It is more likely that neural control circuits will represent (and detect) specific 'local' conjunctions of syntactic features in sentences. For instance, the neural assembly for the verb *chase* could be associated with a neural circuit that represents the fact that the verb requires an agent and a theme, as illustrated in figure 3. Each verb could be associated with a neural circuit that specifies the arguments or thematic relations that the verb requires or allows in a given sentence.

Arguments can be described on different levels of abstraction (Van Valin, 2001). On the lowest level, one can have arguments like *giver*, *runner* and *speaker* and the like. On a more abstract level, one can have arguments like agent, experiencer, recipient, theme or patient. However, these arguments can be described in terms of the 'semantic macro roles' of actor (e.g., agent, experiencer, recipient) and undergoer (e.g., experiencer, recipient, theme, patient). The argument labels (agent, theme) that we have used should be understood as arguments on this level. We have simply used these labels because they are more familiar than actor and undergoer (or X and Y, cf., Pinker, 1989). Thus, in linguistic terms a verb (i.e., its lexical entry) is associated with (at least) one argument structure that specifies the arguments that the verb will have in a given syntactic context. In terms of the model presented here, such an argument structure would be implemented in a neural circuit that controls the binding process illustrated in figures 3 and 8. Figure 12 illustrates two examples in which a verb is associated with only one argument. Thus, *The cat eats* could be interpreted in terms of *cat* as the agent of *eats*, which activates the gating circuits for the agent subassemblies. In contrast, *The glass breaks* could be

interpreted in terms of *glass* as the theme of *breaks*, which activates the gating circuits for the theme subassemblies.

A neural control circuit associated with a verb is an example of a 'lexical frame'. In general, a lexical frame is the syntactic information that is associated with the lexical entry of a word. Lexical frames play an important role in modern theories of grammar (e.g., Pinker, 1989; Weibelhuth, 1995; Jackendoff, 1999; Sag & Wasow, 1999). Evidence for a relation between grammatical and lexical processing is found in studies of language performance (e.g., MacDonald, Perlmutter & Seidenberg, 1994; Bates & Goodman, 1997) and functional neuroimaging (Keller, Carpenter & Just, 2001).

A parsing model that is based on lexical frames is the Unification Space (U-space) model of Vosse and Kempen (2000). The U-space model is a hybrid model, based on both symbolic and dynamic principles. The symbolic part consists of a lexicalist grammar in which syntactic information is represented by lexical frames. Each word in the lexicon is connected to a small structure (frame) of nodes that specifies the nature of the word and the 'syntactic environment' that the word can have in a sentence. The U-space model uses these frames to build a structural representation of a sentence. When a new word in a sentence is processed, the lexical frame of that word will be retrieved from the lexicon. This lexical frame is then copied into a 'unification space'. When more lexical frames enter the unification space, a process of unification starts in which lexical frames are unified by establishing a connection between corresponding nodes. The unification process consists of a dynamic competition between the lexical frames which continues until all lexical frames in the unification space are unified. Various phenomena found in human language processing can be simulated adequately with this model.

In line with this model, we assume that each word is associated with a lexical frame, in the form of a neural control circuit that represents the 'syntactic environment' in which the word can occur. The representation of a sentence will result from an interaction between these circuits and the architecture for sentence representation illustrated in figure 6. In the next section we will describe in general terms how such an interaction could result in the representation of more complex sentences, and how this interaction can result in performance effects related to sentence complexity.

Representation and complexity

Figure 13 shows the representation of the sentence *The mouse that sees the dog chases the cat* in terms of the architecture illustrated in figure 6. The phrase *the mouse that sees the dog*, which contains a (subject-extracted) relative clause, is the agent of the verb *chases* in this sentence. Two extensions of the architecture presented in figure 6 have to be introduced to represent a sentence like this one. The first extension is the introduction of a new subassembly connected to each NP and VP assembly. This subassembly is labeled as a 'relative clause' (rc) subassembly, because it is used to represent a relative clause, as illustrated in figure 13. Thus, when the conjunction *noun-that(comp)-verb* is detected, the gating circuits for the rc subassemblies can be activated, which binds the active NP and VP assemblies (N_1 and V_1 in figure 13) by means of their rc subassemblies. The rc subassemblies provide a site to bind *sees dog* to the NP assembly for *mouse*, which allows the production of answers to specific questions like "Which mouse chases the cat?". This would not be possible with the agent or theme subassemblies. Instead, they can be used to bind *mouse* to the main clause of the

sentence. The binding of *mouse* to the rc subassembly of V_1 provides information that *mouse* is the (extracted) subject of *sees* (via *that*). The next noun (*dog*) can then be bound as the theme of this verb.

The next verb (*chases*) can be interpreted as the verb of the main clause (e.g., due to the conjunction *verb-noun-verb*), which has to be bound to *mouse*, with *mouse* as the agent. However, the dynamic constraint that only one NP assembly can be active at the same time presents a difficulty. The active NP assembly at this moment is N_2 for *dog*, which is bound to V_1 by the theme subassemblies. But *dog* is not the agent of the main clause. To bind *mouse* as the agent of the main clause, N_2 has to be inhibited and N_1 has to be reactivated. To allow the reactivation of N_1 , this assembly was bound to an assembly S, connected to all the NP assemblies (by initially inactive memory circuits). The assembly S belongs to the control circuits, and is used to identify the ‘external argument’ of the verb of the main clause (*mouse* in this case) during sentence processing. Due to this binding, N_1 can be reactivated after the binding of *sees dog* as the relative clause. In this way, *mouse* can be bound as the agent of *chases*, and *cat* as its theme, just as in the sentence *The mouse chases the cat* illustrated in figure 3.

Figure 14 illustrates the representation of the sentence *The mouse that the dog sees chases the cat* in terms of the architecture illustrated in figure 6. In this sentence the object-extracted relative clause *the mouse that the dog sees* is the agent of the verb of the main clause. In terms of the architecture presented in figure 6, the object-extracted relative clause in the sentence illustrated in figure 14 imposes a difficulty that results from the sequence *noun-comp-noun* (*mouse that dog*) in this sentence. The sequence *noun-comp-noun* results in the activation of two NP assemblies, N_1 and N_2 , which both have to be bound to the VP assembly (V_1) of the first verb (*sees*) that follows after the two nouns. However, N_1 and N_2 cannot be simultaneously active. If they were, they would bind in the same manner to V_1 , because the activation of the gating circuits (e.g., for the rc or agent subassemblies) operates for all active NP assemblies.

The difficulty can be resolved by introducing a new kind of structure assembly, labeled T_1 in figure 14. The T_i assemblies are structure assemblies like the NP and VP assemblies, but they do not bind directly to word assemblies. Instead, they only bind to NP and VP assemblies by means of corresponding subassemblies. In linguistic terms, a T assembly acts like a trace (e.g., Caplan, 1995), in the sense that it replaces a NP assembly at an extracted site. Because T assemblies are different from NP assemblies, the gating circuits for the T assemblies can be controlled separately from the gating circuits of the NP assemblies. Thus, in figure 14, N_1 is first bound to T_1 , by means of their rc subassemblies, before it is inhibited. This process also requires a form of dynamic control because N_2 can only be activated, and *dog* can only be bound to N_2 , after this process has been completed. Then N_2 can be bound as the agent to V_1 and T_1 can be bound to V_1 as the theme of this verb. After that, the process of representing the sentence proceeds in the same manner as with the sentence presented in figure 13.

It is clear that the representation of the sentence with the object-extracted relative clause illustrated in figure 14 is dynamically more complex than the representation of the sentence with the subject-extracted relative clause illustrated in figure 13. The increased complexity in representing sentences with object-extracted relative clauses is in line with performance measures on complexity. Sentences with object-extracted relative clauses are more complex than sentences with subject-extracted relative clauses, which follows

from measures like online lexical decision, reading times and response accuracy to probe questions (e.g., see Gibson, 1998). Imaging research has shown an increase in blood flow in language areas with object-extracted relative clauses versus subject-extracted relative clauses (Stromswold, Caplan, Alpert & Rauch, 1996; Just, Carpenter, Keller, Eddy & Thulborn, 1996). In terms of the architecture discussed here, this would result from the additional activation of the T₁ assembly (and the related gating circuits) in figure 14. Furthermore, aphasics have great difficulty in comprehending sentences with object-extracted relative clauses, even when they can comprehend sentences with subject-extracted relative clauses (Caramazza & Zurif, 1976; Caplan & Futter, 1986; Hickok, Zurif & Canseco-Gonzalez, 1993; Grodzinsky & Finkel, 1998). In terms of the architecture discussed here, this could result from the inability to activate and bind T assemblies (Grodzinsky, 2000).

