Kent Academic Repository

Full text document (pdf)

Citation for published version

Bowman, Howard and Su, Li and Barnard, Philip (2006) Semantic Modulation of Temporal Attention: Distributed Control and Levels of Abstraction in Computational Modelling.   Technical report. UKC

DOI

Link to record in KAR

https://kar.kent.ac.uk/14418/

Document Version

UNSPECIFIED

Copyright & reuse
Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research
The version in the Kent Academic Repository may differ from the final published version.
Users are advised to check **http://kar.kent.ac.uk** for the status of the paper. **Users should always cite the published version of record.**

Enquiries
For any further enquiries regarding the licence status of this document, please contact:
researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at **http://kar.kent.ac.uk/contact.html**

Kent Academic Repository

University of Kent

Computer Science at Kent

Semantic Modulation of Temporal Attention: Distributed Control and Levels of Abstraction in Computational Modelling

H Bowman, Su Li and P J Barnard

Technical Report No. 9-06
September 2006

Copyright © 2006 University of Kent
Published by the Computing Laboratory,
University of Kent, Canterbury, Kent CT2 7NF, UK

Semantic Modulation of Temporal Attention: Distributed Control and Levels of Abstraction in Computational Modelling[1]

Howard Bowman[1], Li Su[1] and Phil Barnard[2]

[1] Centre for Cognitive Neuroscience and Cognitive Systems, University of Kent

[2]MRC Cognition and Brain Sciences Unit, Cambridge

1 Introduction

There are now many different approaches to the computational modelling of cognition, e.g. symbolic models (Kieras & Meyer, 1997; Newell, 1990), cognitive connectionist models (McLeod, Plunkett, & Rolls, 1998) and neurophysiologically prescribed connectionist models (O'Reilly & Munakata, 2000). The relative value of different approaches is a hotly debated topic, with each presented as an alternative to the others; that is, that they are in opposition to one another, e.g. (Fodor & Pylyshyn, 1988; Hinton, 1990). However, an alternative perspective is that these reflect different levels of abstraction / explanation of the same system that are complementary, rather than fundamentally opposed.

Computer science, which has often been used as a metaphor in the cognitive modelling domain, gives a clear precedent for think in terms of multiple views of a single system. In particular, in computer science, single systems are routinely viewed from different perspectives and at different abstraction levels. An illustration of this is what is now probably the most widely used design method, the Unified Modelling Language (UML) (Booch, Rumbaugh, & Jacobson, 1999). This approach incorporates multiple modelling notations, each targeted at a particular system characteristic. Furthermore, UML is not unique in emphasizing a multiple perspective approach; see, for example, viewpoints (H. Bowman & Derrick, 2001; H. Bowman, Steen, Boiten, & Derrick, 2002), aspect oriented programming (Kiczales et al., 1997) and refinement trajectories (H. Bowman & Gomez, 2006; Derrick & Boiten, 2001; Roscoe, 1998).

It is not that this perspective has been completely lost on cognitive scientists; indeed, Marr famously elaborated a version of this position in his three levels of cognitive description (Marr, 2000). However, despite Marr's observations, concrete modelling endeavours rarely, if ever, consider multiple abstraction levels in the same context and particularly how to relate those levels.

[1] Full reference is, "Semantic modulation of temporal attention: Distributed control and levels of abstraction in computational modelling." H.Bowman, Su Li, and P.J. Barnard. Technical Report 9-06, Computing Lab, University of Kent at Canterbury, September 2006.

Computer science, and particular software engineering, boasts a plethora of multiple perspective approaches (H. Bowman & Derrick, 2001; H. Bowman & Gomez, 2006; H. Bowman et al., 2002; Derrick & Boiten, 2001; Kiczales et al., 1997; Roscoe, 1998). In some of these, the multiple perspectives offer different views of the system at a single level of abstraction, e.g. (H. Bowman et al., 2002). However, probably the longest standing and most extensively investigated question, is how to relate descriptions at different levels of abstraction, which, in computer science terms, means different stages in the system development trajectory. For example, two commonly considered abstraction levels are, 1) the requirements level, i.e. the abstract specification of "*what* the system must do"; and 2) the implementation level, i.e. the structurally detailed realisation of "*how* the system does it". Furthermore, there has been much work on how to relate the requirements level to the implementation level, which, in logical metaphors, amounts to demonstrating that the implementation *satisfies* the requirements or, in other words, that the implementation is a *model*[2] *of* the requirements. In addition, this issue has been framed in terms of the notion of refinement, i.e. the process of taking an abstract description of a system and refining it into a concrete implementation (Derrick & Boiten, 2001; Roscoe, 1998).

We would argue that these levels have their analogues in the cognitive modelling domain. In particular, we can distinguish between the following two levels of explanation of a cognitive phenomenon.

1. high-level abstract descriptions of the mathematical characteristics of a pattern of data, e.g. (Stewart, Brown, & Chater, 2005); and
2. low-level detailed models of the internal structure of a cognitive system, e.g. (Dehaene, Kerszberg, & Changeux, 1998).

These two levels of explanation really reflect different capacities to observe systems; that is, the extent to which the system is viewed from outside or inside, i.e., as a black or white box. There are clear pros and cons to these forms of modelling, which we discuss now.

1) *Black-box (Extensionalist) Modelling*. With this approach, the system is viewed as a black-box; that is, no assumptions are made about the internal structure of the system and there is no decomposition at all of the black-box into its constituent components. Thus, the point of reference for the modeller is the externally visible behaviour, e.g. the stimulus-response pattern. In the computer science setting, the analogue of black-box modelling would be requirements specification, where logics are often used to express the global observable behaviour of a system (Manna & Pnueli, 1992).

   A critical benefit of black-box cognitive modelling, is that a minimal set of assumptions are made, especially in respect of the system structure. Consequently, there are less degrees of freedom and fewer hidden assumptions; making data fitting and parameter setting both well founded and, typically, feasible. For example, if the system can be described in closed form, key parameters can be determined by solving a set of equations, if not, computational search methods can be applied.

[2] where we are using the term model in its strict logical sense (Gries & Schneider, 1994).

2) *White-box (Intensionalist) Modelling*. In contrast, with this approach, the system is viewed as a white box; that is, the internal (decompositional) structure of the system is asserted. Although we can bring theories of cognitive architecture and (increasingly) neural structure to bear in proposing white-box models, a spectrum of assumptions (necessarily) need to be made. Furthermore, typically, many of these assumptions concern the internal structure of the system. In the computer science setting, the analogue of white-box modelling would be the system specification, where components and interaction patterns are explicitly described in specification languages such as Z (Woodcock & Davies, 1996), process algebra (H. Bowman & Gomez, 2006; Hoare, 1985; Milner, 1989; Roscoe, 1998) or Statecharts (Harel, 1987). Since the information processing revolution, white-box modelling of cognition using a variety of computational metaphors has been extensively explored, e.g. (Newell, 1990; Rumelhart, McClelland, & the-PDP-Research-Group, 1986). While structurally detailed models of cognition are likely to be the most revealing (especially with the current emphasis on neurophysiological correlates), deduction from these models is more slippery and potentially less well founded. Most importantly, many assumptions, such as settings of key parameters, need to be made, many of which may, at best, require complex justification and, at worst, be effectively arbitrary. As a result, parameter setting and data fitting is more difficult and, arguably, less well founded with white-box models.

We can summarise then by saying that black-box modelling describes *what* a cognitive system does and it describes it in a relatively contained and well-founded manner. However, white-box modelling cannot be ignored, since it enables us to describe *how* a cognitive system functions, which is a concern for both traditional information processing and more recent neurophysiological explanations. Thus, a central research question is how to gain the benefit of contained well-founded modelling in the context of structurally detailed descriptions.

A possible strategy for addressing this question is to take inspiration from the computer science notion of *refinement*. Thus, when tackling the computational modelling of a particular cognitive phenomenon, one should start with an abstract black-box analysis of the observable behaviour arising from the phenomenon. For example, this may amount to a characterisation of the pattern of stimulus-response data. However, importantly, a minimum of assumptions should be made. Then, from this solid foundation, one could develop increasingly refined and concrete models, in a progression towards white-box models. Importantly though, this approach enables cross abstraction level validation, showing, for example, that the white-box model is correctly related to the black-box model, i.e. in computer science terms, is related by refinement.

This paper provides an initial step in the direction of multi-level cognitive modelling. However, although the progressive refinement methodology that we propose will be evident, the research is certainly no more than a first step. In particular, all our models sit somewhere between the black and white-box extremes; that is, pushing the metaphor even further, the refinement we present is more from dark-gray to light-gray! More complete instantiations of our methodological proposal awaits further theoretical work on how to relate the sorts of models developed in the cognitive modelling setting.

We will explore the issue of relating abstraction levels in the context of a particular class of modelling, which emphasises *distributed executive control*. This responds to traditional symbolic cognitive architectures in which control is typically centralised. In contrast, with distributed control, there is no central locus of the data state of the system and the system is composed of a set of computational entities (which we call subsystems) that each have a local data state and local processing capabilities. These subsystems evolve independently subject to interaction / communication between themselves. Thus, importantly, there is a distinction between the local and the global view of the system state; individual subsystems only having direct access to their local state and at no point in time does any thread of control have access to a complete view of the state of the system. The selection of such a distributed view of control in the context of cognitive modelling is justified in (P. J. Barnard & Bowman, 2004; H. Bowman & Barnard, 2001).

In order to obtain models that directly reflect distribution of control, we use a modelling technique called process algebra (H. Bowman & Gomez, 2006; Hoare, 1985; Milner, 1989). These originated in theoretical computer science, being developed to specify and analyse distributed computer systems (H. Bowman & Gomez, 2006). A process algebra specification contains a set of top-level subsystems (called processes in the computing literature) that are connected by a set of (predefined) communication channels. Subsystems interact by exchanging messages along channels. Furthermore, process algebra components can be arbitrarily nested within one another, allowing hierarchical description in the manner advocated in (P. J. Barnard & Bowman, 2004; H. Bowman & Barnard, 2001). Process algebra are an appropriate means to consider multi-level modelling, since they offer a rich theory of refinement and formal relationships between specifications (H. Bowman & Gomez, 2006). Although, in this initial investigation in this area, we will not take much direct benefit from these inter level relationships.

We illustrate our approach in the context of a study of temporal attention. Specifically, we model the temporal characteristics of how meaning captures attention. To do this we reproduce data on the key-distractor attentional blink task (P. J. Barnard, Scott, Taylor, May, & Knightley, 2004), which considers how participants' attention is drawn to a distractor item that is semantically related to a target category. Furthermore, (P. J. Barnard et al., 2004) have shown that the level of salience of the distractor, i.e. how related it is to the target category, modulates how strongly attention is captured. The details of this phenomenon will be discussed in section 2.1.

In the remainder of this article, we will use the term *extensionalist* to describe the black-box approach and *intensionalist* to describe the white-box approach (Milner, 1986).

2 Background

2.1 The Key-distractor Attentional Blink Task

The models we develop make an explicit proposal for how meaning captures attention and particularly the temporal characteristics of such capture. The phenomenon we model sits within a tradition of temporal attention research centred on the attentional blink task. (Raymond, Shapiro, & Arnell, 1992) were the first to use the term Attentional Blink (AB). The task they used to reveal this phenomenon involved letters being presented using Rapid Serial Visual Presentation (RSVP) at around ten items a second. One letter (T1) was presented in a distinct colour and was the target whose identity was to be reported. A second target (T2) followed after a number of intervening items. Typically, participants had to report whether the letter "X" was among the items that followed T1. The key finding was that selection of T2 was impaired with a characteristic serial position curve; see Figure 1. T2s occurring immediately after T1 were accurately detected. Detection then declined across serial-positions 2 (and also usually) 3 and then recovered to baseline around lags 5 or 6 (corresponding to a target onset asynchrony in the order of 500 to 600 ms).

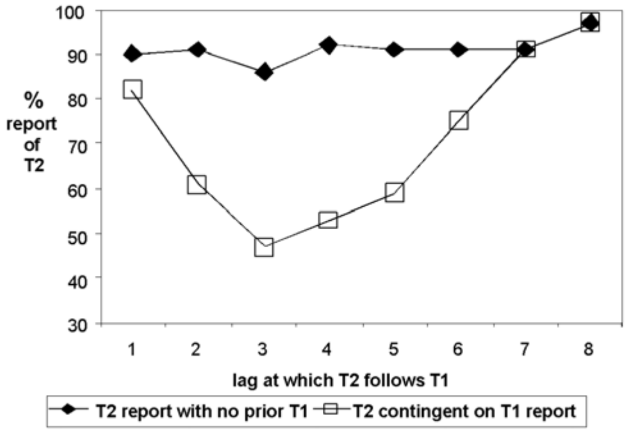

Figure 1: The basic "Attentional Blink" effect for letter stimuli (Raymond et al., 1992). Here, baseline represents a person's ability to report the presence of T2 in the absence of a T1.

As research on the blink and RSVP in general has progressed, it has become evident that the allocation of attention is affected by the meaning of items (Maki, Frigen, & Paulsen, 1997) and their personal salience (K.L. Shapiro, Caldwell, & Sorensen, 1997). There is also evidence from electrophysiological recording that the meaning of a target is processed even when it is not reported (K. L. Shapiro & Luck, 1999). In addition, there are now reports of specific effects of affective variables, e.g. (P. J. Barnard, Ramponi, Battye, & Mackintosh, 2005). In particular, (Anderson, 2005) has shown that the blink is markedly attenuated when the second target is an aversive word.

There are now a number of theoretical explanations and indeed computational models of the AB; see (H. Bowman & Wyble, 2005) for a review. However, apart from the model discussed in (P. J. Barnard & Bowman, 2004), all these proposals seek to explain "basic" blink tasks, in which items in the RSVP stream are semantically primitive, e.g. letters or digits. However, as will become clear shortly, our focus is semantically richer processing. Consequently, none of these previous theories or models is directly applicable to our needs in this article. However, of these previous theories, that introduced by (Chun & Potter, 1995) is most closely related to the model we will propose shortly. Their theory assumes two stages of processing. The first stage performs an initial evaluation to determine “categorical” features of items. This stage is not capacity limited and is subject to rapid forgetting. The second stage builds upon and consolidates the results of the first in order to develop a representation of the target sufficient for subsequent report. This stage is capacity-limited, invokes central conceptual representations and storage, and is only initiated by detection of the target on the first stage. In addition, the recently proposed theory of temporal attention, the Simultaneous Type Serial Token model, takes key inspiration from Chun and Potter's 2-stage model (H. Bowman & Wyble, 2005) in explaining the AB phenomenon.

In order to examine semantic effects in more detail, (P. J. Barnard et al., 2004) used a variant of the AB paradigm in which no perceptual features were present to distinguish targets from background items. In this task, words were presented at fixation in RSVP format. Targets were only distinguishable from background items in terms of their meaning. This variant of the paradigm did not rely on dual target report. Rather, participants were simply asked to report a word if it refers to a job or profession for which people get paid, such as “waitress” and these targets were embedded in a list of background words that all belonged to the same category. In this case, they were inanimate things or phenomena encountered in natural environments; see Figure 2. However, streams also contained a key-distractor item, which, although not in the target category, was semantically related to that category. The serial-position that the target appeared after the key-distractor was varied. We call this the key-distractor AB task.

