DOI 10.1007/s00165-008-0094-3
The Author(s) © 2008. This article is published with open access at Springerlink.com
Formal Aspects of Computing (2009) 21: 513–539

Formal Aspects of Computing

Process algebraic modelling of attentional capture and human electrophysiology in interactive systems

Li Su,[1,4] Howard Bowman,[1] Philip Barnard[2] and Brad Wyble[3]

[1] Centre for Cognitive Neuroscience and Cognitive Systems, University of Kent, Canterbury, UK. E-mail: Li.Su@iop.kcl.ac.uk
[2] MRC Cognition and Brain Sciences Unit, Cambridge, UK
[3] Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA
[4] Section of Cognitive Neuropsychiatry, Department of Psychological Medicine, Institute of Psychiatry at King's College London, Box P068, De Crespigny Park, London SE5 8AF, UK

Abstract. Previous research has developed a formal methods-based (cognitive-level) model of the Interacting Cognitive Subsystems central engine, with which we have simulated attentional capture in the context of Barnard's key-distractor Attentional Blink task. This model captures core aspects of the allocation of human attention over time and as such should be applicable across a range of practical settings when human attentional limitations come into play. In addition, this model simulates human electrophysiological data, such as electroencephalogram recordings, which can be compared to real electrophysiological data recorded from human participants. We have used this model to evaluate the performance trade-offs that would arise from varying key parameters and applying either a constructive or a reactive approach to improving interactive systems in a stimulus rich environment. A strength of formal methods is that they are abstract and the resulting specifications of the operator are general purpose, ensuring that our findings are broadly applicable. Thus, we argue that new modelling techniques from computer science can also be employed in computational modelling of the mind. These would complement existing techniques, being specifically targeted at psychological level modelling, in which it is advantageous to directly represent the distribution of control.

Keywords: Formal methods; HCI; Stimulus rich reactive interfaces; Attentional blink; EEG; Interacting cognitive subsystems

1. Introduction

Applying formal modelling and cognitive frameworks to the design and analysis of interactive systems has gained much attention, particularly in the domain of mission and life critical applications. It has been argued by a number of researchers [DBD98, Bow99] that formal methods provide a powerful way to specify, evaluate, and verify such systems. As an important component in interactive systems, human operators and their cognitive capabilities also determine the success or the failure of such systems. Thus, attention needs to be paid to specifying and

Correspondence and offprint requests to: L. Su, E-mail: Li.Su@iop.kcl.ac.uk

analysing the human cognitive system. In this paper, we argue that formal methods can specify and simulate cognitive frameworks including interaction between users and computer systems. Furthermore, we also explore their general implications for cognitive modelling.

We are interested in a particular type of interactive system, in which humans operate in stimulus rich environments. In such systems, critical stimuli are presented rapidly to a human operator, as found in interfaces in cockpit environments or operations control rooms. In such situations, information overload may impede the ability of the operator to perceive or react to each piece of information. For example, it has been discovered that humans may miss a briefly presented second target[1] (T2) following a previous one (T1) after a few hundred milliseconds [RSA92, BST04]. This temporal attentional limitation is supported by the results of a large number of psychological experiments, known as Attentional Blink (AB) tasks [RSA92].

In the context of stimulus rich environments, we explore two methods of designing a user interface that is optimized to the attentional state of the user. In the simpler *constructive* case, critical stimuli are separated by a temporal interval long enough to ensure that each is fully processed (i.e. that one target does not fall within the attentional blink of another). Another method, which we term the *reactive* approach, ensures that human operators have successfully perceived all targets by determining directly from the user's neurophysiological state whether a target has been acknowledged. Using this information, the system can re-present any missed targets. Both approaches could potentially improve the performance of the user interface, however they require different knowledge in designing such interfaces. For example, an interface using the reactive approach needs to know which target has been perceived. In contrast, an interface using the constructive approach would require knowledge of when the user's attention is unavailable. However, there is a significant drawback with both approaches, that is, either acknowledging or inserting items may delay the presentation.

Research in the area of human–computer interaction has been mainly focused on constructive approaches. In this paper, we will also introduce *Stimulus Rich Reactive Interfaces* (SRRIs), these are human–computer interfaces that throttle the flow of information to the user according to how receptive the user is from one moment to the next by measuring physiological or eletrophysiological feedback, such as skin conductance responses or electroencephalogram (EEG) signals recorded from the scalp. In other words, an SRRI is a way of synchronizing the presentation of information with the momentary attentional state of the user. Such a system should be capable of increasing the amount of information that can be accommodated by the operator.