Recursive productivity and complexity

Further effects of sentence complexity on comprehension result from the recursive occurrence of syntactic structures in sentences. Two examples are illustrated in figures 15 and 16. Figure 15 illustrates the representation of the sentence *The mouse that sees the dog that likes the boy chases the cat* in terms of the architecture discussed here. As with the sentence illustrated in figure 13, this sentence contains a subject-extracted relative clause. In this case, however, the relative clause itself contains a subject-relative clause, which results in the phrase *the mouse that sees the dog that likes the boy* as the agent of the verb *chases* of the main clause. Figure 15 illustrates that the recursive use of a subject-relative clause can be represented in a straightforward manner in terms of the architecture presented here. The representation of *The mouse that sees the dog* proceeds in the same manner as in figure 13. After that, the conjunction *noun-comp-verb* (*dog that likes*) is detected for the second time. As before, this will result in the activation of the gating circuits for the rc subassemblies, which binds the currently active NP and VP assemblies by means of their rc subassemblies. The currently active NP and VP assemblies in this case are N₂ (*dog*) and V₂ (*likes*). The binding of *likes* and *boy* proceeds in the same manner as the binding of *sees* and *dog*. Likewise, the binding of *mouse*, *chases* and *cat* proceeds in the same manner as with the sentence illustrated in figure 13.

The double right-branching sentence *The mouse chases the cat that sees the dog that likes the boy* can be represented in a similar manner as the sentence illustrated in figure 14, by binding the VP assembly of *sees* (V₁) to the NP assembly of *cat* (N₄) by means of their rc subassemblies. Thus, the recursive use of right-branching can be represented in the architecture discussed here. However, this is not the case for the recursive use of center-embedding. Figure 16 illustrates the representation of the sentence *The mouse that the dog that the boy likes sees chases the cat* in terms of the architecture discussed here. As with the sentence illustrated in figure 14, this sentence contains an object-extracted relative clause (*The mouse that the dog ... sees*) as the agent of the verb *chases* of the main clause. In this case, however, the relative clause itself contains an object-extracted relative clause (*the dog that the boy likes*), which is the agent of the verb *sees*. The result is a sentence with a double center-embedding.

Sentences with double center-embeddings of the kind illustrated in figure 16 are so complex that they are unacceptable for most people (e.g., Gibson, 1998). This processing difficulty is reflected in the representation illustrated in figure 16. The

processing of *The mouse that the dog* proceeds in the same manner as illustrated in figure 14, including the activation of T_1 (for *mouse*) and N_2 (for *dog*). In figure 14, the occurrence of the verb *sees* resulted in the binding of N_2 (*dog*) as the agent of *sees* and T_1 (*mouse*) as the theme of *sees*. In this case, however, the noun phrase *the boy* is processed before N_2 and T_1 can be bound to *sees*. As before, a new noun activates a new NP assembly, which results in the inhibition of the previously active NP assembly. In this case, this will result in the activation of N_3 and the inhibition of N_2 . A similar process occurred with the phrase *the mouse that the dog*, which resulted in the activation of N_2 (*dog*) and the inhibition of N_1 (*mouse*). Because *mouse* was not yet bound to *sees*, the assembly T_1 was activated and bound to the rc subassembly of N_1 . In a similar manner, the activation of N_3 will result in the inhibition of N_2 and the activation of the assembly T_2 . As with T_1 and N_1 , T_2 should be bound to the rc subassembly of N_2 . However, this can only occur if T_1 is inhibited at this moment, otherwise T_1 will also be bound to the rc subassembly of N_2 .

Furthermore, N_1 is bound to the assembly S in figures 14 and 16, which enables the reactivation of N_1 when the processing of the relative clause is completed. N_1 was bound to S because *mouse* was interpreted as the external argument of the sentence. However, the binding of N_1 to S prohibits the binding of N_2 to S, because a sentence has only one external argument. Or else, when both N_1 and N_2 are bound to S, an ambiguity will arise about the external argument of the sentence. Thus, when N_2 is inhibited, it cannot be reactivated again (without creating an ambiguity). Likewise, T_1 cannot be reactivated again. Basically, this results from the fact that the phrases *the mouse that the dog* and *the dog that the boy* are syntactically similar structures. As a result, T_1 and T_2 play structurally similar roles in the process of sentence representation. This entails that only one of them can be active at the same moment, or else a conflict between the roles of T_1 and T_2 will occur.

When T_1 (and N_2) are inhibited, the occurrence of the verb *likes* in the sentence will result in the binding of N_3 (*boy*) as the agent of *likes* and T_2 (*dog*) as the theme of *likes*. However, the occurrence of the second verb *sees* in the sentence creates a fundamental difficulty. It should be bound to N_2 and T_1 as indicated with the dashed lines in figure 16. However, binding in this model can only occur between active assemblies, but N_2 and T_1 have already been inhibited, and, as argued above, they cannot be reactivated in an unambiguous manner. As a result, the representation of the sentence will fail.

Following the processing illustrated in figure 14, the verb *sees* could in fact reactivate N_1 , to initiate the representation of the main clause of the sentence. In that case, *mouse* would be bound as the agent to *sees*, and *sees* would be interpreted as the verb of the main clause. However, this creates a conflict with the third verb *chases* that occurs in the sentence. In fact, the representation scheme illustrated in figure 16 would predict that it is easier to process the incorrect sentence *The mouse that the dog that the boy likes sees the cat* (or *The mouse that the dog that the boy likes chases the cat*). This would result in the representation of the phrases *the mouse sees the cat* (or *the mouse chases the cat*) and *the dog that the boy likes*, leaving the relation between both phrases undetermined. The fact that these ungrammatical sentences are more acceptable has been observed in the literature (e.g., Gibson, 1998).

Pinker (1994) argued that the processing difficulty of structures like double center-embeddings results from the difficulty of keeping track of more than one incomplete phrase of the same kind. A similar argument is based on the incomplete dependency hypothesis (e.g., Lewis, 1996; Stabler, 1994), which holds that processing complexity is related to the presence of multiple incomplete syntactic dependencies (e.g., between agent and verb) of the same kind. In the sentence illustrated in figure 16, the phrases *the mouse that the dog* and *the dog that the boy* are two incomplete phrases of the same kind. To complete both phrases, they have to be bound in the manner illustrated in figure 16. However, binding in this model can only occur between active assemblies. As a result, ambiguities will arise when the assemblies of both *the mouse that the dog* and *the dog that the boy* remain active at the same time, as described above.

Memory and capacity

Just and Carpenter (1992) modeled the effect of differences in memory capacity on sentence comprehension. Sentence comprehension in their model is based on symbolic processing (production systems), with activation values assigned to each of the representational elements in the model. When the maximum of activation that can be distributed over these elements is about to be exceeded, a decline of activation of the elements will occur. In turn, this will (eventually) result in a loss of information stored in memory. Individual differences in capacity are related with differences in the maximum of activation that can be assigned in the comprehension process.

Sentence representation in the model discussed here consists of a temporary binding between word assemblies by means of the memory circuits presented in figure 5. A sentence representation remains intact as long as each of the delay assemblies in the representation remains active. Thus, the capacity of sentence representation in this model is related to the sustained activation in the delay assemblies. In a similar manner, comprehension performance in this model will be related to the sustained activation in the delay assemblies (and to the information retained in the neural control circuits, which also depends on sustained activation).

Delay assemblies remain active due to the reverberating nature of the activation in the assembly (e.g., Amit & Brunel, 1997; Wang, 2001). However, over time a reverberating assembly will become unstable, losing its ability to sustain activation without external stimulation (e.g., Brunel, 2000; Koulakov, 2001). The conditions under which this decay of activation occurs are not yet fully known. Therefore, we will model the effect of activation decay on the capacity of sentence representation in a qualitative manner. We assume that a delay assembly loses its sustained activation with a probability p that increases linearly with time t , thus $p = \alpha t$ (for as long as $\alpha t < 1$). The probability that the assembly remains active is then given by $1 - \alpha t$. For simplicity, we divide time t in discrete intervals t_n , and we assume that in each new interval a new delay assembly is activated in the process of sentence representation. In this way, the probability that the first activated delay assembly is still active after n intervals is given by $1 - \alpha n$. Likewise, the probability that the second activated delay assembly is still active after n intervals is given by $1 - \alpha(n-1)$. Continuing in this manner, the probability that the last (n^{th}) activated assembly is still active is given by $1 - \alpha$. The probability P that all n assemblies are still active after n intervals is then given by the joint probability $P = (1 - \alpha n) (1 - \alpha(n-1)) \dots (1 - 2\alpha) (1 - \alpha)$.

In figure 17 we plot this probability for different values of n . It is clear that P gradually declines with time (given by n), which means that information will be lost over time. Even without the oversimplified assumption that assemblies are activated one by one in discrete time intervals in the process of sentence representation, it is clear that sentence representation (comprehension) is a sequential process, which will result in a sequential activation of delay assemblies in this model. Figure 17 shows in a qualitative manner that the probability that a number of n assemblies can be active declines with n . This result can be interpreted in two ways. First, it shows that the number of sentence representations that can remain active is limited. Second, it shows that it is more difficult to retain or even produce a representation of a more complex sentence, because sentence complexity is related with the number of active delay assemblies in this model (e.g., see figure 13 and 14). The rate with which P declines depends on α . In figure 17, we plot P for three values of α . Figure 17 shows that differences in memory capacity (e.g., low versus high capacity readers, Just & Carpenter, 1992) result from different values of α in this model.