Participants could report the target word (accurate report), say “Yes” if they were confident a job word had been there but could not say exactly what it was, or say “No” if they did not see a target, and there were, of course, trials on which no target was presented. When key-distractors were household items, a different category from both background and target words, there was little influence on target report. However, key-distractors that referenced a property of a human agent, but not one for which they were paid, like tourist or husband, gave rise to a classic and deep blink, not unlike that already shown in Figure 1.

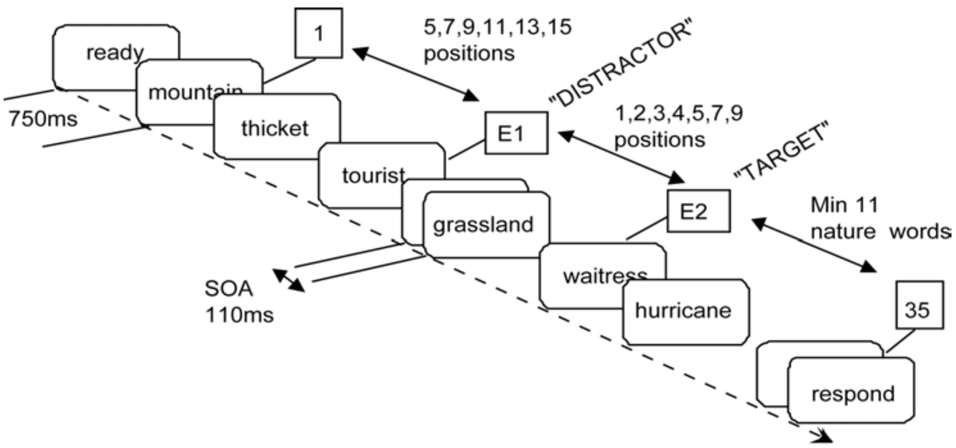

Figure 2: Task schema for the key-distractor blink; adapted from (P. J. Barnard et al., 2004).

(P. J. Barnard et al., 2004) used Latent Semantic Analysis (LSA) (Landauer & Dumais, 1997) to assess similarities between “human” key-distractors and job targets. LSA is a statistical learning method, which uses the co-occurrence of words in texts and principle components analysis to build a multidimensional representation of word meaning. In particular, an "objective measure" of the semantic distance between a pair of words or between a word and a pool of words can be extracted from LSA.

The critical finding of Barnard et al was that the depth of the blink induced by a key-distractor was modulated by the semantic salience of that distractor. That is, using LSA as a metric, the closer the key-distractor was to the target category, the deeper the blink; see Figure 6(b). Thus, the greater the salience of the key-distractor, the greater its capacity to capture attention. Reproducing this modulation of attentional capture by semantic salience is a central objective of this paper.

2.2 Theoretical Foundations

Three principles underlie our models: sequential processing, 2-stages and serial allocation of attention. We discuss these principles in turn.

2.2.1 Sequential Processing

With any RSVP task, items arrive in sequence and need to be correspondingly processed. Thus, we require a basic method for representing this sequential arrival and processing of items. At one level, we can view our approach as implementing a pipeline. New items enter the front of the pipeline (from the visual system), they are then fed through until they reach the back of the pipeline (where they enter working memory). Every cycle, a new item enters the pipeline and all items currently in transit are pushed along one place. We call this the *update cycle*.

The key data structure that implements this pipeline metaphor is a delay-line. This is a simple means for representing time constrained serial order. One can think of a delay-line as an abstraction for items passing (in turn) through a series of processing levels. In this sense, it could be viewed as a symbolic analogue of a sequence of layers in a neural network; a particularly strong analogue being with synfire chains (Abeles, Bergman, Margalis, & Vaadia, 1993).

It is a very natural mechanism to use in order to capture the temporal properties of a blink experiment, which is inherently a time constrained order task. To illustrate the data structure, consider a delay-line of 4 elements, as shown in Figure 3, which records the last 4 time instants of data.

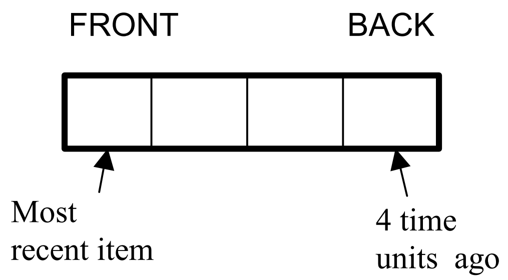

Figure 3: A four item delay line.

The pipeline we will employ in our model will be considerably longer than 4 units and we will not depict it in full here. However, it is worth representing a typical state of a 12 item portion of the overall delay-line during our attentional blink simulations. Figure 4 shows a typical state, where indices indicate the position of the constituent items in the corresponding RSVP item (which will here be a word). We will use this terminology throughout, i.e. a single RSVP item will be represented by a number of constituent (delay line) items, with this number determined by the speed of the delay-line update cycle. In Figure 4, we have assumed 6 constituent items comprise one RSVP item (which is actually consistent with the approach we will adopt in the remainder of the paper and which we will justify shortly).

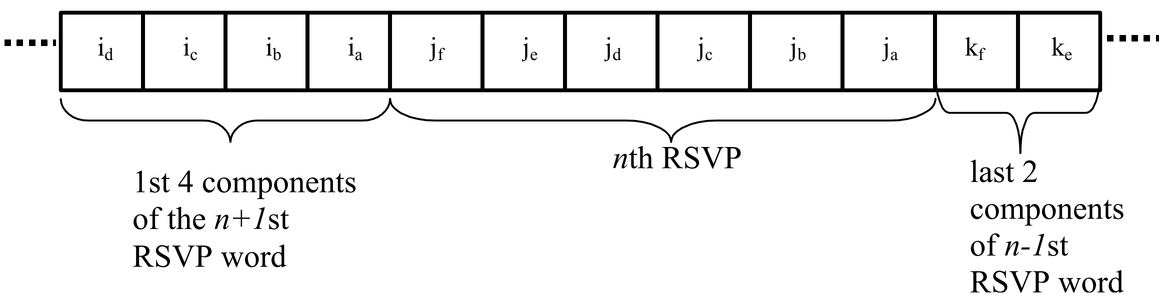

Figure 4: A twelve item delay line with three RSVP items in progress through it.

2.2.2 Two-Stages

Like (Chun & Potter, 1995), (P. J. Barnard et al., 2004) and (P. J. Barnard & Bowman, 2004) argued for a two-stage model, but this time recast to focus exclusively on semantic analysis and executive processing. In particular, (P. J. Barnard & Bowman, 2004) modelled the key-distractor blink task using a two-stage model. In the first stage, a generic level of semantic representation is monitored and initially used to determine if an incoming item is salient in the context of the specified task. If it is found to be so, then, in the second stage, the specific referential meaning of the word is subjected to detailed semantic scrutiny; thus, a word's meaning is actively evaluated in relation to the required referential properties of the target category. If this reveals a match, then the target is encoded for later report. The first of these stages is somewhat akin to first taking a *"glance"* at generic meaning, with the second akin to taking a closer *"look"* at the relationship to the meaning of the target category. These two stages are implemented in two distinct subsystems: the *implicational subsystem* (which supports the first stage) and the *propositional subsystem* (which supports the second) (P. J. Barnard, 1999).

These two subsystems process qualitatively distinct types of meaning. One, implicational meaning, is holistic, abstract and schematic, and is where affect is represented and experienced (P. J. Barnard, 1999). The other is classically "rational", being based upon propositional representation, capturing referentially specific semantic properties and relationships.

As an illustration of implicational meaning, semantic errors make clear that sometimes we only have (referentially non-specific) semantic gist information available to us, e.g. false memories (Roediger & McDermott, 1995) and the Noah illusion (Erickson & Mattson, 1981). In particular, with respect to the latter of these, when comprehending sentences, participants often miss a semantic inconsistency if it does not dramatically conflict with the gist of the sentence, e.g., in a Noah specific sentence, substitution of Moses for Noah often fails to be noticed, while substitution with Nixon is noticed. This is presumably because both Moses and Noah fit the generic (implicational) schema "male biblical figure" (P. J. Barnard et al., 2004), but Nixon does not.

In the context of the task being considered here, these subsystems can be distinguished as follows.

- *Implicational Subsystem*. This performs the broad "categorical" analysis of items considered in Chun and Potter's first stage of processing, by detecting the presence of targets according to their broad categorical features. In the context of this paper, we will call the representations built at this subsystem implicational and we will talk in terms of implicationally salient items, i.e. those that "pass the implicational subsystem test". The implicational subsystem implements the "glance".
- *Propositional Subsystem*. This builds upon the implicational representation generated from the glance in order to construct a full (propositional) identification of the item under consideration, which is sufficient for report. We will describe items that "pass the propositional test" as propositionally salient.

To tie this into the previous section, the implicational and propositional subsystems perform their corresponding salience assessments as items pass through them in the pipeline. We will talk in terms of the *overall delay-line* and *subsystem delay-lines*. The former of which describes the complete end-to-end pipeline, from the visual to the working memory and response subsystems, while the latter is used to describe the portion of the overall pipeline passing through a component subsystem, e.g. the propositional delay-line.

2.2.3 Serial Allocation of Attention

Our third principle is a mechanism of attentional engagement. It is only when attention is engaged at a subsystem that it can assess the salience of items passing through it. Furthermore, attention can only be engaged at one subsystem at a time. Consequently, semantic processes cannot glance at an incoming item, while looking at and scrutinising another. This constraint will play an important role in generating a blink in our models.

When attention is engaged at a subsystem, we say that it is buffered (P. J. Barnard, 1999). Thus, salience assignment can only be performed if the subsystem is buffered and only one subsystem can be buffered at a time. The buffer mechanism ensures that the central attentional resources are allocated serially, while data representations pass concurrently, in the sense that all data representations throughout the overall delay-line are moved on one place on each time step.

2.2.4 Why New Models?

A computational model of the Attentional Blink (AB) was presented in (P. J. Barnard & Bowman, 2004), which produced the basic key-distractor blink curve. We present a sequence of models to extend the previous one, in order to address recent experimental findings and theoretical accounts (P. J. Barnard et al., 2004). In particular, the previous model simply generated a blink curve, however, a central concern of the current investigation is to consider how the depth of the blink curve (and hence the extent of attentional capture) is modulated by the salience of the key-distractor. The first model is an extensionalist model, which minimises the assumptions made about the structure of the system. In contrast, the second model is intensionalist in character, which makes more detailed assumptions about the internal structure of the system.

3 Extensionalist AB Model

3.1 Data Representations

This model works at a high level of theoretical abstraction, e.g. words are expressed by their roles in Barnard et al's blink task: background, target and distractor (P. J. Barnard et al., 2004). This only distinguishes between different word types, i.e. background (*Back*), target (*Targ*) and key-distractor (*Dist*):

$$Word\_tp ::= Back \mid Targ \mid Dist$$

where distractor has two types: high salient (HS) and low salient (LS):

$$Dist ::= HS \mid LS$$

The data representations in the model are defined as follows:

$$Rep ::= (\ Word\_tp, Sal, Sal\ )$$

which associates salience assessments with words in the RSVP stream. The first element in the representation is the word type identity. The second and the third elements are an implicational and a propositional salience assessment respectively. The salience assessments are initially set to un-interpreted.

3.2 Architecture

Similar to the previous model (P. J. Barnard & Bowman, 2004), the current model is also composed of two ICS subsystems, along with source and sink components. As shown in Figure 5, the two ICS subsystems include the implicational subsystem (Imp) and propositional subsystem (Prop). The source and the sink summarise the perceptual subsystems and the maintenance and response subsystems respectively. The source outputs a data representation to Imp every 20ms of simulated time. Barnard et al's experiment consists of 35 words presented at a rate of 110ms per word (P. J. Barnard et al., 2004). Hence, a word in the model contents six data representations, which approximates the 110ms presentation. The data representations are then passed through the two subsystems. Finally, the data representations reach the sink for working memory encoding and later report.

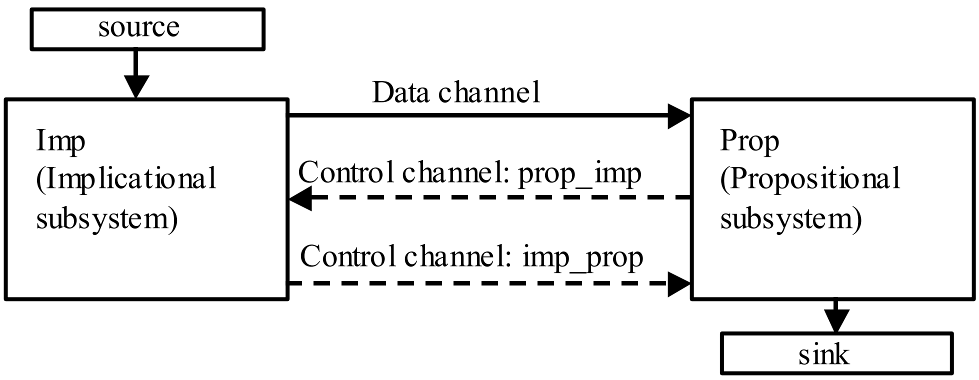

Figure 5 Top-level structure

3.3 Delay Lines

Within each subsystem, there is a single delay line. The delay lines in both subsystems increment by one slot every 20ms. Then the first item of the implicational delay line becomes the most recent data representation input from the source. The last item of the implicational delay line is removed and passed to Prop via the data channel between the two subsystems. The propositional delay line has the same behaviour as the implicational delay line. Consequently, data representations arrive at the sink every 20ms. Both delay lines can hold 10 data representations. This reflects both the memory and the amount of time spent on a subsystem's information transformation.

3.4 Salience Assignment

Each subsystem assigns salience to the data representation entering it. The salience assignment is performed at the delay line of the subsystem in buffered mode. We will define the mode of a subsystem in the relevant section. As we have explained previously, a word is composed of several data representations, e.g. six in the current simulation, the semantic meaning of a word is built up gradually through time. Hence, a subsystem can access the meaning of a word by looking across several data representations that belong to the same word. In most cases, one does not have to process all the data representations in order to assemble the meaning. That is, the meaning of a word usually emerges from the first few representations. In the current model, we assume that the meaning of the word can be obtained after processing the first three data representations. In addition, each subsystem assigns salience to data representations according to predefined probabilities, which will be explained in the relevant sections.

3.5 Buffer Movement

The subsystem that is buffered decides when the buffer moves and where it moves to. Initially, Imp is buffered. It passes the buffer to Prop when it detects an implicationally salient word. A word is seen to be implicationally salient if it has at least three implicationally salient data representations. Then Prop takes a detailed look at the word in order to make sure that it belongs to the target category. Prop can only do so if it is buffered. However, the buffer moves with a delay. It is set to 200ms, which reflects the average time required to relocate the attentional resources. The mean delay was 210ms in the previous model (P. J. Barnard & Bowman, 2004). It will become clear that the delay of buffer movement plays an essential role in generating the blink.

When Prop is buffered and detects an implicationally uninterpreted word, the buffer is passed back to Imp. This is because propositional meaning builds upon coherent detection of implicational meaning. Thus, when faced with an implicationally uninterpreted item, Prop is no longer able to assign salience and the buffer has to return to Imp to assess implicational meaning. Prop needs to look across three data representations in order to know that the implicational meaning is un-interpreted.

In real life situations, stimuli do not arrive as rapidly as in AB experiments, so Imp and Prop will normally interpret the representation of the same word or event for an extended period. However, in laboratory situations, such as RSVP, items may fail to be implicationally processed as the buffer moves between subsystems. The delayed buffer movement dynamic provides the underlying mechanism for the blink, i.e.