In order to explore the limits of temporal attention deployment in humans and how the design of an interface can help to circumvent this limit, we have developed a cognitive-level model based on formal methods [SBB07] that simulates both behavioural data and electrophysiological indicators of target processing. We have used a modelling technique called process algebra [Bow06, Hoa85, Mil89], which originated in theoretical computer science, being developed to specify and analyse distributed computer systems [Bow06]. A process algebra specification contains a set of top-level subsystems (processes) that are connected by a set of communication channels. Subsystems interact by exchanging messages along channels. Furthermore, process algebra components can be arbitrarily nested within one another, allowing hierarchical decomposition in the manner advocated in [Bar03].

Abstract specification of cognitive theories through formal models can enhance our understanding of the deeper properties of such complex cognitive architectures. These models can take the form of connectionist or other neurally inspired implementations that capture some of the biophysical aspects of the brain. Recently, the development of tool-based assistance for the implementation of such models has emerged in the context of both more traditional production system methodologies [CFS96] and connectionism [RMP86, Rei00]. In the context of this paper, a tractable computational model of human operators is central to explaining the underlying attentional mechanisms and evaluating/verifying many desired properties in the design of effective human–computer interfaces. Thus, in this paper, we will first outline the experimental paradigm and model, and then use our model to explore the potential benefit of both approaches in the stimulus rich environment. We will concentrate on illustrating how the modelling approach can provide insights into the design of both approaches, and discuss some general implications for applying formal methods to cognitive neuroscience and human–computer interaction, in particular, the modelling of cognitive architectures.

The paper is organised as follows. Section 2 provides background information concerning the attentional blink task and a description of electrophysiological recording, i.e. EEG, and how it is related to target processing. This section also introduces a cognitive framework that our modelling is based on. In Sect. 3, we present a cognitive model of a human operator. Section 4 describes how this model simulates the attentional limitations

[1] For simplicity of presentation, we use the term "target", although, more accurately, we should probably talk in terms of "potential targets", since, while the system knows what is likely to be salient, it is only the human operator that knows what is truly significant.

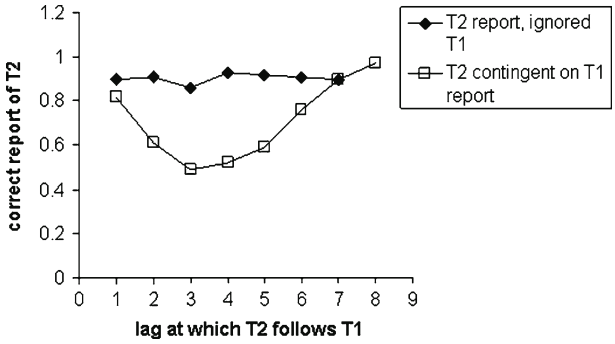

Fig. 1. The basic AB effect for letter stimuli. Here, the blink condition (*unfilled squares*) is shown as T2 accuracy conditional on T1 report, reflecting the effect on T2 report of successfully attending to T1. Baseline (*filled diamonds*) represents a person's ability to report the presence of T2 when subjects were told to ignore T1; adapted from [RSA92]

of a human with reference to the attentional blink and also produces simulated EEG signals directly from the model's components. Section 5 embeds this model into an SRRI. Sections 6 and 7 evaluate the constructive and reactive approaches with respect to improving SRRIs. Section 8 concludes with a general discussion on formal modelling of cognitive architectures and future work.

2. Background

2.1. Temporal bottleneck in attention

In the context of this paper, we first explore the temporal attentional limitation of human operators, and then go on to develop arguments about how this affects their ability to interact with computer systems. Raymond et al. [RSA92] designed an AB task, which involves letters being presented at the same spatial location using *Rapid Serial Visual Presentation* (RSVP) at around 10 items per 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" (T2) was among the items that followed T1. This condition is called the blink condition, where subjects have to report the identity of T1 and presence of T2. The key finding was that detection of T2 was impaired with a characteristic serial position curve; see the curve with unfilled squares in Fig. 1, where lag indicates the number of distractors intervening between T1 and T2, i.e. lag-1 refers to T2 immediately following T1; lag-2 refers to one intervening distractor, etc. T2s occurring immediately after T1 were accurately detected (a phenomenon typically described as lag-1 sparing). Detection then declined across serial-positions 2 (and also sometimes 3) and then recovered to baseline around lags 5 or 6 (corresponding to a target onset asynchrony in the order of 500–600 ms). However, subject's ability to detect the presence of T2 when T1 was to be ignored is unaffected by the serial position of T2. This condition is called the baseline condition, e.g. see the curve with filled diamonds in Fig. 1.