Figure 17 illustrates that effect of memory capacity on sentence representation (and comprehension) in this model does not result from a maximum of activation, but instead results from the neural dynamics of reverberating (sustained) activity. However, the overall amount of activation in the language areas of the brain could influence the dynamics of reverberating activity, for instance, due to an increase of noise activity in the delay assemblies (e.g., see Brunel, 2000). Likewise, attention could have an effect on sustained activation, as observed in the visual cortex. The incorporation of these factors would provide a more elaborate model of the influence of neural dynamics on memory capacity and sentence comprehension, compared to the effect illustrated in figure 17.

Discussion

In this article we discussed a neural implementation of basic aspects of language processing. In particular, we concentrated on the question of how sentence structure can be represented in terms of neural assemblies, and how information can be retrieved from that representation.

A fundamental aspect of sentence representation is the combinatorial productivity that underlies this form of representation. The magnitude of the lexicon of a native language user is in the order of 10^5 (e.g., Bloom, 2000). However, the number of sentences that can be formed in a natural language exceeds this magnitude by far. Linguistic theories typically assert the infinite productivity of language, which means that an unlimited number of different sentences can be formed in a natural language (e.g., Pinker, 1989; Weibelhuth, 1995; Jackendoff, 1999; Sag & Wasow, 1999; Chomsky, 2000). However, even if the number of words in a sentence is limited, the number of sentences that can be produced is enormous. For instance, the set of sentences that can be formed with a sentence length of 20 words is in the order of 10^{20} Pinker (1998). Any model of language processing has to account for the ability to represent any sentence from a set of sentences of this magnitude. This is not a great problem for symbolic processing models (and linguistic theories). These models and theories represent sentences in terms of individual words as constituents. A sentence is represented by a representation of the sentence structure, with copies of word representations inserted in the appropriate 'slots'. In a similar manner, to achieve combinatorial productivity of

sentence representation, activation models of language processing have relied on hybrid architectures, in which symbolic representations and processes are combined with activation processes or resources (e.g., Just & Carpenter, 1992; Miikkulainen, 1996; Vosse & Kempen, 2000).

In contrast, productivity of sentence representation is a much harder problem for neural models of language processing. For instance, McClelland and Kawamoto (1986) presented a neural model of thematic role assignment in sentence processing. However, this model was restricted to only one sentence structure. Furthermore, the model could not represent different tokens of the same type, such as *dog_{agent}* and *dog_{theme}* in *dog chases dog*. The Simple Recurrent Network (SRN) models of sentence representation (e.g., Elman, 1991; Christiansen & Chater 1999) are more flexible in this respect. However, SRNs generate sentence representations on the basis of a statistical average of the set of sentences they are trained with. As a result, SRNs are limited in their productivity of sentence representation, because they have to be trained with a substantial amount (in the order of 10%) of the set of sentences that they can represent (e.g., see Van der Velde, Van der Kleij van der Voort & de Kamps, *subm*). Furthermore, SRNs can only predict the category of a word that would follow in a sentence context. Thus, they can predict that a noun is likely to follow after *The mouse chases (...)* or *The cat chases (...)*. However, they cannot answer specific probe (or 'binding') questions, which means that they cannot distinguish between *The mouse chases the cat* and *The cat chases the mouse*.

The neural architecture that we have presented in this article can represent different sentence structures in a productive manner. Furthermore, it can answer specific questions about the thematic relations expressed in the sentence. Productivity of sentence representation is achieved because the model uses word representations as constituents in the representation of a sentence. Words are represented by means of neural assemblies (Hebb, 1949), in line with recent proposals of word representation in the brain (Pulvermüller, 1999; 2000). Sentence structure is represented by first binding words temporarily to structural assemblies, which are then bound temporarily to one another in a manner that preserves the thematic relations between the words. This two-step representation process solves the problems associated with the neural implementation of productive sentence representation (Van der Velde, 1995). First, it allows the representation of different tokens (copies) of the same word, because each token is bound to a different structure assembly. Second, it allows the representation of sentence structure. The difference between *The mouse chases the cat* and *The cat chases the mouse* cannot be represented on the basis of associations between the word assemblies for *mouse*, *cat* and *chases*. But when the word assemblies are first bound to structure assemblies, the two sentences can be distinguished by means of different bindings between the structure assemblies. Third, the two-step representation process results in combinatorial productivity of sentence representation. Sentence structure is represented on the level of the structure assemblies. Individual words only need to be bound to the appropriate structure assemblies to represent a particular sentence. In this way, arbitrary thematic relations can be expressed between arbitrary words (e.g., *The chair chases the table*).

By straightforward extensions of the basic representational architecture, more complex sentence structures, including ones with relative clauses, can be represented as well, again in a productive manner. The information (thematic relations) stored in the

architecture can be retrieved by asking probe or binding questions. These questions provide the information that can be used to reactivate the overall structure of the sentence that is stored in the architecture. Using this reactivated structure, the answer to the question can be produced.

Temporal binding between two assemblies in this model is achieved by means of reverberating activity in a delay assembly in the memory circuit that connects the assemblies. Reverberating activity is used in the brain to retain information temporarily (e.g., Fuster, 1973; Wang, 2001). To activate the delay assembly in the memory circuit, the two assemblies connected by the circuit have to be active simultaneously at a given time. As we showed, this requirement has consequences for the way in which sentences are processed, which can be related with the difficulty of processing certain sentence structures (like sentences with double-embeddings) observed in human language performance. Hence, differences in sentence complexity can be related to the dynamics of the architecture presented here. Finally, capacity for sentence representation in the architecture presented here can be related to the way in which reverberating activity will decay over time (e.g., Brunel, 2000; Koulakov, 2001).

Obviously, the architecture we presented here is only a first step towards a theory of language processing in the brain. In particular, we have only described the process of sentence representation in general terms. In line with the notion of lexicalist parsing (e.g., Jackendoff, 1999; Vosse & Kempen, 2000), we assume that a word will activate a neural circuit that represents information about the nature of the word (e.g., whether it is a noun or a verb) and the structural relations that the word can have with other words in the sentence. Interactions between these neural circuits will then result in the activation of selective gating circuits (e.g., those for 'agent' or 'theme'). In turn, the activated gating circuits enable the binding of active structure assemblies in relation with the thematic relations between the words in the sentence. In this way, a representation of a sentence unfolds in time, in line with the incremental way in which a sentence representation is generated in human language processing (e.g., Gibson, 1998). A fully developed theory of language processing in the brain will have to describe these neural circuits and their interactions in an explicit manner.

Acknowledgements

Correspondence address: Frank van der Velde, Unit of Cognitive Psychology, Leiden University, Wassenaarseweg 52, 2333 AK Leiden, The Netherlands. E-mail: vdvelde@fsw.leidenuniv.nl

References

- Abeles, M. (1991). *Corticonics*. Cambridge: Cambridge University Press.
- Amit, D. J. (1989). *Modeling brain function*. Cambridge: Cambridge University Press.
- Amit, D. J. (1995). The Hebbian paradigm reintegrated: Local reverberations as internal representations. *Behavioral and Brain Sciences*, 18, 617-657.
- Amit, D. J. & Brunel, N. (1997). Global spontaneous activity and local structured (learned) delay activity in cortex. *Cerebral Cortex*, 7, 237-252.
- Bates, E. & Goodman, J. C. (1997). On the inseparability of grammar and the lexicon: Evidence from acquisition, aphasia and real-time processing. *Language and Cognitive processes*, 12, 507-584.

- Bloom, P. (2000). *How children learn the meanings of words*. Cambridge, MA: MIT Press.
- Brunel, N. (2000). Persistent activity and single-cell frequency-current curve in a cortical network model. *Network: Computation in Neural Systems*, **11**, 261-280.
- Calvin, W. H. (1995). Cortical columns, modules, and Hebbian cell assemblies. In M. A. Arbib, G. Adelman and P. H. Arbib (Eds.) *The handbook of brain theory and neural networks*, (pp. 269-272). 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.
- Caplan, D., Baker, C. & Dehaut, F. (1985). Syntactic determinants of sentence comprehension in aphasia. *Cognition*, **21**, 117-175.
- Caplan, D. & Futter, C. (1986). Assignment of thematic roles by an agrammatic aphasic patient. *Brain and Language*, **27**, 117-135.
- Caramazza, A. & Zurif, E. (1976). Dissociation of algorithmic and heuristic processes in sentence comprehension: Evidence from aphasia. *Brain and Language*, **3**, 572-582.
- Chomsky, N. (2000). *New horizons in the study of language and mind*. Cambridge: Cambridge University Press.
- Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, **23**, 157-205.
- Compte, A., Brunel, N., Goldman-Rakic, P. S. & Wang, X. J. Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. *Cerebral Cortex*, **10**, 910-923.
- Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, **7**, 195-225.
- Fodor, J. A. & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: a critical analysis. *Cognition*, **50**, 151-170.
- Fuster, J. M. (1973). Unit activity in prefrontal cortex during delayed-response performance: Neuronal correlates of transient memory. *Journal of Neurophysiology*, **36**, 61-78.
- Gerstner, W. (1995). Time structure of the activity in neural-network models. *Physical Review E*, **51**, 738-758.
- Gerstner, W. (2000). Population dynamics of spiking neurons: fast transients, asynchronous states, and locking. *Neural Computation*, **12**, 43-89.
- 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.
- Grodzinsky, Y. & Finkel, L. (1998). The neurology of empty categories: Aphasics' failure to detect ungrammaticality. *Journal of Cognitive Neuroscience*, **10(2)**, 281-292.
- Grodzinsky, Y. (2000). The neurology of syntax: Language use without Broca's area. *Behavioral and Brain Sciences*, **23**, 1-17.
- Grossberg, S. (2000) The complementary brain: unifying brain dynamics and modularity. *Trends in Cognitive Sciences*, **4**, 233-246.
- Hebb, D. O. (1949). *The Organisation of Behaviour*. New York : Wiley.
- Hickok, G., Zurif, E. & Canseco-Gonzalez, E. (1993). Structural description of agrammatic comprehension. *Brain and Language*, **45**, 371-395.