- When the buffer moves from Imp to Prop, because the distractor was found implicationally salient, salience assessment cannot be performed on a set of words (i.e. a portion of the RSVP stream) entering Imp following the distractor. Hence, when these implicationally uninterrupted words are passed to Prop, propositional meaning cannot be accessed. If a target word is within these words, it will not be detected as implicationally salient and thus will not be reported.
- There is normally lag-1 sparing in AB experiments, i.e. a target word immediately following the distractor is likely to be reported. This arises in our model because buffer movement takes time, hence, the word immediately

following the distractor may be implicationally interpreted before the buffer moves to Prop.

- When the buffer moves back to Imp, it assigns salience to its data representations again. After this, target words entering the system will be detected as implicationally and propositionally salient and thus will be reported. Hence, the blink recoveries.

3.6 Parameter Setting

We investigate the key parameters used in the simulation in this section. The first set of parameters is the refresh rate of the system. There are two time scales in the model. The first one is how fast the data representations are passed between subsystems, and the second one is how fast each of the subsystems updates its state. We assumes that all subsystems input/output a data representation every 20ms. As a result, all delay lines also increment by one slot every 20ms. This assumption is justified by the observation that underlying neural mechanisms can represent updates around every 20ms (Bond, 1999; Rolls & Stringer, 2001). In addition, since a word is presented for 110ms (P. J. Barnard et al., 2004) , a word is comprised of 6 data representations. The refresh rate of the state in subsystems is every 5ms. This assumption is not constrained by neural biological facts, but by implementation requirements. For example, it has to be less than the refresh rate of data representation. This fine grain of time course allows us to be more discriminating with regard to the temporal properties of the attentional blink. However, a high refresh rate will have implementation costs, in terms of how long simulations take to run.

The second set of parameters is the delay of buffer movement. In the current model, the buffer can move in two directions, i.e. from Imp to Prop and vice versa. So, there are two buffer movement parameters, i.e. D_1 denotes the delay of buffer movement from Imp to Prop and D_2 denotes the delay of buffer movement back in the other direction. In our model, lag-1 sparing sets the lower bound of D_1. In order to report targets that immediately follow distractors, the buffer should actually move no sooner than 120ms after the time when Imp determines that the buffer needs to move. (This is the time when the first three data representations of the lag-1 item have been processed at Imp, given that each item/word contains 6 data representations, data representations are passed every 20ms and Imp can make a decision about salience once it has seen three data representations.) Furthermore, the onset of the blink sets the upper bound of D_1. In order to miss lag-2 targets, the buffer should move no later than 220ms after the time when Imp determines that the buffer needs to move. (This is the time when the first two data representations of the lag-2 item have entered Imp.) It can be seen that lag-1 sparing and the sharp onset of the blink set the range of values D_1 can be selected from. The recovery of the blink is a function of both delay parameters and the length of the delay line. (The length of the delay line is set by D_1. This will be discussed later.) However, from the slow recovery of the human blink data, we can see that D_2 is less constrained. For the sake of simplicity, we assume that the buffer moves in both direction at the same speed. This assumption reflects the symmetry in the relocation of attentional resources. (However, in the relevant sections, it can be seen how this simple assumption fails and how we solve it.) In summary, we can write the following (in)equations.

$$120ms < D_1 < 220ms$$
$$D_1 = D_2$$

However, this is not enough to know where to set D_1 and D_2 within the 120ms to 220ms range. A relevant issue here is the extent to which lag-1 and lag-2 items are processed. Although we will not explain exactly how our model realises the sort of partial awareness that arises at these lags until the relevant sections, the human data (P. J. Barnard et al., 2004) already suggests that targets at lag-2 are processed to the extend that subjects are aware of the presence of the target words. This suggests that the first part of the lag-2 item is implicationally processed before the buffer moves away. Hence, D_1 is likely to be closer to 220ms than 120ms. As a result, we assume that D_1 and D_1 are 200ms, which is approximately the length of one and a half words. Note, in this extensionalist model, we do not add noise to parameters, however, the later intensionalist model will add noise to the setting of D_1 and D_2.

The third set of parameters is the length of the delay lines, which is set by D_1. We denote the length of the implicational delay lines by L_1, which is measured by the number of data representations it holds. It also determines how long it takes for a data representation to travel through it. Given that a data representation reflects 20ms of information and each subsystem looks across three data representations on the delay line, the following inequation ensures that the buffer moves to Prop in time to catch 1) an item that initiates buffer movement at Imp and 2) a lag-1 target.

$$D_1 < (L_1 + L_w - 3) \times 20ms$$

where L_w denotes the length of a word, which is measured by the number of data representations it is composed of. In the current model, $L_w = 6$. So, the right hand side of the inequation becomes $(L_1 + 3) \times 20ms$. This is the time from when Imp determines that the buffer needs to move, to when the last three data representations of a lag-1 target have entered Prop. Rearranging the above inequation, we obtain the following.

$$L_1 > D_1 \div 20ms - 3 \text{, so}$$
$$L_1 > 7 \text{, given } D_1 = 200ms.$$

The recovery of the blink sets the upper bound of L_1. Given that performance recovers by around lag-5 and the buffer moves back to Imp when the lag-2 item enters Prop. L_1 is generally constrained by the following inequations, which ensure that the buffer moves back to Imp before the lag-5 item enters Imp.

$$D_2 < (4 \times 6 - L_1) \times 20ms \text{, so,}$$
$$L_1 < 14 \text{, given } D_2 = 200ms.$$

In summary, $7 < L_1 < 14$. Hence, we assume that the length of the delay line is 10 data representations. The length of the propositional delay line L_2 is less constrained in this model. By symmetry again, we assume that $L_1 = L_2 = 10$.

3.7 Raw Results

An epoch denotes 9 simulation trials. Each trial within an epoch has the same parameter setting, except for the serial position of targets relative to distractors. On one hand, the model will produce a blink if it detects an implicationally salient distractor. The simulation result of one epoch is shown in Figure 6(a) if an implicationally salient distractor is detected. On the other hand, the simulation result is a flat line of 100% correct report if the distractor is missed. However, humans perceive information in a noisy environment, i.e. salient items may be missed by humans. In the current model, we assume that Prop is perfect in distinguishing targets and non-targets and the likelihood of detecting an implicationally salient item determines the depth of the blink curve. The simulation results show that the widths of all blink curves are the same. This is because the width of the blink is a function of the length of the delay line, the duration of the word presentation, and the total delay of the implicational and propositional buffer movements. These parameters are constants in this model.

There are different likelihoods of detecting implicationally salient items. The first one is the probability of judging targets to be implicationally salient $P_{imp}(Targ)$, which is set by the baseline performance of human subjects. Barnard et al, have reported that humans correctly report the target's identity on average on 67% of trials with no distractor. At high lags the blink curve also recovers to this baseline performance (P. J. Barnard et al., 2004). The second one is the probability of detecting a background word as implicationally salient $P_{imp}(Back)$, which is assumed to be 0. The third one is the probability of detecting a distractor as implicationally salient $P_{imp}(Dist)$, which becomes $P_{imp}(HS)$ in the high salient condition or $P_{imp}(LS)$ in the low salient condition. According to our model, the deepest point in the blink curve reflects the joint probability of missing the distractor and detecting the target $P_{imp}(\neg Dist \wedge Targ)$.[3] We can then obtain the probabilities $P_{imp}(\neg HS \wedge Targ)$ for both high and $P_{imp}(\neg LS \wedge Targ)$ for low salience conditions based on the blink curves in Figure 6(b). In summary, we have obtained the following:

$$P_{imp}(Targ) = 0.67$$
$$P_{imp}(Back) = 0$$
$$P_{imp}(\neg HS \wedge Targ) = 0.34$$
$$P_{imp}(\neg LS \wedge Targ) = 0.54$$

Detecting targets and distractors are two independent events. So,

[3] Note, this is the only way that a target can be detected at the deepest point of the blink. That is, if a distractor is detected as implicationally salient, a target at the deepest point of the blink will be missed.

$$P_{imp}(\neg Dist \wedge Targ) = P_{imp}(\neg Dist) \times P_{imp}(Targg) = (1 - P_{imp}(Dist)) \times P_{imp}(Targ)$$

$$= \begin{cases} (1 - P_{imp}(HS)) \times P_{imp}(Targ) = 0.34 & \textit{in the high salient condition} \\ (1 - P_{imp}(LS)) \times P_{imp}(Targ) = 0.54 & \textit{in the low salient condition} \end{cases}$$

Given $P_{imp}(Targ) = 0.67$, we can obtain the following:

$(1 - P_{imp}(HS)) \times 0.67 = 0.34$ and $(1 - P_{imp}(LS)) \times 0.67 = 0.54$, hence,
$P_{imp}(HS) = 1 - 0.34 \div 0.67 = 0.49$ and $P_{imp}(LS) = 1 - 0.54 \div 0.67 = 0.19$.

This calculation determines how the model behaves and how epochs are combined to generate a blink curve. The results of the simulation contain three types of epochs:

- A flat line at 100%, which corresponds to the model missing the distractor and detecting the target. The percentage of this occurring is:

$$\begin{aligned} & perc(1) \\ & = P_{imp}(\neg Dist) \times P_{imp}(Targ) \times P_{prop}(Targ) \\ & = P_{imp}(\neg Dist \wedge Targ) \times P_{prop}(Targ) \\ & = P_{imp}(\neg Dist \wedge Targ) \times 1 \\ & = P_{imp}(\neg Dist \wedge Targ) \end{aligned}$$

- A flat line at 0%, which corresponds to missing the target. The percentage of this occurring is:

$$\begin{aligned} & perc(0) \\ & = P_{imp}(Dist \wedge \neg Targ) + P_{imp}(\neg Dist \wedge \neg Targ) \\ & = P_{imp}(\neg Targ) \end{aligned}$$

- A blink curve as shown in Figure 6(a), which corresponds to detecting both the distractor and the target. The percentage of this occurring is:

$$\begin{aligned} & perc(blink) \\ & = P_{imp}(Dist) \times P_{imp}(Targ) \times P_{prop}(Targ) \\ & = P_{imp}(Dist \wedge Targ) \times 1 \\ & = P_{imp}(Dist \wedge Targ) \end{aligned}$$

These three situations cover all the possible results in the simulation, accordingly their probabilities sum to 100, i.e.

$$\begin{aligned} & perc(1) + perc(0) + perc(blink) \\ &= P_{imp}(\neg Dist \wedge Targ) + P_{imp}(\neg Targ) + P_{imp}(Dist \wedge Targ) \\ &= 100\% \end{aligned}$$

In the high salience condition:

$$perc(1) = P_{imp}(\neg Dist \wedge Targ) = P_{imp}(\neg HS \wedge Targ) = 34\%,$$
$$perc(0) = P_{imp}(\neg Targ) = 100\% - 67\% = 33\%,$$
$$\text{and } perc(blink) = P_{imp}(Dist \wedge Targ) = 100\% - 34\% - 33\% = 33\%$$

In the low salience condition:

$$perc(1) = P_{imp}(\neg Dist \wedge Targ) = P_{imp}(\neg LS \wedge Targ) = 54\%,$$
$$perc(0) = P_{imp}(\neg Targ) = 100\% - 67\% = 33\%,$$
$$\text{and } perc(blink) = P_{imp}(Dist \wedge Targ) = 100\% - 54\% - 33\% = 13\%$$

Then, according to these proportions, we average across these different outcomes across all epochs for all conditions. We obtain the curves shown in Figure 6(c,d) for high and low salience conditions respectively.

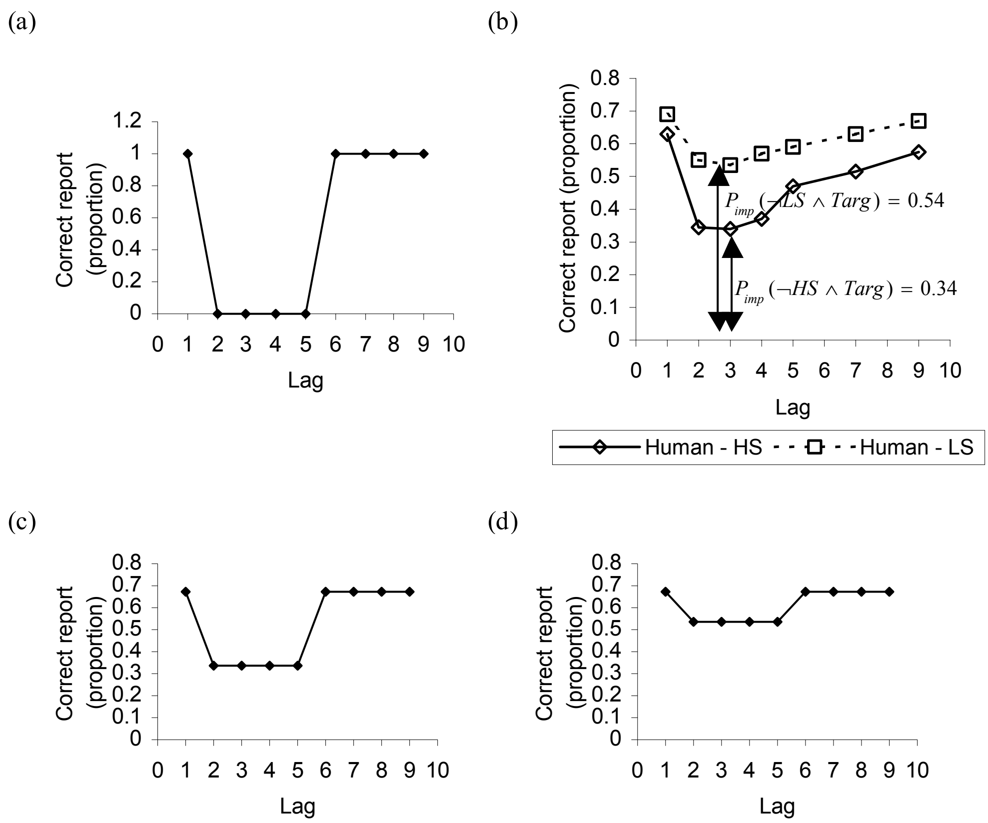

Figure 6 T2 accuracy by lag from extensionalist model and humans. (a) Simulation results from a single epoch. (b) T2 accuracy by lag in humans for high salience and low salience key distractors. (c) Average across epochs – high salience condition. (d) Average across epochs – low salience condition.

3.8 Adding Noise

In the previous model (P. J. Barnard & Bowman, 2004), noise was added to the delay of the buffer movement in order to reflect individual differences and variance in performance within individuals. Different amounts of noise were added at different stages of processing, i.e. the delay of buffer movement from Imp to Prop is less noisy than the buffer movement in the opposite direction. This produces a sharp blink onset and a shallow blink recovery. Our extensionalist modelling abstracts from this level of explanation. In the current model, the simulation results are averaged across multiple epochs and then convolved with a noise function. As a reflection of the high level of abstraction of this model, randomness is imposed globally and externally. This technique does not require specification of either the source or the dynamics of noise inside the model. As a result, assumptions about the internal structure of the system are minimised and also the number of simulation runs is reduced.