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 [MFP97] and their personal salience [SCS97]. There is also evidence from electrophysiological recording that the meaning of a target is processed even when it is not reported [Sha99]. In addition, there are now reports of specific effects of affective variables, e.g. [BRB05]. In particular, [And05] has shown that the blink is markedly attenuated when the second target is an aversive word.

These AB experiments have their counterpart in real life. For instance, when driving a car at 70 mph on the motorway, the driver monitors the traffic ahead and signposts. Events appear continuously and rapidly as those in RSVP streams, but they are somewhat inter-related. When a hazard occurs, e.g. another car suddenly changes lanes (cuts in front of you), it intrudes into this constant schema. These are high salient stimuli (targets in AB) for safe driving, but attending to such hazards (maybe reacting by breaking) could potentially knock out the driver's attention for about 500 ms. Under this situation, a driver is more likely to miss an important signpost or a junction. So, the robust AB serves as a useful paradigm to be generalised from lab to practical settings, and semantic salience is particularly relevant because most real world tasks relate to meaning or executive function.

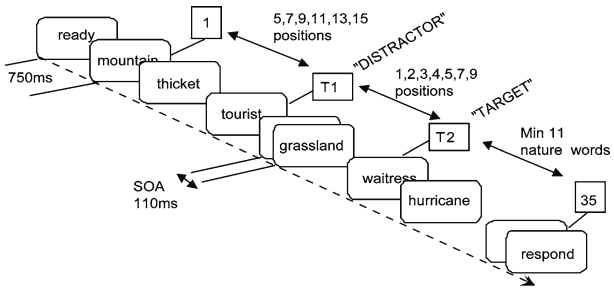

Fig. 2. Task schema for the key-distractor blink; adapted from [BST04]

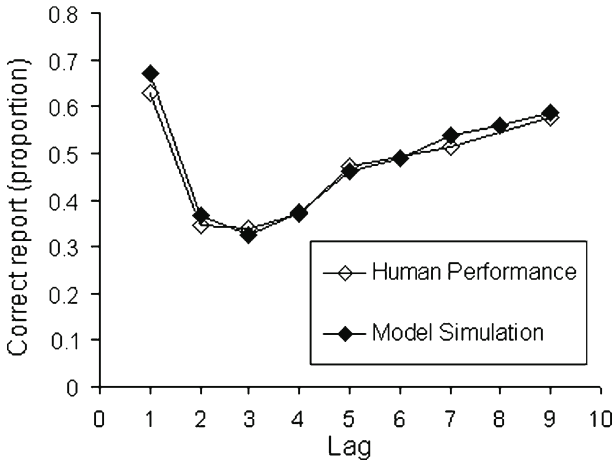

Fig. 3. The attentional capture by meaning effect in humans [BST04] and model simulations [SBB07], target report accuracy by lag of target, relative to the position of the key-distractor

In order to examine semantic effects, Barnard et al. [BST04] used a variant of the AB paradigm in which words were presented at fixation in RSVP format, at around 10 items per second. Targets were only distinguishable from background items in terms of their meaning. However, much of the empirical research and theorising on the AB has used primitive stimuli, e.g. letters or digits, which could be discriminated perceptually rather than semantically. Accordingly, most models of the AB only account for relatively low-level features, and have not yet tackled the more interesting and important aspects of meaning and emotion. We focus on high-level features such as meaning and affect, so our model can provide more direct insights into social and practical issues. This is because day-to-day human behaviour can be easily related to evaluation of meaning and emotion.

In Barnard et al.'s experiment [BST04], participants were simply asked to report a word if it referred to a job or profession for which people get paid, such as a waitress, and these targets were embedded amongst distractors (called background words in this paper) that all belonged to the same category, e.g. nature words. However, streams also contained a key-distractor item, which, although not in the target category, was semantically related to that category, e.g. tourist, vegetarian and so on; see Fig. 2. Note that the key-distractor is distinguished from other background words, since the similarity between the key-distractor and the target is directly manipulated in the experiment [BST04]. Moreover, the serial-position that the target appeared after the key-distractor was varied. The effect of attentional capture by meaning is encapsulated in the serial position curve (with unfilled diamonds) in Fig. 3. That is, the key-distractor drew attention away from the target with a clear temporal profile. In the following sections, we will call Barnard et al.'s experiment the key-distractor AB experiment, and a cognitive model of human temporal attention is described. The simulation result of this model is the serial position curve (with filled diamonds) in Fig. 3, which can be compared to the performance of real participants shown in the same graph.