- Itty, L. & Koch, C. (2001) Computational modelling of visual attention. *Nature Reviews Neuroscience*, **2**, 194-203.
- Jackendoff, R. (1999). Parallel constraint-based generative theories of language. *Trends in Cognitive Sciences*, **3**, 393-400.
- Just, M. A., Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, **99**, 122-149.
- Just, M. A., Carpenter, P. A., Keller, T. A., Eddy, W. F. & Thulborn, K. R. (1996). Brain activation modulated by sentence comprehension. *Science*, **274**, 114-116.
- Keller, T. A., Carpenter, P. A. & Just, M. A. (2001). The neural basis of sentence comprehension: a fMRI examination of syntactic and lexical processing. *Cerebral Cortex*, **11**, 223-237.
- Koulakov, A. A. (2001). Properties of synaptic transmission and the global stability of delayed activity states. *Network: Computation in Neural Systems*, **12**, 47-74.
- Lewis, R. (1996). A theory of grammatical but unacceptable embeddings. *Journal of Psycholinguistic Research*, **25**, 93-116.
- MacDonald, M. C., Perlmutter, N. J. & Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution. *Psychological Review*, **101**, 676-703.
- McClelland, J. L. & Kawamoto, A. H. (1986). Mechanisms of sentence processing: Assigning roles to constituents of sentences. In J. L. McClelland and D. E. Rumelhart (eds.) *Parallel Distributed Processing Vol 2* (pp 272-325), Cambridge, MA: MIT Press.
- Miller, E. K. (2000). The prefrontal cortex and cognitive control. *Nature Reviews Neuroscience*, **1**, 59-65.
- Mountcastle, V. B. (1998). *The Cerebral Cortex*. Cambridge, MA: Harvard University Press.
- Miikkulainen, R. (1996). Subsymbolic case-role analysis of sentences with embedded clauses. *Cognitive Science*, **20**, 47-73.
- Pinker, S. (1989). *Learnability and cognition*. Cambridge, MA: MIT Press.
- Pinker, S. (1994). *The language instinct*. London: Penguin.
- Pinker, S. (1998). *How the mind works*. London: Penguin
- Pulvermüller, F. (1999). Words in the brain's language. *Behavioral and Brain Sciences*, **22**, 253-336.
- Pulvermüller, F. (2001). Brain reflections of words and their meaning. *Trends in Cognitive Sciences*, **5**, 517-524.
- Sag, I. A. & Wasow, T. (1999). *Syntactic theory: A formal introduction*. Stanford: CSLI Lecture Notes.
- Sougné, J. (1998). Connectionism and the problem of multiple instantiation. *Trends in Cognitive Sciences*, **2**, 183-189.
- Stabler, E. P. (1994). The finite connectivity of linguistic structures. In: C. Clifton, L. Frazier and K. Rayner (Eds.), *Perspectives on sentence processing*. Hillsdale, NJ: Erlbaum.
- Stromswold, K., Caplan, D., Alpert, N. & Rauch, S. (1996). Localization of syntactic comprehension by positron emission tomography. *Brain and Language*, **52**, 452-473.
- Usher, M. & Niebur, E. (1996). Modeling the temporal dynamics of IT neurons in visual search: A mechanism for top-down selective attention. *Journal of Cognitive Neuroscience*, **8**, 311-327.

- 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. (1997). On the use of computation in modelling behavior. *Network: Computation in Neural Systems*, **8**, 1-32.
- Van der Velde, F. (2001). The study of brain function: Methodological considerations. In: Amit, D. & Parisi, G. (Eds), *Frontiers of life, Vol III, part two: Building blocks for intelligence systems*. San Diego: Academic 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**, (4), 479-491.
- Van der Velde, F., Van der Voort van der Kleij, G. T. & de Kamps, M. (subm). Lack of combinatorial productivity in language processing with simple recurrent networks.
- Van Valin, R. D. (2001). *An introduction to syntax*. Cambridge: Cambridge University press.
- Wang, X-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neurosciences*, **24**, 455-463.
- Wilson, H. R. & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, **12**, 1-14.
- Vosse, Th. & Kempen, G. (2000). Syntactic structure assembly in human parsing: A computational model based on competitive inhibition and a lexicalist grammar. *Cognition*, **75**, 105-143.
- Weibelhuth, G. (ed.), (1995). *Government and binding theory and the minimalist program*. Cambridge, MA: Blackwell.

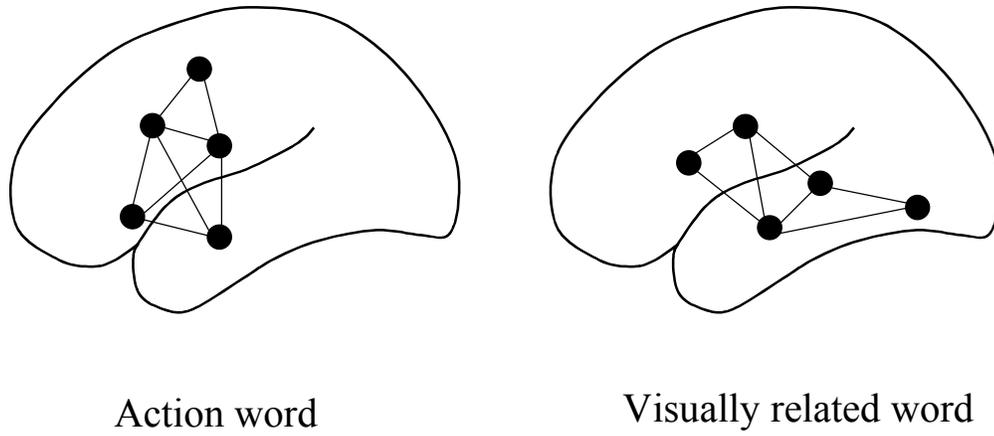
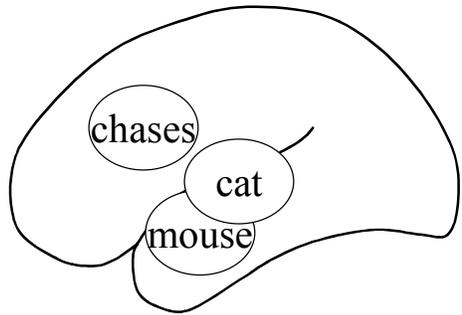
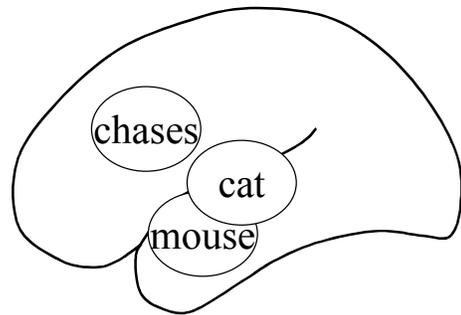


Figure 1. Illustration of the neural assemblies, distributed over the brain, that could represent action words (left) or visually related words (right).



The mouse chases the cat



The cat chases the mouse

Figure 2. Illustration of the word assemblies for *mouse*, *cat* and *chases* that will be activated with the sentence *The mouse chases the cat* (left) and *The cat chases the mouse* (right).