O'Reilly and Munakata use a similar approach to adding noise in their PDP++ simulation (O'Reilly & Munakata, 2000). They derive a noisy rate coded activation function by convolving a Gaussian-distributed noise (GDN) function with a noise free output function. The resulting function approximates the average effect of a population of neurons, whose spike timing is subject to random variation (Shadlen & Newsome, 1994). The convolution in our model has the following form:

$$F(t) = f \otimes g = \int g(\tau) f(\tau - t) d\tau ,$$

where $f(t)$ denotes a noise function and $g(t)$ denotes the averaged raw simulation results. Note that this function also extends the raw results into the pre-lag-1 and post-lag-9 areas. We assume that the percentage of correct report of the target in these areas is the baseline performance 67%. The reason for this is that, in the pre-lag-1 area, the targets come before the distractors. In the post-lag-9 area, there is a large time gap between the targets and the distractors. Hence, in both cases, the processing of the targets will not be affected by the processing of the distractors.

Inspired by O'Reilly and Munakata's approach, we also use a GDN function. The idea is that the performance of lag-N is more similar to its adjacent lags, e.g. lag-(N-1) or lag-(N+1), than more distant lags. Figure 7(a) compares the simulation results of the convolution using a GDN function with $var = 0.4$ (*var* denotes the standard deviation) and the experimental results of human subjects (P. J. Barnard et al., 2004). As is shown in Figure 7(a), a convolution with a single GDN function with $var = 0.4$ gives reduced lag-1 performance, and almost the same steepness of blink onset as recovery. It does not fit the behavioural data (P. J. Barnard et al., 2004). Although we only show the effect of one GDN function, other GDN functions (with different settings of standard deviation) do not work better than this. One reason for this is that we have made inaccurate assumptions about $D_1 = D_2$ in the previous section. As we have explained previously, the buffer movement delay sets the steepness of the blink and the assumption of $D_1 = D_2$ ensures that the onset and recovery of the blink have the same steepness.

The solution requires us to adjust D_2. However, the nature of extensionalist modelling prevents us from making more assumptions about the variation of D_2 inside the model. Hence, one way to improve the simulation results is to gradually increase the deviation of the GDN function by serial position, i.e. the GDN function is narrower at the earlier lags and broader at the later lags. After a simple search through the parameter space, we chose the nine GDN functions shown in Figure 7(b). At lag-1, the standard deviation of the GDN is 0.14 and it increases by 0.15 at each lag. We call this approach a convolution with sliding noise. This approach is not only consistent with the idea used in the previous model (P. J. Barnard & Bowman, 2004), but also preserve the extensionalist modelling approach. In the relevant section, we will introduce an intensionalist approach to this issue. The intuition behind this approach is that there is less noise in the earlier stage processing than in the later stage processing, which influence the blink onset and recover respectively. The extensionalist approach to noise used here also has the benefit of being very general; in particular, with respect to the randomness of buffer movement delay used in (P. J. Barnard & Bowman, 2004). As is shown in Figure 7(c), the convolution can smooth the original simulation results. The dotted lines are the results of convolutions using nine GDN functions individually. The solid line applies the convolution with sliding noise, i.e. different GDN functions at different serial positions. This results in a sharp blink onset and a shallow blink recovery. In the next section, we will investigate how this model matches the human data.

The convolution technique reduces the performance at lag-1 and high lags below 100%. This is because every data point in the result is affected by all other data points in the blink curve. We should also note that the performance at lag-1 is better than high lags (post recovery). This is because the convolution with sliding noise ensures the earlier lags are less influenced by distant lags.

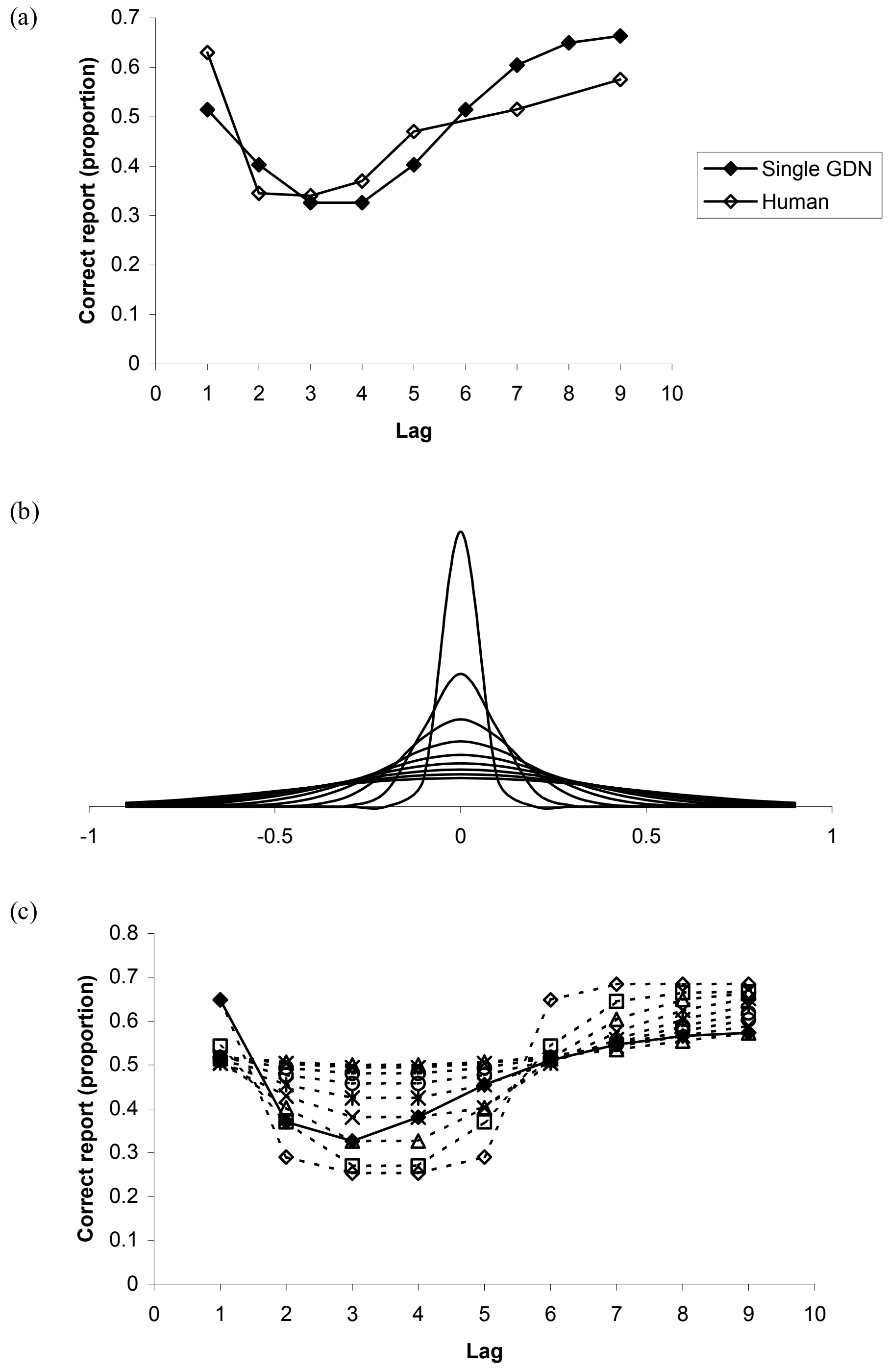

Figure 7 (a) Convolution using a single GDN ($var = 0.4$) vs. Human data in the high salient condition. (b) Nine GDN functions. (c) Convolution with sliding noise in the high salient condition.

3.9 Awareness, Partial Processing and Three Types of Responses

Barnard et al, use three types of responses: report of the target identity, “no job seen” and “yes, I saw a job, but could not report its identity” (P. J. Barnard et al., 2004). This reflects different degrees of awareness of the target presence. In our simulations, the meaning of a target word can also be processed to three different degrees. We argue that different degrees of processing result in different types of response.

Implicational salience	*Propositional salience*	*Responses*
Fully interpreted (salient)	Fully interpreted (salient)	Correct report of identity
Fully un-interpreted	Fully un-interpreted	"No" responses
Partially interpreted	Partially interpreted	"Yes" responses

Table 1 Different degrees of processing of meaning and their corresponding responses (targets)

As is shown in Table 1, both implicationally and propositionally fully interpreted words can be reported correctly with their identity. In the current model, being fully interpreted means that at least three data representations have been processed. This situation is shown as the solid line in Figure 8, these words can occur at positions 1, 6, 7, 8 and 9. In the second situation, some target words can be implicationally fully un-interpreted because Imp is not buffered. In this model, Prop is only able to access the propositional meaning for implicationally interpreted representations. Hence, these words will also be propositionally fully un-interpreted. This demonstrates the situation where subjects are completely unaware of the presence of target words, i.e. the "no" responses. These words can occur at positions 3 and 4, as shown as the dashed line in Figure 8. Finally, some target words can be partially processed. This means less than three of the data representations of the target word have been both implicationally and propositionally processed; shown as the dotted line in Figure 8. Target words at position 2 and 5 are partially interpreted. This arises since distractors can trigger the buffer to move. It moves with a delay of 200ms, which means it moves at the time point when the first data representation of the second word following the distractor has just been processed. Then Prop partially assigns propositional meaning only according to one interpreted data representation. Similarly, only the last data representation of the fifth word following the distractor is processed. This simulates that subjects are aware of the presence of the target word due to partially interpreted meaning; however, its identity cannot be reported due to lack of information. Hence, they will use the "yes" response. Note, other types of partially interpreted words do not occur in the model since buffer movement dynamics ensure that an implicationally interpreted data representation will always be propositionally interpreted as well.

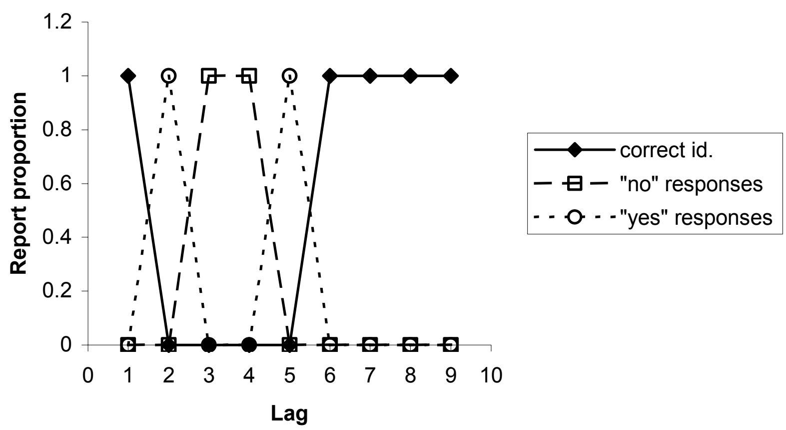

Figure 8 Proportion of three types of response from Simulation (single epoch in the blink condition)

We adjust the simulation results shown in Figure 8 according to the probabilities of detecting different types of words, and apply the sliding noise convolution technique. It was reported by Barnard et al, that correct report of no job seen averaged 85% when no target occurred in the stream. Hence, 15% of the responces are "yes" responses, since they could not report the identities of targets. This is also reflected at the high lags, where "yes" responses return to around 10%. We combine epochs for "yes" and "no" responses in the following way.

- For "yes" responses: 15% of epochs are flat lines at 100%, 49% ($P_{imp}(HS) = 0.49$) in high salient condition and 19% ($P_{imp}(LS) = 0.19$) in low salient condition of epochs are the dotted curve shown in Figure 8, the rest of the epochs are flat lines at 0%.

- For "no" responses: 18% of epochs are flat lines at 100%, since the sum of the baseline performance of correct report of identities, the baseline performance of "yes" responses and the baseline performance of "no" responses is 1 ($100\% - 67\% - 15\% = 18\%$), 49% ($P_{imp}(HS) = 0.49$) in high salient condition and 19% ($P_{imp}(LS) = 0.19$) in low salient condition of epochs are the dashed curve shown in Figure 8, the rest of the epochs are flat lines at 0%.

The resulting percentages of correct report of target identities, "no" responses and "yes" responses are shown in Figure 9. These graphs also illustrate the difference in performance between the high and the low salience conditions. The results are consistent with the experimental results (P. J. Barnard et al., 2004) shown in the same graph. Note, the model does not completely fit the behavioural data at positions 4 and 5, but, broadly speaking, it makes an excellent qualitative, and indeed quantitative fit to the human data. In particular, the capacity of the model to reproduce modulations in blink curve depth by distractor salience is the key effect that our model has been targeted at reproducing.

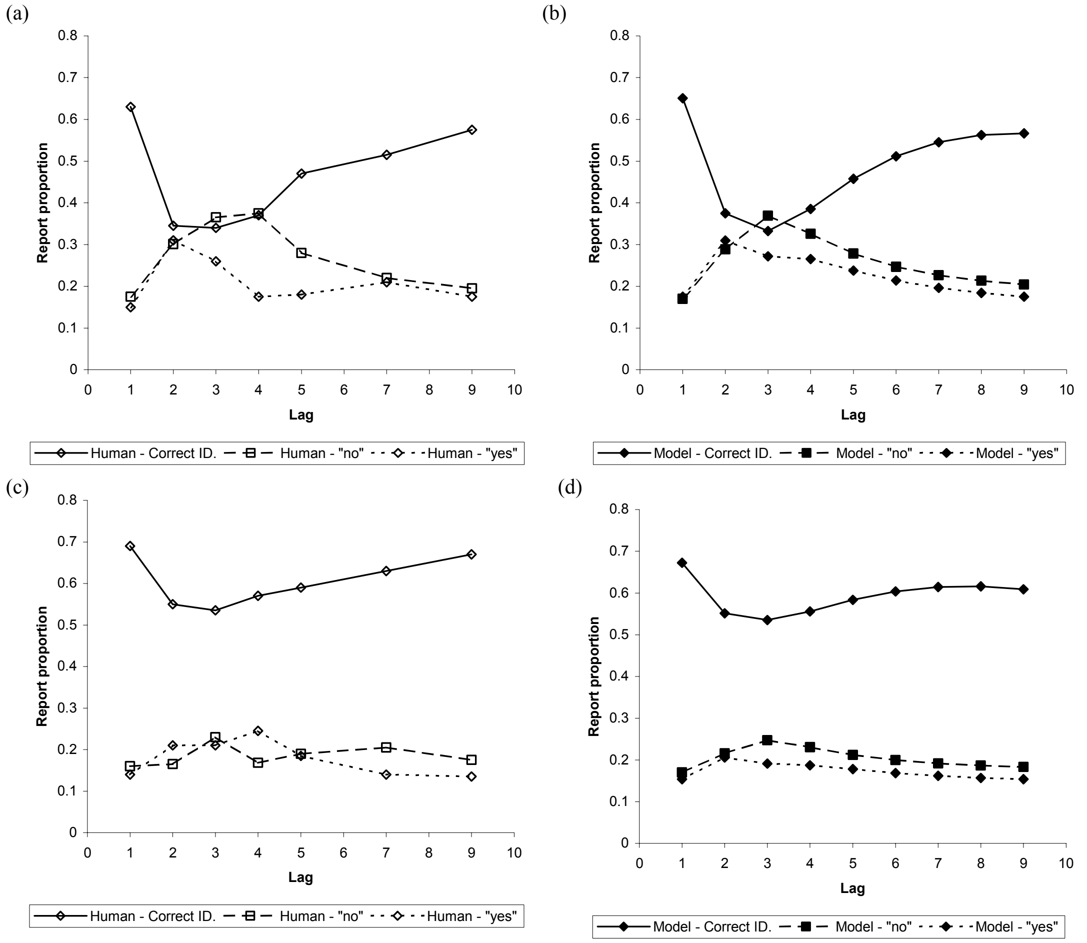

Figure 9 (a) Human performance in high salient condition, (b) simulation results in high salient condition, (c) human performance in low salient condition, (d) simulation results in low salient condition.

4 Intensionalist AB Models

As explained previously, the extentionalist model works at a high level of theoretical abstraction. In contrast, the intensionalist model works at a lower level of abstraction. We will present two approaches to decomposing the extensionalist model: 1) modelling the meaning of individual words and assigning salience according to that meaning; 2) modelling noise inside the model rather than through the extensionalist convolution approach. Finally, we combine these two approaches and present our full intensionalist model.

4.1 Approach 1: Modelling Meaning

4.1.1 Data Representations

Words are individually modelled, with word meanings represented using Latent Semantic Analysis or LSA (Landauer & Dumais, 1997), which will be explained in relevant sections. However, the set of words can be defined as follows:

Word ::= {word1, word2, word3, ……}

An example of an RSVP stream is *(island, ticket, tourist, television, waitress, ……)*, where each item in the stream is of type *Word*, i.e. equal to *wordi* for some *i*. A set of measurements can be associated with each individual word. Each measurement is a function, which returns a real value. These are obtained from LSA and describe the semantic distance between a word and a set of words, i.e. LSA cosines. In our model, we measure the distance between individual words and a pool of words that all belong to a certain category, i.e.

measure(Word,Pool1) : real
measure(Word,Pool2) : real
……

The data representation in the current model is almost the same as the previous model, i.e.

Rep ::= (Word, Sal, Sal)

4.1.2 Architecture and Delay Lines

The structure of this model and the delay lines are the same as the previous model.

4.1.3 Salience Assignment

In the previous section, we have explained that a data representation is set to be both implicationally and propositionally undefined when it initially enters Imp. In our extensionalist model, each subsystem assigns salience to data representations according to predefined probabilities. In this intensionalist model, we hypothesize that a word is assigned to be salient if the semantic distance (LSA cosine) between the word and the target category is smaller than a specified threshold. The implicational threshold is bigger than the propositional threshold as shown in Figure 10. This realises the theory that Imp takes a "glance" at the generic meaning and Prop takes a closer "look" at the meaning in order to determine word identity (P. J. Barnard & Bowman, 2004).

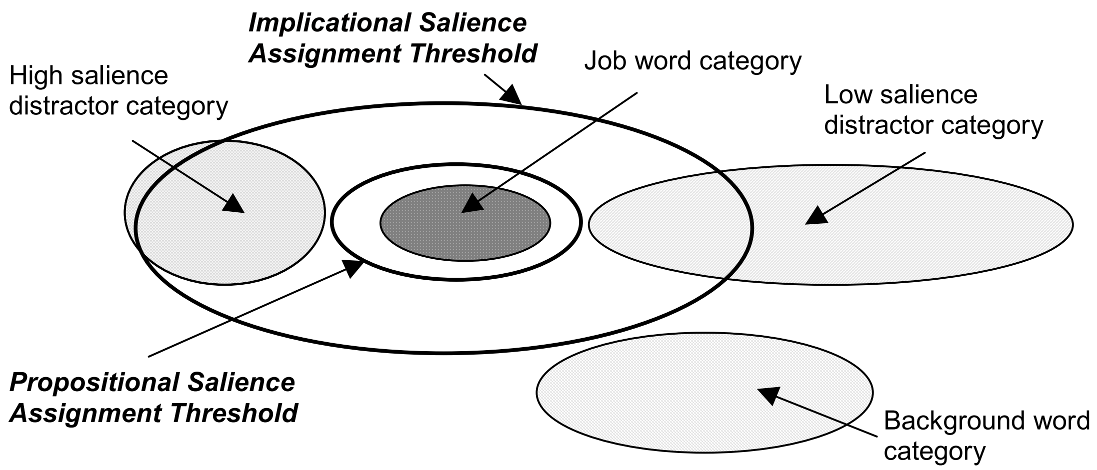

Figure 10 Salience assignments. Semantics in LSA are expressed in a high dimensional space. This illustration is restricted to 2D for ease of depiction.

In Figure 10, there are two spaces, i.e. the implicational salience space and the propositional salience space. The propositional salience space is a subset of the implicational salience space. As a result, these two spaces define four types of word, i.e. type P, type I, type J and type U.

- The area defined by the propositional threshold is called the propositional salience space.
- The area defined by the implicational threshold is called the implicational salience space.
- Words in the propositional salience space are type P words.
- Words in the implicational salience space are type I words.
- Words not in the implicational salience space are type U words, which are both implicationally and propositionally unsalient.
- Words in the implicational salience space, but not in the propositional salience space, are type J words, which are implicationally salient but propositionally unsalient.

The target word (i.e. job word) category is within the propositional salience space. Hence, they are both implicationally and propositionally salient. On the other hand, background word categories are disjoint from the implicational salience space. Hence, they are both implicationally and propositionally unsalient. Distractors can be of two types: type J and type U. It will be clear later that only target job words can be reported and only type J distractors may produce a blink. Note, this graph is an idealised illustration of the semantic space. It describes only one possible configuration of these word categories, by which the cognitive system may produce the desired AB phenomenon. However, it is not true to think that humans actually perceive words in these ways. Indeed, the human system would be much more complicated and noisy than this idealised conceptualisation. In the relevant sections, we will discuss how to approximate this idealised perspective using more realistic and constrained materials.

4.1.4 Buffer Movement, Parameter Setting

The buffer movement and parameter setting in this model are the same as the previous model.

4.1.5 Raw Results

In the current model, the semantic similarity expressed in the distance between the distractor and the target word's category determines the depth of the blink curve. The result of one epoch can also be one of the following three types according to whether the distractor and the target are implicationally salient.

- A blink curve as shown in Figure 11(a), which corresponds to epochs that have type J distractors and type P targets;
- A flat line at 100% (no blink) as shown in Figure 11(b), which corresponds to epochs that have type U distractors and type P targets;
- A flat line at 0% as shown in Figure 11(c), which corresponds to epochs in which the target is implicationally unsalient, i.e. type U targets. This does not occur in the idealised setting, however, it may occur in the real LSA. These epochs determine the baseline performance.

Note, we do not show type J targets here, since they are related to partial responses, which will be explained in the relevant sections. As we have discussed in previous sections, the widths of all blink curves are the same. A distractor category contains both J and U type words. Hence, simulation results of words in the distractor category can be either step functions or flat lines. We need to average the single epoch result of all distractors. This results in an intermediate level of the bottom of the curve. The level depends on the proportion of type J words in the distractor category, i.e. the more type J words, the lower the level, hence the deeper the blink. As shown in Figure 10, the high salient distractors category has more type J words than the low salient distractors category, so the blink curve shown in Figure 11(d) will be deeper than the curve shown in Figure 11(e). Note, Figure 11 only qualitatively illustrates that the semantic space affects the depth of the blink curves. However, in the next section, we will investigate the quantitative relationship between the semantic space and blink depth.

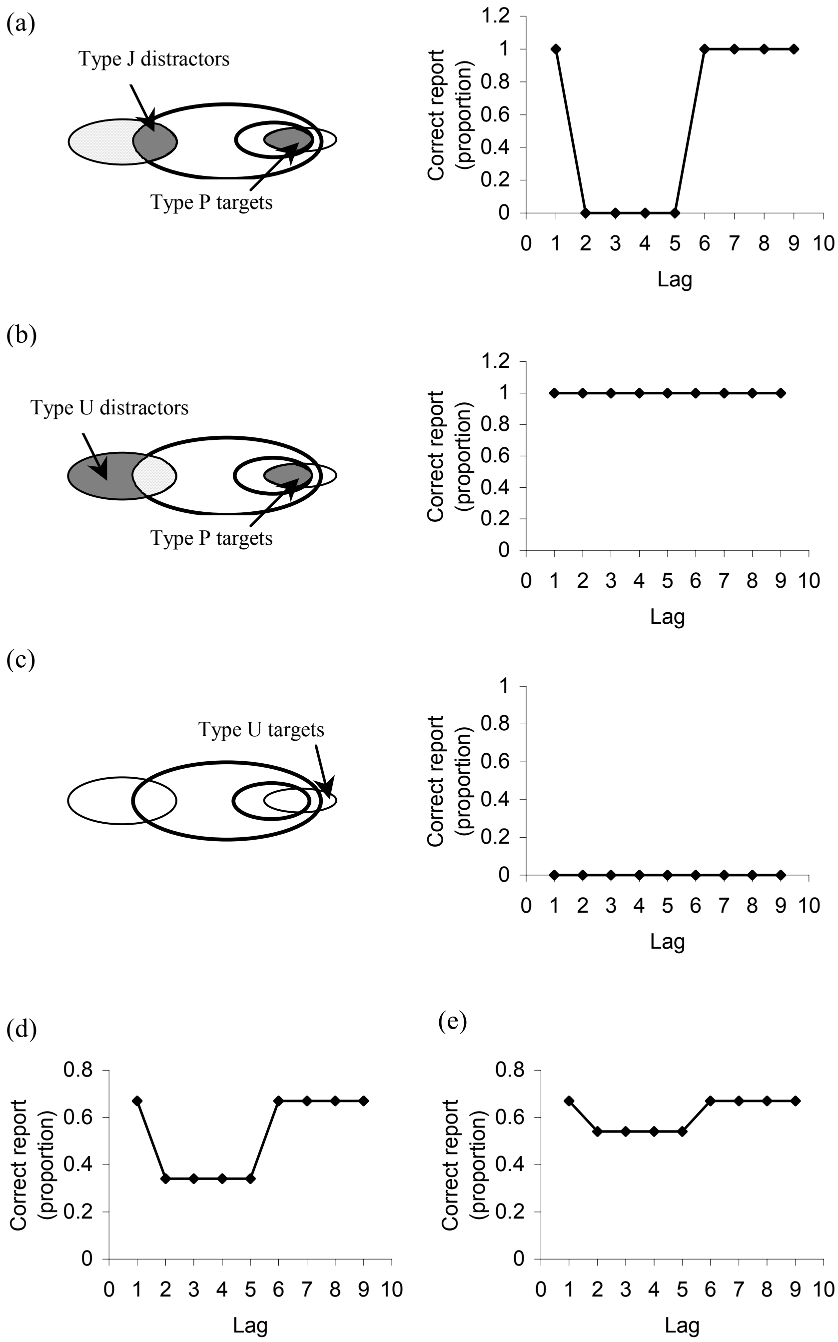

Figure 11 Simulation results from single epochs: (a) Epochs with type J distractors and type P targets. (b) Epochs with type U distractors and type P targets. (c) Epochs with type U targets. (d) Average – high salient condition. (e) Average – low salient condition.

4.1.6 Adding Noise

We also use the convolution technique to add noise in this model.

4.1.7 LSA Calculations and Semantic Space

We have hypothesized the idealised semantic space (Figure 10) and a model based on this space in the previous sections. However, we have not yet explained how this space is derived from real LSA calculations. In this section, we will explore suitable constructions of the semantic space with respect to the comparison of simulation results with human performance. The comparison will also serve as a validation of the ICS account of the AB phenomenon and our model. As we have explained in the previous sections, implicational meaning is vital in determining the depth of the blink. It is also responsible for the difference in blink depth between HS and LS conditions. Hence, we concentrate on implicational meaning in this section and assume that Prop is perfect in distinguishing target jobs from distractors and background words.

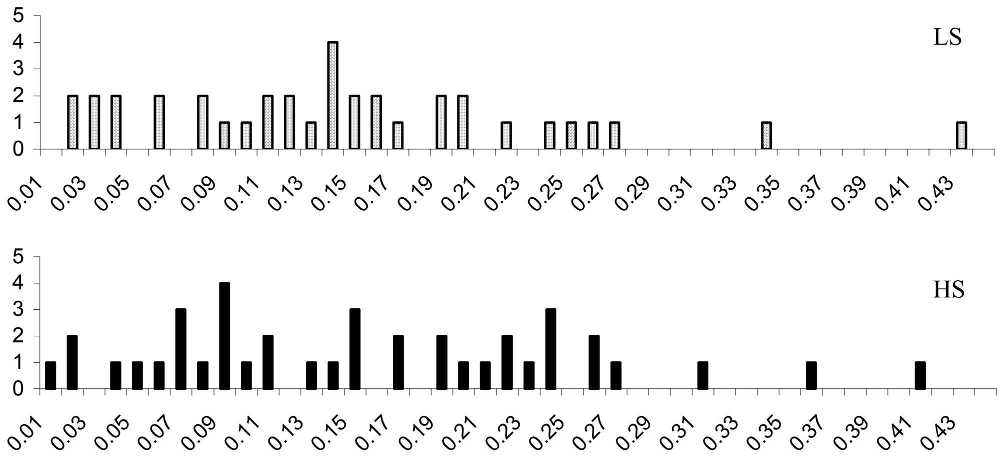

Figure 12 Distribution of HS and LS distractors in the semantic space based on the LSA cosines between distractors and the task instructions. X-axis denotes LSA-cosines, and y-axis denotes frequency, i.e. number of words with that LSA value.

In (P. J. Barnard et al., 2004), one of the LSA analyses of the distractor words calculates the cosine between each distractor and the task instructions. As an initial step, we use this calculation to form our semantic space. In our model, the depth of the blink curve depends on the percentage of distractors above and below the implicational threshold. Figure 12 is a histogram that shows how HS and LS distractors are distributed in this semantic space. We can place the implicational threshold at 0.15. As a result, 52.5% of HS distractors are type J words and 44.4% of LS distractors are type J words. Moreover, 60.9% of targets are type P words (also type J words by default) and 39.1% of targets are type U words. In theory, the model would produce blink curves as shown in Figure 13 after applying the convolution. It can be seen that this does not match the human performance. Furthermore, a simple search of these distributions demonstrates that placing the implicational threshold at alternative values will not improve the simulation results.

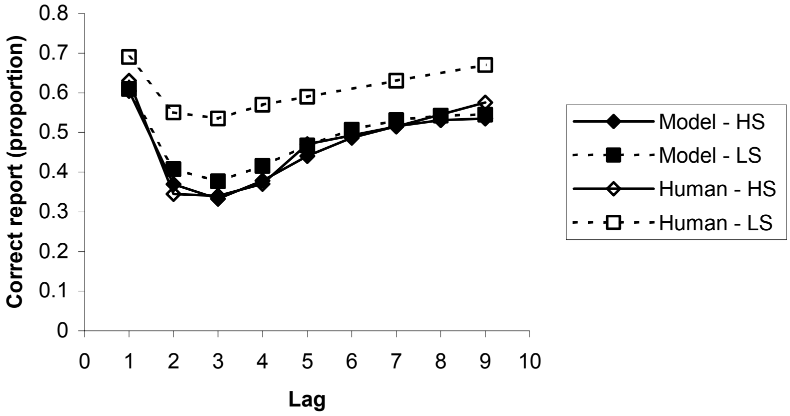

Figure 13 Percentage of correct report of target identity across serial positions after distractor

The reason why the model does not reproduce the human data is because the two distributions are too similar. In particular, broadly speaking, the commonest cosines are the same in both distributions. Thus, no single threshold can be found by which a large proportion of the HS distribution is above it, but a much smaller proportion of the LS distribution is above it. This is reflected in the distributions of the LSA cosines of HS and LS distractors in relation to task instruction not being statistically different, i.e. T-test value between the two distributions in Figure 13is 0.597. This suggests that the relation to task instructions is not a suitable measurement for interpreting implicational meaning in this model.