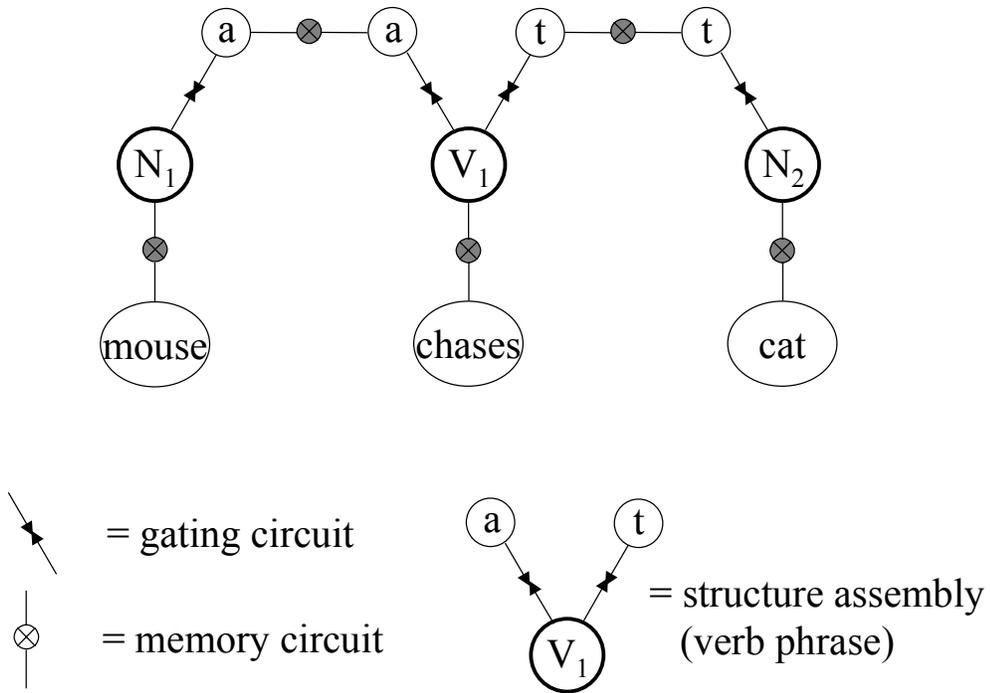


Figure 3. The representation of the sentence *The mouse chases the cat* with the word assemblies illustrated in figure 2 and the structure assemblies for 'noun-phrases' (N_i) and 'verb-phrases' (V_i). Assemblies are connected with gating or memory circuits. A structure assembly consists of a main assembly (N_i or V_i) and subassemblies for agent (a) and theme (t).

Gating Circuit

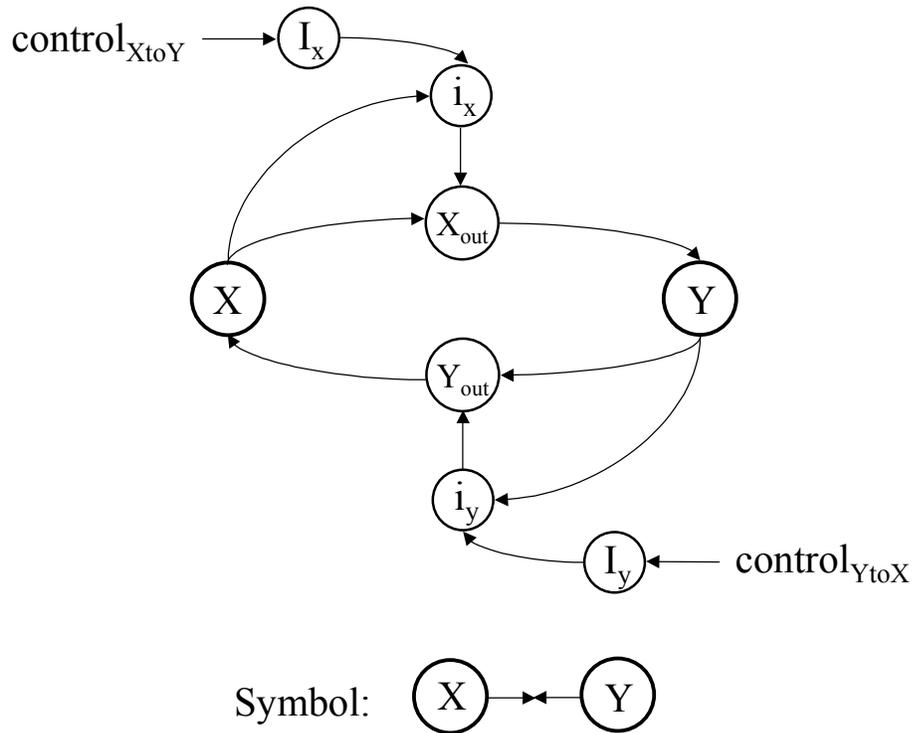


Figure 4. A bi-directional gating circuit, based on disinhibition. X , Y , X_{out} , and Y_{out} are populations of excitatory neurons. I_x , I_y , i_x and i_y are inhibitory neurons. The gating circuit can be activated with the control signals $control_{XtoY}$ and $control_{YtoX}$, which derive from excitatory populations outside the circuit.

Memory Circuit

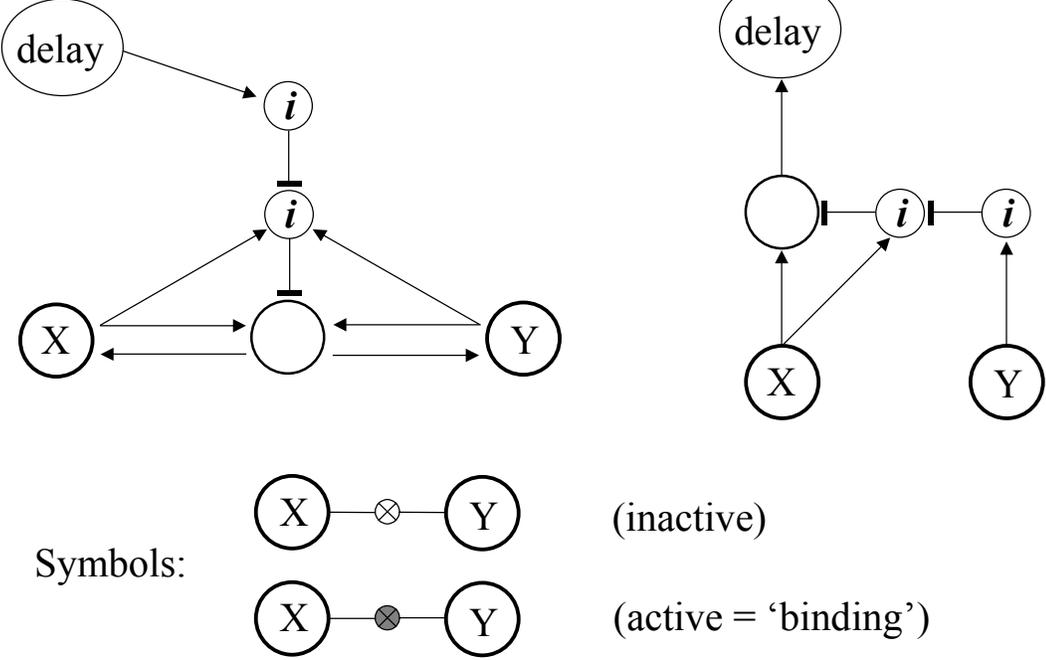


Figure 5. Memory circuit (left). A memory circuit consists of a gating circuit presented in figure 4, in which the control signals derive from a delay assembly. A delay assembly is activated by a disinhibition circuit (right).

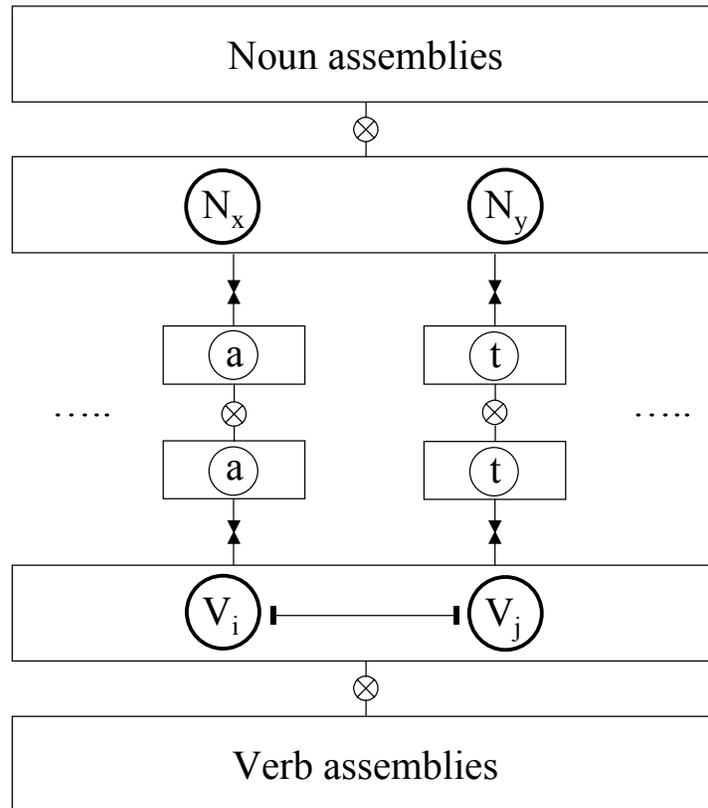


Figure 6. Architecture for sentence representation with neural assemblies. Verb assemblies are connected to verb-phrase assemblies (V_i) by means of (initially inactive) memory circuits. Likewise, noun assemblies are connected to noun-phrase assemblies (N_i) by means of (initially inactive) memory circuits. The verb-phrase and noun-phrase assemblies are connected through their subassemblies for agent (a) and theme (t), by means of (initially inactive) memory circuits.