In response to the failure of this simple approach, we explore more sophisticated analysis of the semantic meaning derived from LSA. In this analysis, we calculated the LSA cosine in relation to different generic word categories (see Appendix A), which are likely to contribute to implicational meaning (P. J. Barnard et al., 2004). Generic words (pools) denote a set of words that describe the generic meaning of a word category. As shown in Table 2, only human relatedness and household relatedness are statistically significantly different between the HS and LS distractors. Hence, we chose the most informative calculation, i.e. the human relatedness, to construct a semantic space and see whether the model can reproduce the human data.

	HS Average	LS Average	HS v LS T-test
Human	0.166	0.103	< 0.001
Occupation	0.092	0.073	0.24
Payment	0.029	0.016	0.1
Household	0.081	0.139	0.01
Nature fillers*	0.079	0.077	0.88
Target Jobs*	0.157	0.146	0.6

Table 2 Cosines to Generic Pools. * These two pools are not generic pools, but are related to the AB task.

We plotted the same histogram as Figure 13 using human relatedness. (Not shown in this paper.) The implicational threshold can also be placed at 0.15. This time it will give enough separation between the HS and LS conditions (21/40 vs. 6/36), and the model, in theory, will produce a similar difference in performance as humans. Figure 14 shows how many words are implicationally salient for HS, LS, nature and target words. However, there is a problem here, since about 15% of the nature words are implicational salient, but only 32% of target words are implicational salient. We would expect the vast majority, if not all, target words to be implicationally salient. Hence, the model will not produce the human data. In addition, similar problems arise with other settings of the implicational threshold. This means that human relatedness does not fully characterise implicational meaning.

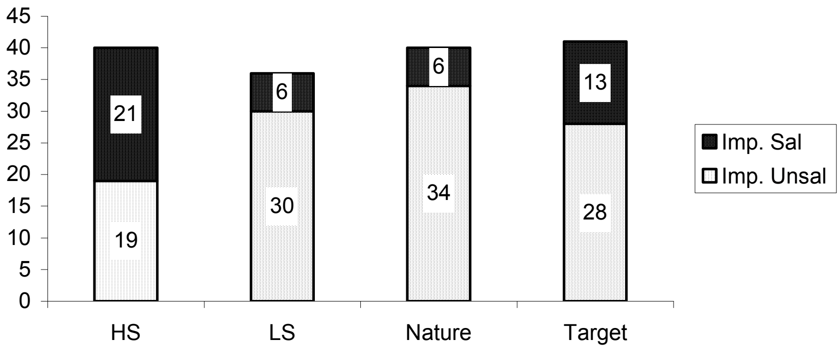

Figure 14 Number of type U (implicationally unsalient) and J (implicationally salient) words in HS, LS, nature and target words category

The failure of this analysis again suggests that a more sophisticated approach is required. It suggests that, in the context of this experiment, implicational meaning is a multidimensional evaluation. In order to further investigate this issue, we integrate the LSA calculations in relation to generic human, generic occupation, generic payment, generic household, and nature pool. We did not include the target pool and the reason will become clear later. The result is computed as a weighted sum of these five different values.

$$m = \frac{\cos_1 \cdot w_1 + \cos_2 \cdot w_2 + \cdots + \cos_n \cdot w_n}{n},$$

where n is 5, each cos value is an LSA distance to a pool of words, and each w is a weight with respect to its pool. These weights indicate how informative each dimension is as a predictor of implicational meaning. We can regard the m calculation as a new LSA measurement that characterises implicational meaning. Effectively, we are "skewing" the LSA space according to the extraction of implicational meaning. The five weights characterise this skewing, reflecting the relative emphasis that the implicational schema puts on each of the five dimensions.

Note that the above equation is actually the activation of a linear neuron with n inputs (O'Reilly & Munakata, 2000). Hence, we construct a neural network to determine these weights.

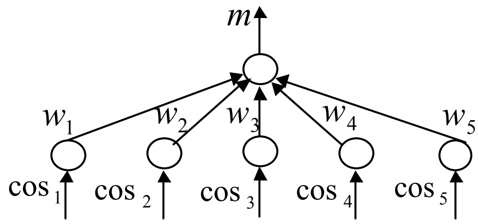

Figure 15 A neural network that computes the *m* value.

The network shown in Figure 15 has two layers. The input layer contains five neurons, one for each of the five word pools. Activations of the neurons are LSA cosines. The output layer is a single neuron with activation the *m* value. We train the network using all the words we use in the AB experiment. The learning algorithm used is the delta rule (O'Reilly & Munakata, 2000). The inputs are LSA cosines and the expected output is 1 for targets and 0 for non-targets. The learning finishes when the weight settles, i.e. their changes are smaller than a given value (*e=0.0001*). Note, this stop criterion does not guarantee accurate identification of the targets. However, this is consistent with the fact that Imp cannot identify targets accurately. Using the trained network, we can calculate the *m* values for all words. We will then derive the implicational semantic space based on it.

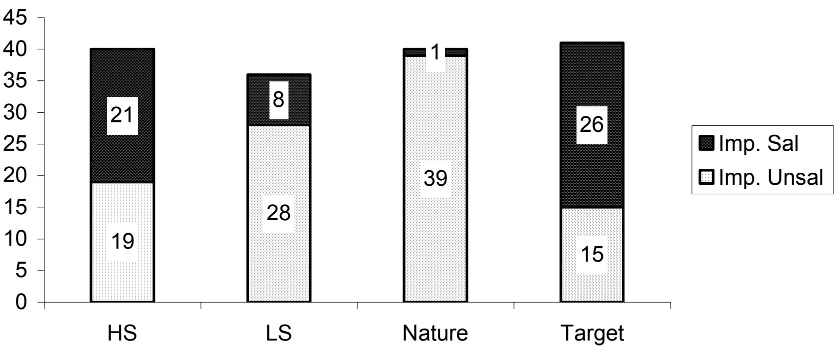

Figure 16 Number of type U (implicationally unsalient) and J (implicationally salient) words in HS, LS, nature and target word categories

Figure 16 shows that this new calculation solves the issue in Figure 14. 52.5% of high salient distractors and 22.2% of low salient distractors are implicationally salient. Nature words are mainly implicationally unsalient, except for one word, "pinnacle". (We exclude this word from our simulation.) 63.4% of target words are implicationally salient, except for "bookie, captain, chauffeur, colonel, dentist, dustman, headmaster, housemaid, lumberjack, optician, potter, constable, florist, and shepherd". Some of these words, e.g. dustman, headmaster, housemaid, lumberjack, potter and publican, are rarely used in American vocabulary[4]. Other words, e.g. bookie, captain, colonel, constable and courier, are not often used as jobs or professions in American. So, their relation to jobs and professions is lower than would be the case in the UK. Hence, the American LSA calculations may not reliably reflect our experimental results in the UK. We assume that 82.9% of targets are implicationally salient after reconsidering the difference between American and

[4] This is determined by whether they have an entry in the Cambridge Dictionary of American English.

British English. Note, the result is different from our idealised model in Figure 10. However, taking into account human perception, this new calculation gives a nice approximation of the extentionalist model.

4.1.8 Partial Awareness and Three Types of Responses

We have explained how Imp assigns salience according to LSA. In order to fully explore partial processing of the targets, we also investigate the salience assignment at Prop. Imp considers one level of meaning, which encodes an abstract schematic model. Prop considers another level of meaning, which encodes a description of the referential and specific relationships in semantic space. As we have discussed previously, Imp plays a key role in reproducing modulations in blink curve depth by distractor salience. Hence, Prop was regarded to be prefect in distinguishing targets. However, in this section, Prop also makes mistakes in detecting targets. 7% of the time, Prop assigns unsalient to a salient data representation. As we have discussed previously, 82.9% of the targets are implicationally salient. 17.1% of targets are both implicationally and propositionally unsalient (type U targets). 66.7% of targets are both implicationally and propositionally salient (type P targets), i.e. at least three data representations are correctly assigned by Prop ($82.9\% \times (1-7\%)^3 = 66.7\%$). 0.001% of targets are implicationally but propositionally fully unsalient, i.e. none of the data representations are correctly assigned by Prop ($82.9\% \times (7\%)^6 = 0.001\%$). About 16.2% ($100\% - 66.7\% - 17.1\% = 16.2\%$) of targets are partially processed by Prop, i.e. only one or two data representations are correctly assigned by Prop. These calculations determine the baseline performance of different responses:

- Baseline of correct report of target identities is 66.7%,
- Baseline of "yes" responses is 16.2%,
- Baseline of "no" responses is 17.1%.

The raw simulation results are averaged according to the LSA evaluation in Figure 16, baseline performance calculated previously, and convolved with sliding noise. Figure 17 compares human data on the left and simulation results on the right. The top two are the results of the HS condition and the bottom two are the results of the LS condition. It can be seen from Figure 17 that the model produces very similar results to human subjects.

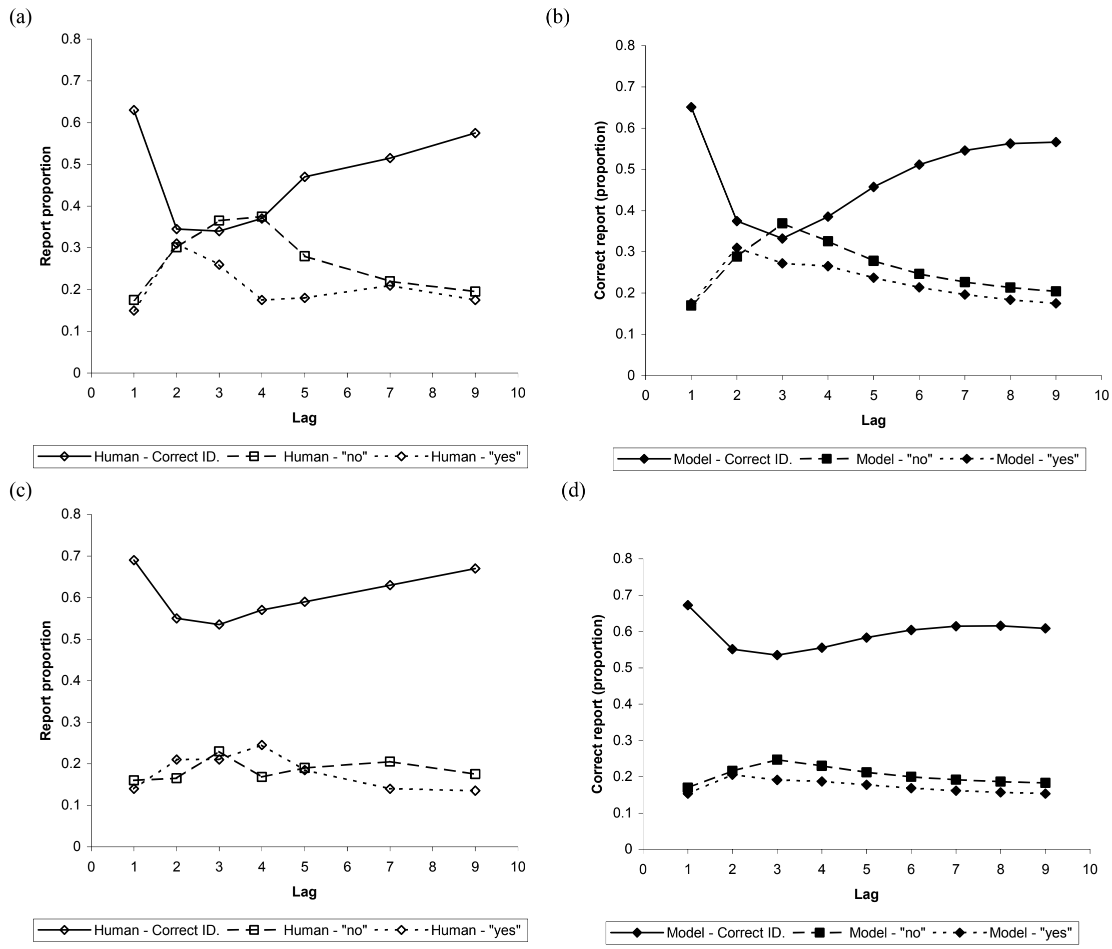

Figure 17 (a) Human performance in high salient condition, (b) simulation results in high salient condition, (c) human performance in low salient condition, (d) simulation results in low salient condition.

4.2 Approach 2: Modelling Noise

4.2.1 Data Representaions

The data representations in this model are the same as the extensionalist model, i.e. words are modelled by their types.

4.2.2 Architecture and Delay Lines

The structure of this model and the delay lines are the same as the previous models.

4.2.3 Salience Assignment

The salience assignment in this model is the same as the extentionalist model, i.e. using probabilities derived from the external observations of human data.

4.2.4 Buffer Movement, Parameter Setting and Noise Levels

The key parameters used in this intentionalist model approach are the same as the extentionalist model, except that we reconstruct the model by introducing noise to the delay of the buffer movement within the simulation. As a reflection of the fact that this is a more concrete model than the previous one, convolutions are not used here. Similar to (P. J. Barnard & Bowman, 2004), different amounts of noise are added into the buffer delay at different stages, i.e. less noise is added to the delay of buffer movement from Imp to Prop than the delay of buffer movement in the opposite direction. This is also justified by the extensionalist model, i.e. the sliding noise ensures that the noise level increases by lag. As we have discussed previously, the buffer movement delay is not assigned arbitrarily. Instead, the time course of the blink constrains the range and the distribution of buffer delay. Hence, we have performed a simple search on the distribution of the buffer moving from Imp to Prop in order to fit with the human data (P. J. Barnard et al., 2004). As a result, we found that one of the simplest distributions is setting it to either 180 or 220ms with equal probability. However, the buffer movement in the other direction is less constrained by the time course, so we have to perform a wider range of search in the parameter space for a suitable distribution. The result is that it may be sampled from the following ten possibilities: 20, 100, 180, 260, 340, 420, 500, 580, 660 and 740ms according to a normal distribution. A probability mass of 0.0228 is associated with 20 and 740; 0.0441 with 100 and 660; 0.0918 with 180 and 580; 0.1499 with 260 and 500; and 0.1914 with 340 and 420.

4.2.5 Results

Figure 18 compares human data on the left and simulation results on the right. The top two are the results of the HS condition and the bottom two are the results of the LS condition. It can be seen from Figure 18 that the model produces very similar results to human subjects.