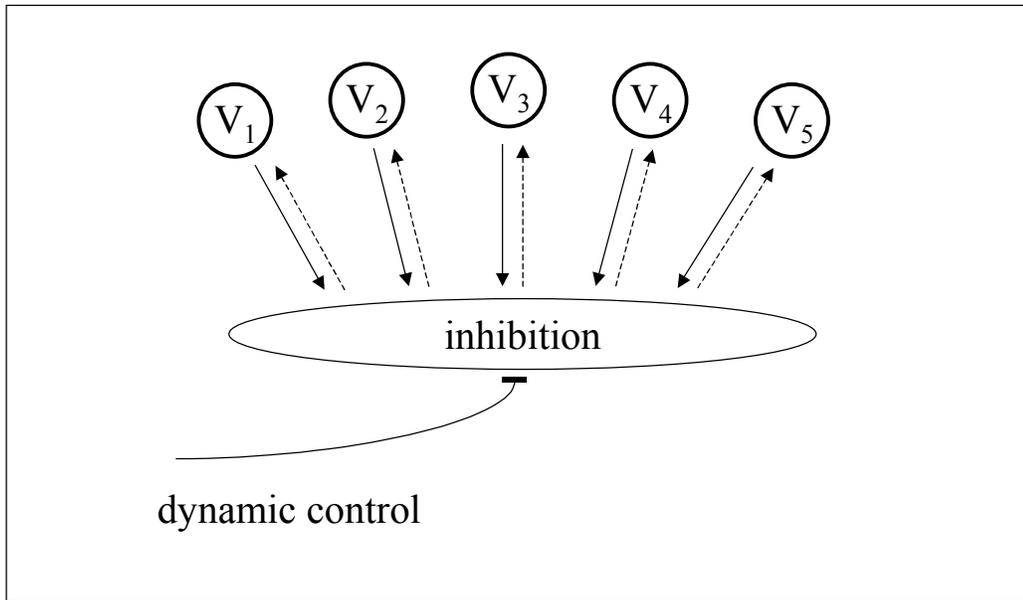


Figure 7. Competition between verb-phrase assemblies (V_i) by means of a population of inhibitory neurons. This population can itself be inhibited by means of 'dynamic control'.

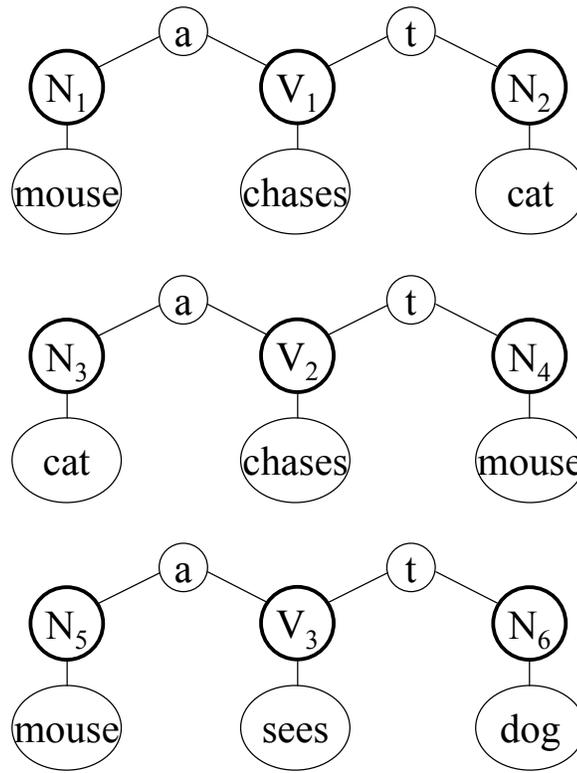


Figure 8. Simultaneous representation of *The mouse chases the cat*, *The cat chases the mouse* and *The mouse sees the dog* in the architecture illustrated in figure 6. N_i and V_i are noun-phrase and verb-phrase assemblies, respectively, with subassemblies for agent (a) and theme (t).

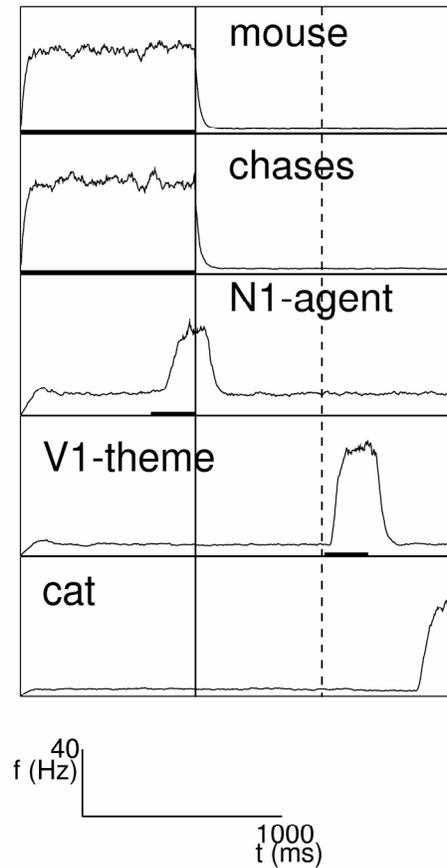


Figure 9. Simulated activations for the word assemblies for *mouse*, *chases* and *cat*, and the subassemblies for N₁-agent and V₁-theme. The simulations result from the production of the answer to the question "Whom does the mouse chase?" when the sentences illustrated in figure 8 are represented simultaneously. The question initiates the activation of *mouse*, *chases* and the gating circuits for 'agent'. After a competition process, the assembly for *cat* is activated as answer to the question. Horizontal bars and vertical lines indicate the timing of particular events (see text for explanation).

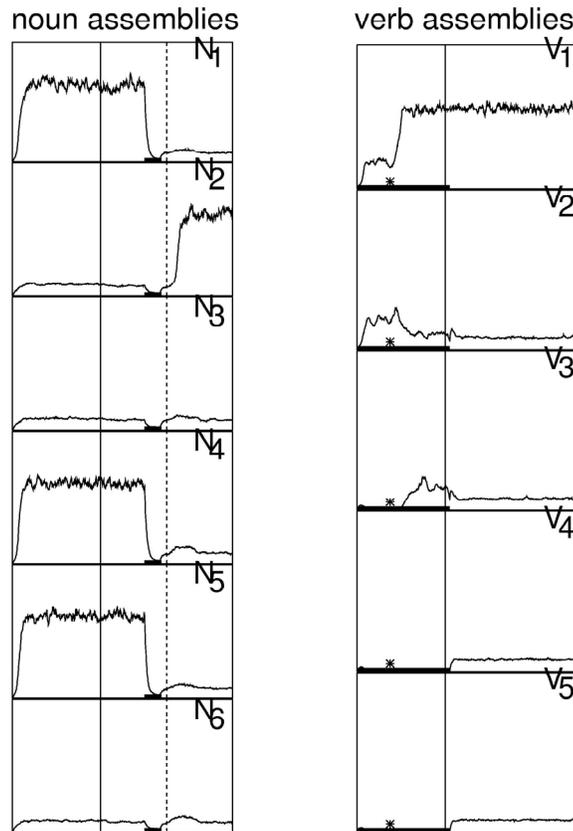


Figure 10. Activations for the noun-phrase assemblies N_1 , N_2 , N_3 , N_4 , N_5 and N_6 (left) and the verb-phrase assemblies V_1 , V_2 , V_3 , V_4 and V_5 (right), during the competition process between the verb-phrase assemblies that results in the activation of *cat* in figure 9. Horizontal bars, the asterisk and vertical lines indicate the timing of particular events, in relation with figure 9 (see text for explanation).

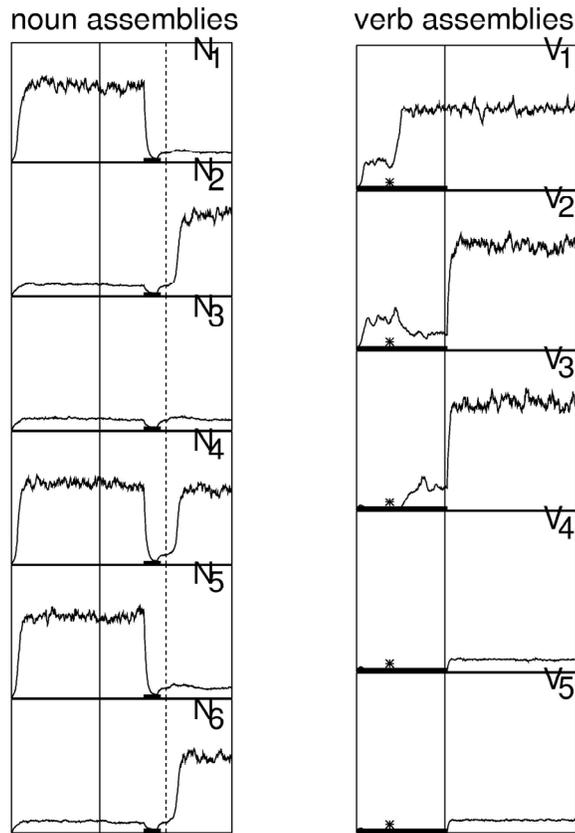


Figure 11. Activations of the assemblies illustrated in figure 10 when the competition process between the verb-phrase assemblies fails as a result of a different timing of events. Horizontal bars, the asterisk and vertical lines indicate the timing of particular events, in relation with figure 9 (see text for explanation).

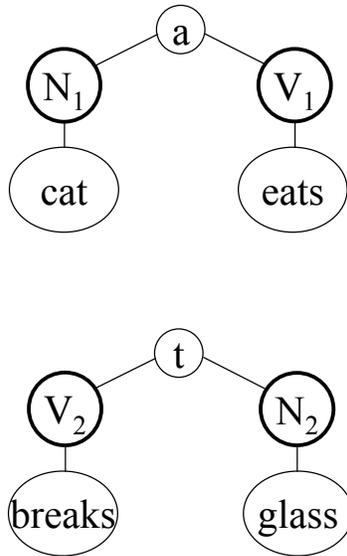


Figure 12. Representation of the sentences *The cat eats* and *The glass breaks* in the architecture illustrated in figure 6.

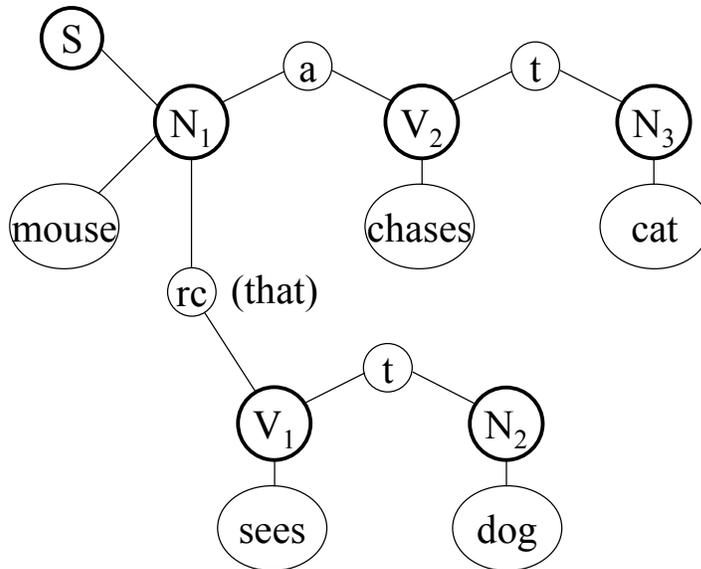


Figure 13. Representation of the sentence *The mouse that sees the dog chases the cat* in the architecture illustrated in figure 6.

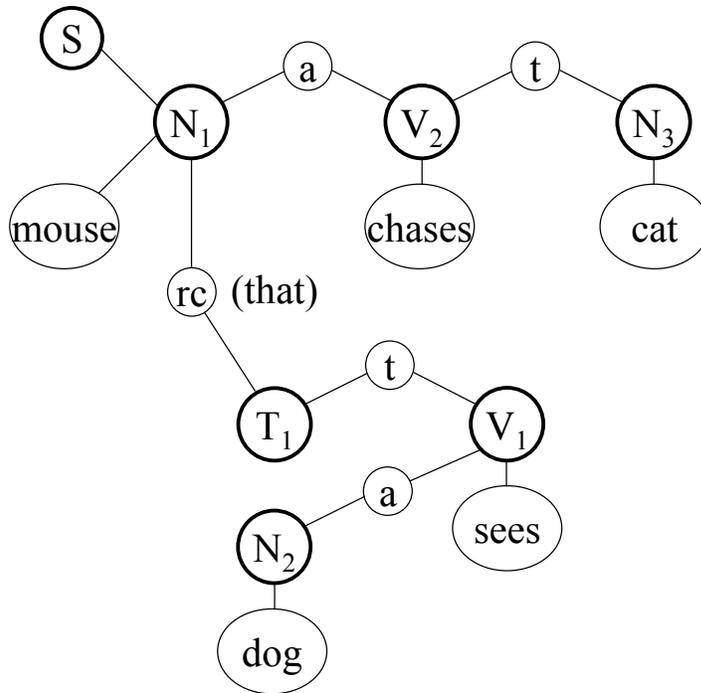


Figure 14. Representation of the sentence *The mouse that the dog sees chases the cat* in the architecture illustrated in figure 6.

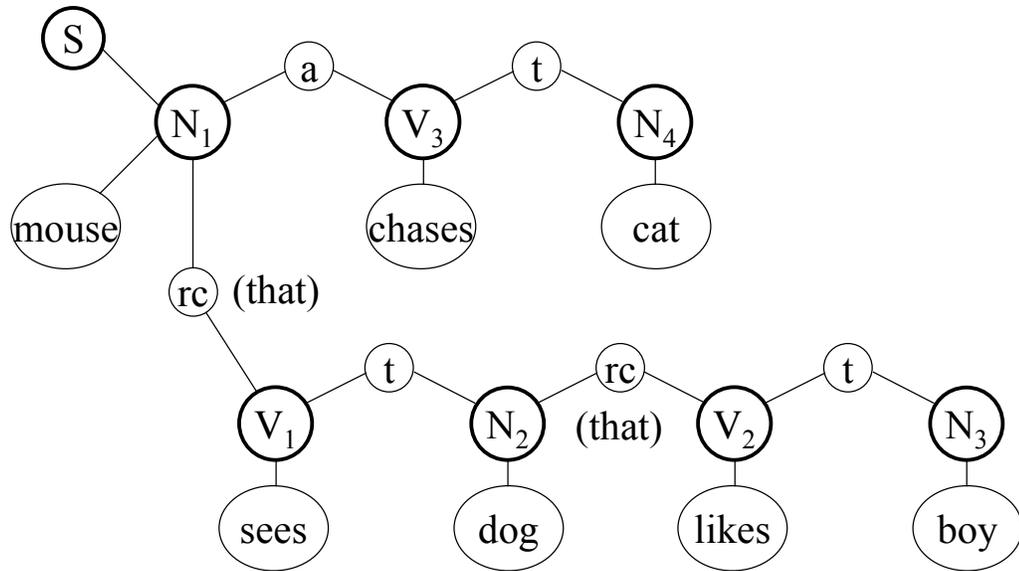


Figure 15. Representation of the sentence *The mouse that sees the dog that likes the boy chases the cat* in the architecture illustrated in figure 6.

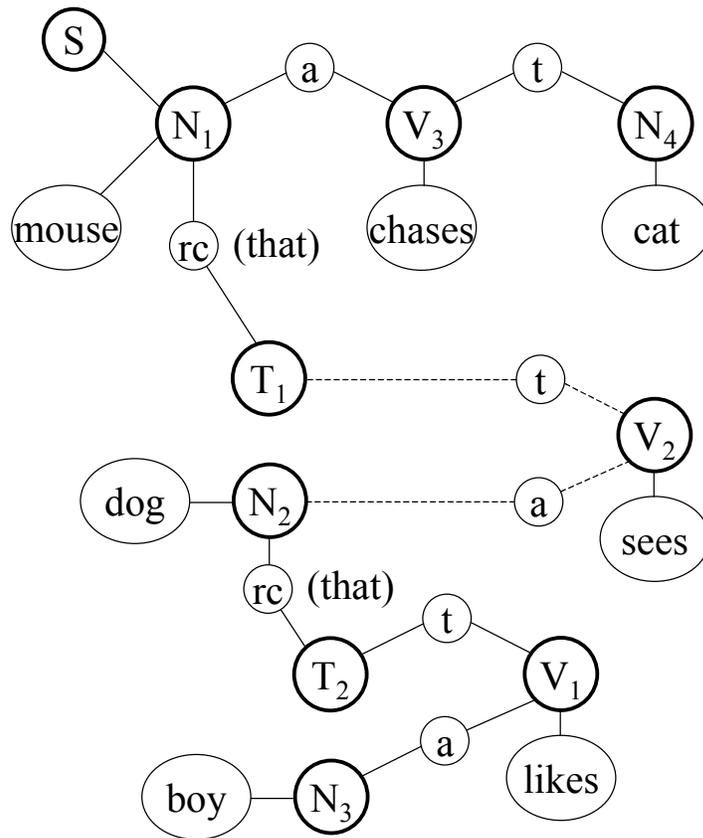


Figure 16. Representation of the sentence *The mouse that the dog that the boy likes sees chases the cat* in the architecture illustrated in figure 6. Dashed lines indicate bindings that result in ambiguities during the processing of this sentence (see text for explanation).