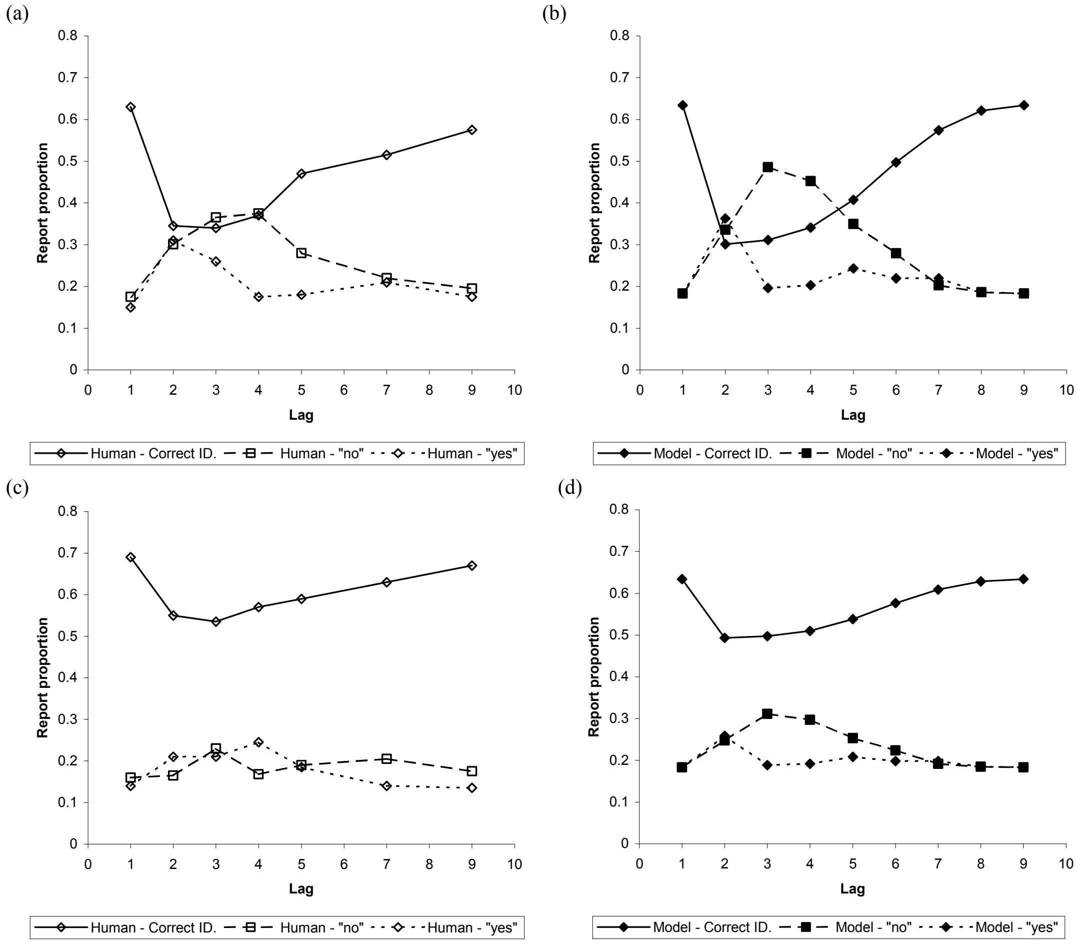

Figure 18 (a) Human performance in high salient condition, (b) simulation results in high salient condition, (c) human performance in low salient condition, (d) simulation results in low salient condition.

4.3 Full Intensionalist Model

In this section, we combine the two approaches 1) modelling meaning using LSA 2) introducing noise inside the model simulations. We have also obtained very similar results from the previous models, as shown in Figure 19.

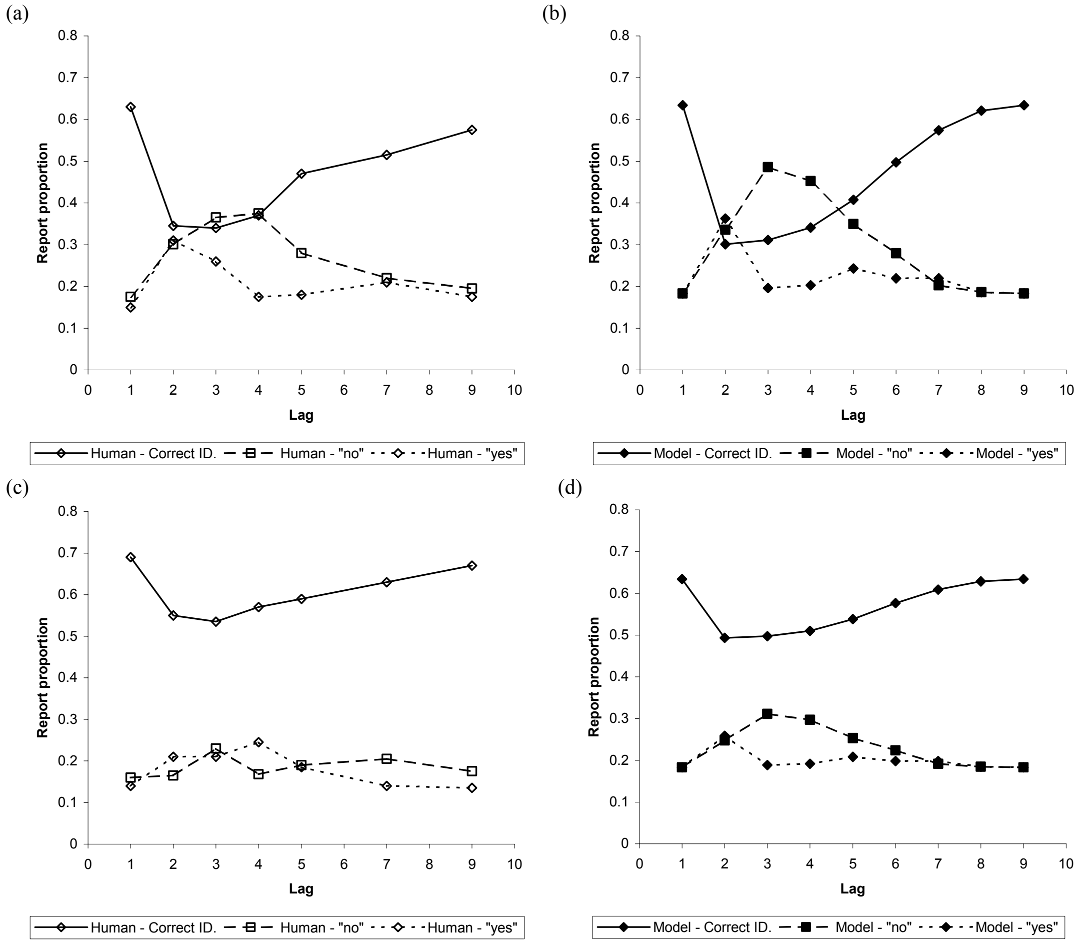

Figure 19 (a) Human performance in high salient condition, (b) simulation results in high salient condition, (c) human performance in low salient condition, (d) simulation results in low salient condition.

5 Comparing the Extensionalist and the Intensionalist Models

	Extensionalist Model	Modelling Meaning	Modelling Noise	Intentionalist Model
Data Representation	By word types: *Word_tp ::= Back | Targ | Dist*	By word identities: *Word ::= {word1, word2, word3, ...}*	By word types: *Word_tp ::= Back | Targ | Dist*	By word identities: *Word ::= {word1, word2, word3, ...}*
Structure	2 subsystems	2 subsystems	2 subsystems	2 subsystems
Delay Lines	10 items	10 items	10 items	10 items
Salience Assignment	According to the probabilities derived from human data: $P_{imp}(Targ) = 0.67$ $P_{imp}(Back) = 0$ $P_{imp}(HS) = 0.49$ $P_{imp}(LS) = 0.19$	Proportion of type I, J, P and U words derived from the LSA calculations: $I(Targ) = 82.9\%$ $P(Targ) = 66.7\%$ $J(Back) = 2.5\%$ $J(HS) = 52.5\%$ $J(LS) = 22.2\%$	According to the probabilities derived from human data: $P_{imp}(Targ) = 0.67$ $P_{imp}(Back) = 0$ $P_{imp}(HS) = 0.49$ $P_{imp}(LS) = 0.19$	Proportion of type I, J, P and U words derived from the LSA calculations: $I(Targ) = 82.9\%$ $P(Targ) = 66.7\%$ $J(Back) = 2.5\%$ $J(HS) = 52.5\%$ $J(LS) = 22.2\%$
Buffer	$D_1 = D_2 = 200ms$	$D_1 = D_2 = 200ms$	D_1 is randomly	D_1 is randomly

Movement Delay			sampled from a narrow distribution around 200ms. D_2 is randomly sampled from a broad distribution from 20ms to 740ms.	sampled from a narrow distribution around 200ms. D_2 is randomly sampled from a broad distribution from 20ms to 740ms.
Awareness and types of Responses	Partial processing at fixed serial position	Partial processing at fixed serial position	Partial processing at different serial positions according to buffer delay distribution.	Partial processing at different serial positions according to buffer delay distribution.
Baseline performance	According to the probabilities derived from human data: 15% of the responses are "yes" responses, 18% of the responses are "no" responses.	Proportion of type I, J, P and U target words derived from the LSA calculations and partial process: $P(Targ) = 66.7\%$ $U(Targ) = 17.1\%$ 16.2% of targets are partially processed.	According to the probabilities derived from human data: 15% of the responses are "yes" responses, 18% of the responses are "no" responses.	Proportion of type I, J, P and U target words derived from the LSA calculations and partial process: $P(Targ) = 66.7\%$ $U(Targ) = 17.1\%$ 16.2% of targets are partially processed.
Combining epochs (correct report of target identities)	$perc(1) = P_{imp}(\neg Dist \wedge Targ)$ $perc(0) = P_{imp}(\neg Targ)$ $perc(blink) = P_{imp}(Dist \wedge Targ)$ Note: in all epochs, blink onset and recovery are at the same lag, due to the fixed buffer delay.	$perc(1) = (1 - J(Dist)) \times P(Targ)$ $perc(0) = 1 - P(Targ)$ $perc(blink) = J(Dist) \times P(Targ)$ Note: in all epochs, blink onset and recovery are at the same lag, due to the fixed buffer delay.	$perc(1) = P_{imp}(\neg Dist \wedge Targ)$ $perc(0) = P_{imp}(\neg Targ)$ $perc(blink) = P_{imp}(Dist \wedge Targ)$ Note: in each epoch, blink onset and recovery are at different lags, due to different buffer delays.	$perc(1) = (1 - J(Dist)) \times P(Targ)$ $perc(0) = 1 - P(Targ)$ $perc(blink) = J(Dist) \times P(Targ)$ Note: in each epoch, blink onset and recovery are at different lags, due to different buffer delays.
Combining epochs ("yes" responses)	$perc(1) = 15\%$ $perc(blink) = P_{imp}(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$	$perc(1) = J(Targ) = 16.2\%$ $perc(blink) = J(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$	$perc(1) = 15\%$ $perc(blink) = P_{imp}(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$	$perc(1) = J(Targ) = 16.2\%$ $perc(blink) = J(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$
Combining epochs ("no" responses)	$perc(1) = 18\%$ $perc(blink) = P_{imp}(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$	$perc(1) = U(Targ) = 17.1\%$ $perc(blink) = J(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$	$perc(1) = 18\%$ $perc(blink) = P_{imp}(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$	$perc(1) = U(Targ) = 17.1\%$ $perc(blink) = J(Dist)$ $perc(0) = 100\% - perc(1) - perc(blink)$
Adding Noise	Convolution with sliding noise	Convolution with sliding noise	Add noise in D_1 and D_2 as shown above	Add noise in D_1 and D_2 as shown above
Simulation Results	Similar to the human behaviour and capturing the effect of blink depth.	Similar to the human behaviour and capturing the effect of blink depth. Also, similar to the results of the extensionalist model.	Similar to the human behaviour and capturing the effect of blink depth. Also, similar to the results of the extensionalist model.	Similar to the human behaviour and capturing the effect of blink depth. Also, similar to the results of all other models.

Table 3 Comparison between models at different levels of abstraction

The key findings from this comparison are:

- The two mechanisms for modulating depth of the blink achieve similar effects, i.e. from probabilities observed externally from the human data and from underling semantic meaning derived from LSA.
- The two methods of introducing noise have similar effects on the blink curves, i.e. via convolving with sliding noise and adding different levels of noise to buffer movement delays.

6 Discussion

6.1 Attentional Capture by Meaning

We have provided a concrete account of attentional capture by meaning and the temporal dynamics of that process. Key principles that underlie this account are the division of the processing of meaning into two stages, which are supported by the implicational and propositional subsystems. Each subsystem assesses a different type of meaning: implicational and propositional, respectively. In the context of Barnard et al's key-distractor blink task, attention is captured when a key-distractor is interpreted as implicationally salient. This then causes attention (i.e. the buffer) to be redeployed to the propositional subsystem, in order to enable a more detailed (propositional) assessment of the salience of the key-distractor. Critically, this redeployment of attention leaves a temporal window in which implicational salience is not assessed, leaving the system vulnerable to missing even highly salient items. It is through this mechanism that the model blinks.

A number of key findings have arisen from our modelling. Firstly, we have provided further evidence for the applicability of Latent Semantic Analysis in the context of attentional capture by meaning. That is, we have shown that a model that measures semantic distance using LSA can reproduce the key-distractor blink and semantic modulations of blink depth. Furthermore, we have shown that these LSA calculations are consistent with a more extensionalist approach in which probabilities of ascribing implicational and propositional salience are derived directly from the blink curve; see section 3.7. This is an illustration of how a multilevel modelling approach can provide converging evidence for a theoretical position. The identification of a schema for implicational salience assessment (see section 4.1.7) is an important further contribution.

Secondly, we have clarified the characteristics of attentional redeployment when meaning captures attention. In particular, at an extensionalist level, the need to use a convolution with sliding noise (see section 3.8) suggests that temporal noise increases systematically by serial position. At a more intensionalist level, this sliding noise is realised as variance in the buffer movement delay. That is, there is little variance in the delay in buffer movement from implic to prop, while there is a good deal of variance in the delay in buffer movement from prop back to implic. One possible explanation of this finding would be that there is greater variance in propositional than implicational salience assessment.[5] Thus, the indication is that there is less variance in extracting semantic gist (at Implic) than extracting referential meaning (at Prop). This is not surprising, since, unlike Prop, Implic does not have to fully analysis and generate a concrete referent, which is likely to be affected by many variables. Thus, it seems likely that Implic has a more fixed timecourse than Prop.

From a more general perspective, we present a spectrum of computational models of attentional capture by meaning, each of which provides an excellent fit to blink data. Furthermore, our multi model and level approach provides a high degree of converging evidence for our theoretical proposals.

6.2 Cognitive Architectures

The model presented here can be placed within the context of a broad cognitive theory: the Interacting Cognitive Subsystems (ICS) architecture (P. Barnard, 1985; P.J. Barnard, 1999). Distributed control is inherent in ICS: subsystems are independent components, which interact through exchange of data representations over communication channels (P. J. Barnard, 1999; P. J. Barnard & Bowman, 2004; H. Bowman & Barnard, 2001; H. Bowman & Faconti, 1999). ICS asserts that cognition emerges as the product of the interaction between a set of autonomous subsystems. Both the delay-line and buffering concepts that we use have their roots in ICS. However, most significantly, the implicational - propositional distinction reflects ICS' dual-subsystem central engine (Teasdale & Barnard, 1993).

6.3 Multi-level Cognitive Modelling

We have provided a case study for how multilevel modelling can be applied in the cognition setting. However, to really reap the benefit of such modelling, new simulation paradigms and methods need to be investigated, such as the following.

1) In computer science there are a number of mathematically characterised refinement relations and transformations (H. Bowman & Gomez, 2006; Derrick & Boiten, 2001; Roscoe, 1998), along with supporting decision procedures to check such relations (Roscoe, 1998). An important research agenda is to investigate how these notions of refinement carry over to the cognitive modelling setting. In particular, can systematic refinement based cognitive modelling trajectories be developed? A central question that needs to be addressed is how refinement carries over to a setting in which data distributions (e.g. reaction times) need to be simulated; typically in computer science, requirements are completely deterministic.

2) A thorny question that refinement prompts is how to relate symbolic and subsymbolic (neural) models. One might argue that the archetypal abstraction level distinction in cognitive modelling is that between the symbolic and the

[5] Note, the model is set-up such that each subsystem has a fixed basic salience assessment delay, which effectively sets a minimal speed of assessment. Variance in salience assessment is added on top of this basic delay through variance in the corresponding buffer movement delay.

subymbolic, with the former being inherently abstract and the latter tied directly to "neural implementation". Thus, to span the full spectrum of abstraction levels, a means to relate symbolic and subsymbolic would really need to be identified. This relationship is, of course, also central to key questions in cognitive neuroscience, such as how language is coded in the brain (Fodor & Pylyshyn, 1988). Despite a number of decades of work, the symbolic-subsymbolic question has still not been conclusively answered. However, important strides have been made, e.g. (d'Avila Garcez, Broda, & Gabbay, 2002). In addition, (Su, Bowman, & Wyble, 2005, 2006) have made a preliminary contribution on how to encode neural networks in communicating automata (which are related to the process algebra used here) and also explore how formal verification methods, particularly model-checking (H. Bowman & Gomez, 2006), can be used in this context. This suggests a possible junction between the type of modelling notations employed in this paper and neural networks.

3) A less ambitious, although fully justified, approach would be to develop refinement-like relationships within a single modelling class. For example, within the neural networks domain, is it possible to relate the kind of abstract (rate-coded) connectionist models employed in cognitive psychology, e.g. Stroop models (Cohen, Dunbar, & McClelland, 1990), to the neurophysiologically more detailed models used in neuroscience, e.g. (Dehaene et al., 1998)?

It is unrealistic though at present to believe that we can support all "refinement steps" with formally defined relationships. Despite this, it may still be beneficial to think in terms of refinement trajectories, even though they are not at present formally supported. That is, the discipline provided by multilevel modelling may be valuable even if it is reliant upon informal consistency and refinement relationships. Viewing systems from different perspectives and levels of abstraction is just a useful exploratory method for understanding systems, and it is one that the cognitive modelling domain should not miss.

7 Appendix A

List of targets:
accountant, apprentice, architect, baker, bodyguard, bookie, builder, butcher, captain, chauffeur, colonel, constable, courier, dentist, doctor, dustman, editor, engineer, florist, headmaster, housemaid, janitor, jockey, librarian, locksmith, lumberjack, mechanic, mercenary, merchant, optician, parson, playwright, potter, printer, publican, salesman, secretary, shepherd, surgeon, therapist, waitress.

List of high salient distractors:
Accomplice, admirer, adversary, auntie, cousin, coward, disciple, egghead, enthusiast, extrovert, grandson, heathen, heckler*, heretic, hooligan, hunchback, husband, informant, kinsman, loudmouth*, neighbour, opponent, patron, pedestrian, pragmatist, raconteur, refugee, saviour, scoundrel, shopper, spectator, stranger, sweetheart, thinker, tourist, vegetarian, visionary, visitor, volunteer, voter, widow, witness.

List of low salient distractors:

barometer, brolly*, button, cabinet, cellophane, chandelier, cosmetic, cupboard, curtain, deodorant, detergent, dictionary, elastoplast*, flagon*, footrest*, freezer, hammer, handle, ladder, ladle, lantern, notepaper, oven, percolator, picture, piggybank*, pillow, porcelain, potstand*, projector, radiator, settee, souvenir, spatula, spotlight, staircase, tablecloth, tankard, television, toothpaste, trolley, wireless.

List of nature word distractors:
arctic, vegetation, river, precipice, atmosphere, ocean, blizzard, valley, seabed, permafrost, thicket, volcano, pebble, hurricane, cavern, sandbank, quicksand, escarpment, canyon, terrain, crater, spinney, amethyst, pasture, thunder, horizon, rainforest, undergrowth, glacier, prairie, equator, meteor, daylight, boulder, mountain, tundra, continent, wilderness, lava, pinnacle.

List of background word fillers:
archipelago, backwoods, beach, biosphere, brook, channel, cliff, cloud, cloudburst, coastline, crevasse, crevice, cyclone, desert, diamond, drought, sediment, earthquake, eruption, estuary, everglades, fissure, fjord, floodplain, frost, geyser, gorge, grassland, habitat, hailstone, headwind, hillside, hoarfrost, iceberg, icecap, inlet, island, landscape, lightning, limestone, meadow, monsoon, moonlight, moraine, mudflats, outback, outcrop, pampas, plains, plateau, puddle, quartz, rainbow, raindrop, rapids, reef, riverbank, riverbed, saltmarsh, sandstorm, savannah, seashore, shoreline, skyline, snowflake, straits, stream, sunshine, swamp, tempest, tornado, breeze, tributary, causeway, waterfall, wetlands, whirlpool, woodland.

Note, LSA calculations were obtained from a database at University of Colorado Boulder http://lsa.colorado.edu/. * denotes words that do not have LSA entries in the database.

List of generic human words:
human, people, mankind, womankind, someone, mortal, fellow, sentient, folk, soul.

List of generic payment words:
payment, fee, remuneration, recompense, bribe, salary, honorarium, income, earnings, wages.

List of generic household words:
ornament, device, utensil, gadget, tool, possession, decoration, fitting, fixture, furnishing.

List of generic occupation words:
occupation, profession, job, trade, employment, work, business, career, livelihood, vocation.

8 Appendix B

Instead of using nine different GDN functions, we are interested to explore whether a single noise function could also be used to produce the blink curve. Several different noise functions have been considered. One of them is shown in Figure 20(a). This function is based on the typical waveform of crosstalk noise generated by resistor and capacitor (RC) circuits, which is a circuit with a voltage source (battery) connected in series with a resistor and a capacitor (Heydari & Pedram, 2005; Takahashi, Hashimoto, & Onodera, 2001). It is widely believed that neural circuits can be modelled using RC circuits (Koch, 1999). Hence, neural networks introduce the same type of noise as generic RC circuits. This approach is also consistent with the mechanism used in the previous model, i.e. the underlying mechanism of salience assessment is less noisy in the earlier stages than later. As is shown in Figure 20(b), the convolution can smooth the original simulation results. The shape of the noise function also produces a sharp blink onset and shallow blink recovery. However, this approach does not solve the reduced lag-1 performance problem.

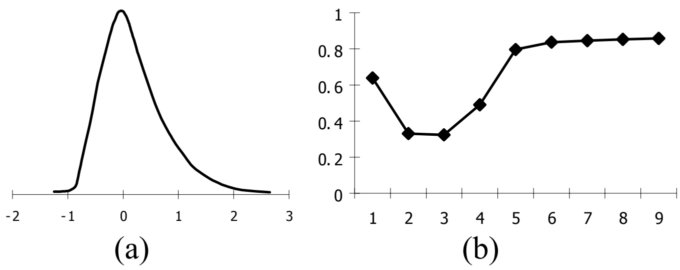

Figure 20: (a) a noise function. (b) The result of the convolution.

9 References

Abeles, M., Bergman, H., Margalis, E., & Vaadia, E. (1993). Spatiotemporal Firing Patterns in the Frontal Cortex of Behaving Monkeys. *Journal of Neurophysiology, 70*, 1629-1638.

Anderson, A. K. (2005). Affective influences on the attentional dynamics supporting awareness. *J. Exp Psychol Gen, 134*(2), 258-281.

Barnard, P. (1985). Interacting Cognitive Subsystems: A Psycholinguistic Approach to Short-term Memory. In A. Ellis (Ed.), *Progress in the Psychology of Language, Vol. 2* (pp. 197-258). Hove, UK: Lawrence Erlbaum Associates Ltd.

Barnard, P. J. (1999). Interacting Cognitive Subsystems: Modelling Working Memory Phenomena with a Multi-processor Architecture. In A. Miyake & P. Shah (Eds.), *Models of Working Memory* (pp. 298-339). Cambridge, UK: Cambridge University Press.

Barnard, P. J. (1999). Interacting Cognitive Subsystems: modelling working memory phenomena within a multi-processor architecture. In *Models of Working Memory: Mechanisms of active maintenance and executive control* (pp. 298-339).

Barnard, P. J., & Bowman, H. (2004). Rendering information processing models of cognition and affect computationally explicit: Distributed executive control and the deployment of attention. *Cognitive Science Quarterly, 3*(3), 297-328.

Barnard, P. J., Ramponi, C., Battye, G., & Mackintosh, B. (2005). Anxiety and the Deployment of Visual Attention over Time. *Visual Cognition, 12*(1), 181-211.

Barnard, P. J., Scott, S., Taylor, J., May, J., & Knightley, W. (2004). Paying attention to meaning. *Psychol Sci, 15*(3), 179-186.

Bond, A. H. (1999). Describing Behavioural States Using a System Model of the Primate Brain. *American Journal of Primatology, 49*, 315-338.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modelling Language User Guide*: Addison-Wesley.

Bowman, H., & Barnard, P. J. (2001). Computational Modelling of Distributed Executive Control. *Technical Report No. 12-01, Computer Science at Kent*.

Bowman, H., & Derrick, J. (Eds.). (2001). *Formal Methods for Distributed Processing, A Survey of Object Oriented Approaches*. London: Cambridge University Press.

Bowman, H., & Faconti, G. (1999). Analysing Cognitive Behaviour using LOTOS and Mexitl. *Formal Aspects of Computing, 11*, 132-159.

Bowman, H., & Gomez, R. S. (2006). *Concurrency theory, calculi and automata for modelling untimed and timed concurrent systems*: Springer.

Bowman, H., Steen, M. W. A., Boiten, E. A., & Derrick, J. (2002). A formal framework for viewpoint consistency. *Formal Methods in System Design, 21.*(2), 111-166.

Bowman, H., & Wyble, B. (2005). The Simultaneous Type, Serial Token Model of Temporal Attention and Working Memory. *Psychological Review (to appear)*.

Chun, M. M., & Potter, M. C. (1995). A Two-Stage Model for Multiple Target Detection in Rapid Serial Visual Presentation. *Journal of Experimental Psychology: Human Perception and Performance, 21*(1), 109-127.

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the Stroop effect. *Psychol Rev, 97*(3), 332-361.

d'Avila Garcez, A. S., Broda, K. B., & Gabbay, D. M. (2002). *Neural-Symbolic Learning Systems, Foundations and Applications*: Springer.

Dehaene, S., Kerszberg, M., & Changeux, J. P. (1998). A neuronal model of a global workspace in effortful cognitive tasks. *Proc Natl Acad Sci U S A, 95*(24), 14529-14534.

Derrick, J., & Boiten, E. (2001). *Refinement in z and object-z: Foundations and advanced applications*: Springer.

Erickson, T. D., & Mattson, M. E. (1981). From words to meaning: a semantic illusion. *Journal of Verbal Learning and Verbal Behavior, 20*, 540-551.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and Cognitive Architecture: A Critical Analysis. *Cognition, 28*, 3-71.

Gries, D., & Schneider, F. B. (1994). *A Logical Approach to Discrete Maths*: Springer-Verlag.

Harel, D. (1987). Statecharts: A visual Formalism for complex systems. *Science of Computer Programming, 8*, 231-274.

Heydari, P., & Pedram, M. (2005). Capacitive Coupling Noise in High-Speed VLSI Circuits. *IEEE Tans. On Computer Aided Design, 24*(3).

Hinton, G. E. (1990). Special Issue of Journal Artificial Intelligence on Connectionist Symbol Processing (edited by Hinton, G.E.). *Artificial Intelligence, 46*(1-4).

Hoare, C. A. R. (1985). *Communicating Sequential Processes*. London: Prentice-Hall.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C. V., Loingtier, J.-M., et al. (1997). Aspect-Oriented Programming. In *proceedings of the European Conference on Object-Oriented Programming (ECOOP), LNCS 1241*: Springer-Verlag.

Kieras, D. E., & Meyer, D. E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12*, 391-438.

Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*: Oxford University Press.

Landauer, T. K., & Dumais, S. T. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of the Acquisition, Induction and Representation of Knowledge. *Psychological Review, 104*, 211-240.

Maki, W. S., Frigen, K., & Paulsen, K. (1997). Associative Priming by Targets and Distractors During Rapid Serial Presentation. *Journal of Experimental Psychology: Human Perception and Performance, 23*, 1014-1034.

Manna, Z., & Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems*: Springer-Verlag.

Marr, D. (2000). Vision. In R. Cummins & D. D. Cummins (Eds.), *Minds, Brains and Computers, The Foundation of Cognitive Science (An Anthology)* (pp. 69-83): Blackwell.

McLeod, P., Plunkett, K., & Rolls, E. T. (1998). *Introduction to Connectionist Modelling of Cognitive Processes*: Oxford University Press.

Milner, R. (1986). Process Constructors and Interpretations. In H.-J. Kugler (Ed.), *Information Processing 86* (pp. 507-518): IFIP.

Milner, R. (1989). *Communication and Concurrency*. Hemel Hempstead, UK: Prentice-Hall.
Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, Massachusetts: Harvard University Press.
O'Reilly, R. C., & Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain.*: MIT Press.
Raymond, J. E., Shapiro, K. L., & Arnell, K. M. (1992). Temporary Suppression of Visual Processing in an RSVP Task: An Attentional Blink. *Journal of Experimental Psychology: Human Perception and Performance, 18*(3), 849-860.
Roediger, H. L., & McDermott, K. B. (1995). Creating false memories: Remembering words not presented in lists. *Journal of Experimental Psychology: Learning, Memory and Cognition, 21*, 803-814.
Rolls, E. T., & Stringer, S. M. (2001). A Model of the Interaction between Mood and Memory. *Network: Computation in Neural Systems, 12*, 89-109.
Roscoe, A. W. (1998). *The Theory and Practice of Concurrency*: Prentice-Hall.
Rumelhart, D. E., McClelland, J. L., & the-PDP-Research-Group. (1986). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition. Volume 1: Foundations and Volume 2: Psychological and Biological Models*. Cambridge, Massachusetts: A Bradford Book, The MIT Press.
Shadlen, M. N., & Newsome, W. T. (1994). Noise, neural codes, and cortical organization. *Current Opinion in Neurobiology, 4*.
Shapiro, K. L., Caldwell, J. I., & Sorensen, R. E. (1997). Personal names and the attentional blink: The cocktail party revisited. *Journal of Experimental Psychology: Human Perception and Performance, 23*, 504-514.
Shapiro, K. L., & Luck, S. J. (1999). The attentional blink: A front-end mechanism for fleeting memories. In V. Coltheart (Ed.), *Fleeting memories, Cognition of Brief Visual Stimuli* (pp. 95-118). Boston, Massachusetts: A Bradford Book, MIT Press.
Stewart, N., Brown, G. D. A., & Chater, N. (2005). Absolute Identification by Relative Judgment. *Psychological Review, 112*(4), 881-911.
Su, L., Bowman, H., & Wyble, B. (2005). Symbolic encoding of neural networks using communicating automata with applications to verification of neural network based controllers. In *Proceedings of the IJCAI-05 Workshop on Neural-Symbolic Learning and Reasoning, NeSy'05, Edinburgh, UK (Position paper), 4 pages.*
Su, L., Bowman, H., & Wyble, B. (2006). *Formal modeling of connectionism using concurrency theory, an approach based on automata and model checking* (Technical Report No. 1-06): Computing Laboratory, University of Kent.
Takahashi, M., Hashimoto, M., & Onodera, H. (2001, 2001). *Crosstalk Noise Estimation for Generic RC Trees.* Paper presented at the Proceedings of the International Conference on Computer Design: VLSI in Computers & Processors (ICCD'01).
Teasdale, J. D., & Barnard, P. J. (1993). *Affect, Cognition and Change: re-modelling depressive thought*. Hove: Lawrence Erlbaum Associates.
Woodcock, J., & Davies, J. (1996). *Using Z, Specification, Refinement and Proof*: Prentice-hall.