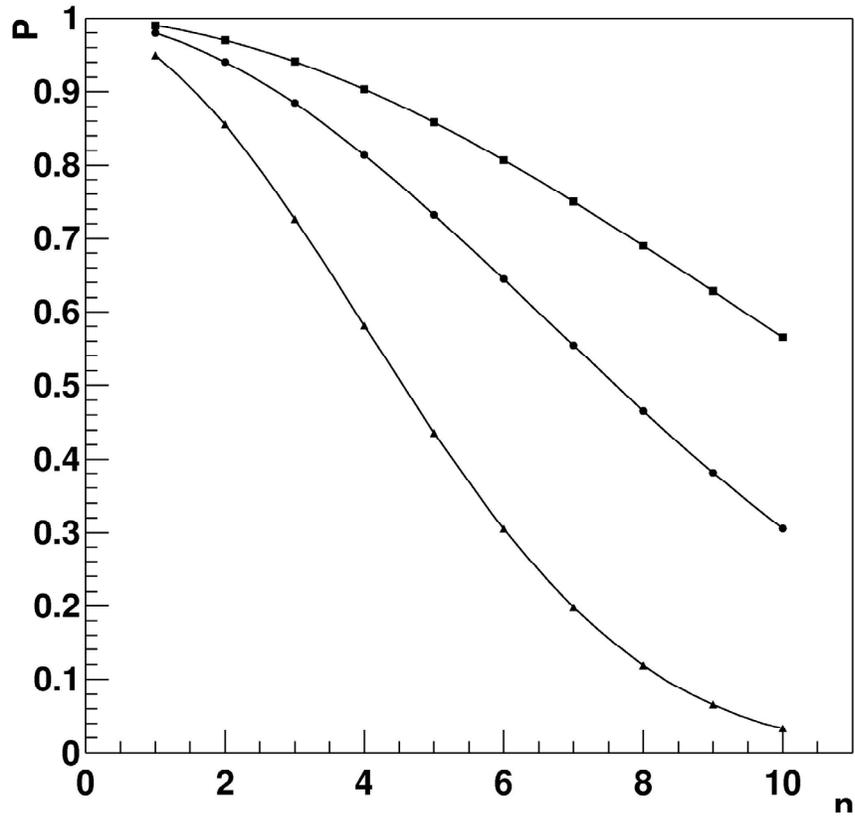


Figure 17. Memory capacity in the architecture illustrated in figure 6. The probability that n delay assemblies can remain active declines with n . Different capacities result from different values of α (with $\alpha = 0.01$, $\alpha = 0.02$ and $\alpha = 0.05$ for 'high', 'medium' and 'low' capacity).

Appendix

The simulations are based on a network of excitatory and inhibitory populations. The activation of population i is modeled with the population rate A_i , which is defined as the fraction of neurons which fire during a time interval $[t, t + dt]$, divided by dt (Wilson and Cowan, 1972; Gerstner 1995). The population rate A_i is given by (with $\alpha = E$ for excitatory populations and $\alpha = I$ for inhibitory populations):

$$\tau_\alpha \frac{dA_i}{dt} = (-A_x + F(\sum_j w_{ij} A_{ij})) + A_i I_{noise} \quad (1)$$

τ_E (τ_I) is the time constant for an excitatory (inhibitory) population. We took $\tau_E = 10$ ms and $\tau_I = 5$ ms, reflecting the fact that inhibitory neurons operate faster than excitatory neurons. Population time constants are believed to be dominated by synaptic time constants (Gerstner, 2000) and the values we chose are of that order of magnitude. The w_{ij} (or $w_{j \rightarrow i}$) are the efficacies from population j onto population i . We respect Dale's law (Abeles, 1991): w_{ij} is negative if and only if j is an inhibitory population. Every 1 ms a noise current, given by the fraction I_{noise} of the population activation, is injected into each population with $\mu = 0, \sigma = 0.02$.

For $F(x)$ we took:

$$F(x) = \frac{f_{max}}{(1 + e^{-\beta(x-\theta)})} \quad (2)$$

with $f_{max} = 30$ Hz, $\beta = 1$ and $\theta = 3$.

Populations are grouped in a hierarchy of circuits. The lowest level consists of the bidirectional gating circuits (figure 4) and the memory gating circuits (figure 5). Each of these circuits consist of 8 populations, as given by the circuit in figure 4: $X, Y, X_{out}, Y_{out}, i_x, i_y, I_x,$ and I_y . X or Y may be part of two circuits (i.e., a gating and a memory circuit). The circuit in figure 4

is described by the following equations:

$$\begin{aligned}
\tau_E \frac{dX}{dt} &= -X + F(\text{input}_X + w_{Y_{out} \rightarrow X} Y_{out}) + X I_{noise} & (3) \\
\tau_E \frac{dY}{dt} &= -Y + F(\text{input}_Y + w_{X_{out} \rightarrow Y} X_{out}) + Y I_{noise} \\
\tau_E \frac{dX_{out}}{dt} &= -X_{out} + F(w_{X \rightarrow X_{out}} X - w_{i_x \rightarrow X_{out}} i_x) + X_{out} I_{noise} \\
\tau_E \frac{dY_{out}}{dt} &= -Y_{out} + F(w_{Y \rightarrow Y_{out}} Y - w_{i_y \rightarrow Y_{out}} i_y) + Y_{out} I_{noise} \\
\tau_I \frac{di_x}{dt} &= -i_x + F(w_{X \rightarrow i_x} X - w_{I_x \rightarrow i_x} I_x) + i_x I_{noise} \\
\tau_I \frac{di_y}{dt} &= -i_y + F(w_{Y \rightarrow i_y} Y - w_{I_y \rightarrow i_y} I_y) + i_y I_{noise} \\
\tau_I \frac{dI_x}{dt} &= -I_x + F(\text{control}_{XtoY}) + I_x I_{noise} \\
\tau_I \frac{dI_y}{dt} &= -I_y + F(\text{control}_{YtoX}) + I_y I_{noise}
\end{aligned}$$

Here, input_X (input_Y) is the external input (i.e., from outside the circuit) on X (Y). We took $w_{X \rightarrow X_{out}} = w_{Y \rightarrow Y_{out}} = w_{X \rightarrow i_x} = w_{Y \rightarrow i_y} = 0.25$ and $w_{i_x \rightarrow X_{out}} = w_{i_y \rightarrow Y_{out}} = w_{I_x \rightarrow i_x} = w_{I_y \rightarrow i_y} = 1$. We took $w_{Y_{out} \rightarrow X} = w_{X_{out} \rightarrow Y} = 0.1$. The gating circuit can be activated by the control signals control_{XtoY} and control_{YtoX} . These control signals derive from two populations outside the circuit: I_{XtoY} and I_{YtoX} respectively. Their activation is given by f_{max} . They act on I_X and I_Y respectively with efficacy $w_{control} = 0.2$.

A memory circuit is a gating circuit in which the control signal is given by activation of a delay assembly. In our simulation, the memory circuits were simulated as gating circuits, in which the control signal was either 0 ('memory off') or f_{max} ('memory on'). The efficacies in the memory circuit are the same as in the gating circuit, except $w_{Y_{out} \rightarrow X} = w_{X_{out} \rightarrow Y} = 0.2$ in the memory circuit.

Besides the excitatory and inhibitory populations described above, we also

used delay populations in the simulations, in particular for the main assemblies of the structure assemblies. A delay population retains a high level of activity in absence of external stimulation, once the activation is raised beyond a certain level. Such populations have been investigated in great detail by Amit and Brunel (1997), Compte et al. (2000) and others. Based on these studies we assume the following properties for a delay population:

- A delay population is active once its input has been above a threshold θ_{delay} in the past and it has not been deactivated since.
- A delay population is deactivated once the net afferents to the assembly passes a certain negative threshold θ_{deact} (i.e., there is net inhibition).
- If a delay population is inactive, it functions as an ordinary population of excitatory neurons.

The use of inhibition to reset the network is studied in Compte et al., (2000). The assumption of the existence of θ_{deact} is justified by the need for such a population to be stable under small noise fluctuations. To be able to treat these populations as part of the network, we assume that:

- If the activity of the delay population is above threshold θ_{delay} and net input is excitatory, then the time constant of the population is $\tau_{delay} = \tau_E$.
- If the activity of the delay population is above threshold θ_{delay} but decreasing, while net input is above θ_{deact} , the time constant is very large: τ_{inf} .
- If net input is below θ_{deact} the time constant is reset to τ_E and, since net input is negative, memory activity will decay within approximately τ_E ms.

We took $\theta_{deact} = -0.2$, $\theta_{delay} = 4$ and $\tau_{delay} = 10000$ ms.

A structure assembly consists of a main assembly and subassemblies (e.g., for agent and theme). The main assembly is a delay population. The external input on the main assemblies derives from the word assemblies. Word assemblies are active with activation given by f_{max} and fraction I_{noise} of the population activation. Word assemblies act on their structure assemblies with efficacy $w_{input} = 0.2$. The subassemblies are excitatory populations, which are connected to their main assembly with gating circuits.

The VP structure assemblies (five in the simulations) are connected with a central inhibitory pool (see figure 7). The inhibitory pool acts on the VP assemblies with efficacy $w_{pool \rightarrow VP} = 0.03$. The inhibitory pool receives input from the VP assemblies with efficacy $w_{VP \rightarrow pool} = 0.03$. The central inhibitory pool creates competition between the VP assemblies. In that case, the competition is ‘on’. However, the central inhibitory pool can itself be inhibited. In that case, the competition is ‘of’.

The connection structure of the model at the highest level is as follows. Each agent subassembly of the NP assemblies is connected with each agent subassembly of the VP assemblies, by means of a memory circuit. These connections constitute a matrix of memory circuits between the agent subassemblies of the NP and VP assemblies. Likewise a matrix of memory circuits exist between the theme subassemblies of the NP and VP assemblies. In all, the simulated model consisted of 624 populations. Integration of the system of equations (1) evolved simultaneously for the entire model, using fourth-order Runge-Kutta integration with an integration time step $h = 0.01$ ms.