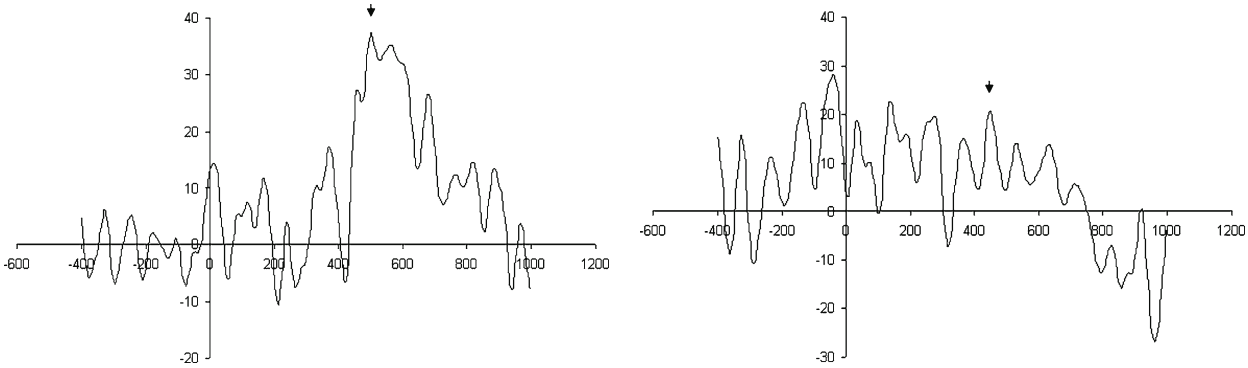

Fig. 4. Examples of raw P3s recorded from human participants (targets were presented at time 0, the y-axis units are millivolts (relative to a reference electrode), and the x-axis denotes time in milliseconds). Note that EEG is often plotted with the y-axis reversed, e.g. negative up and positive down, we have avoided this convention for reasons of presentational clarity

2.2. Electroencephalogram and brain–computer interface

The direct brain to computer interface [Vid73, LHB00, BPS03] has been one of the most useful applications to have arisen from electrophysiological neuroscience. By recording brain waves from human subjects with motor impairments, it has been possible to give the severely disabled a means of communication with the world [Cil93]. Using these techniques, a subject can passively view a matrix of characters and guide a computer interface that is recording and analysing EEG signals to the intended words he or she wishes to communicate. For some people, the electrical signal produced by their brain is the only means by which they can consciously communicate with the world. The Brain–Computer Interface (or BCI) field is now quite mature, using cutting edge technology to maximise the accuracy and speed of the information that can be read from the disabled subject's brain [MKH03].

Work in the BCI field has almost exclusively focused on human-to-computer information transfer. However, recording brain signals can serve to enhance the reliability of information flow in both directions, an application that is virtually unexplored. Wyble et al. [WCB06] have described a system that can use brainwaves to warn a computer that its user may have missed a critical piece of information. This warning will allow the computer to re-present missed information until it is perceived.

When one records EEG from the human scalp, the signal measured is deflected by ongoing cognitive operations. In the EEG literature, these deflections are referred to as components, which are observed in the Event Related Potential (or ERP) that emerges from averaging together a large number of trials, time locked to the onset of a salient stimulus.

ERP components can be manipulated experimentally; hence, they have been associated with particular cognitive processes. Whereas the early part of the waveform is associated with sensory processing of target stimuli, the later part is associated with high-level processing of a stimulus. A typical late component is the P3 (i.e. the third positive peak of the ERP, also referred to as the P300 due to its typical latency of 300 ms post-stimulus). Although some researchers have identified a frontally located P3a component, which is elicited by infrequent but task-irrelevant stimuli, we focus on the P3b, which appears maximally over parietal electrode sites. The P3b (called P3 from here on) is present for stimuli that are both infrequent and relevant to the task [SSH75]. As the P3 is only observed in the ERP for target stimuli that are detected by the subject [VLS98], it can be assumed to be an indication of an item being encoded into working memory [Don81].

Depending on the amount of noise in the signal, one normally has to average across a considerable number of trials to obtain a clean ERP waveform. However, the P3 component is often large enough to be detected even in the raw EEG. Of course, one cannot draw conclusions about P3 latency and shape from raw P3s; however, Wyble et al. [WCB06] have shown that the raw EEG is often clear enough to be detected on a trial-by-trial basis. Thus, the algorithm used in our approach focuses on these raw P3s.

Two examples of raw P3s recorded from human participants are shown in Fig. 4 (targets were presented at time 0). The diagram on the left hand side shows a clear P3, while the P3 in the diagram on the right-hand side is less obvious. The shape and time course of the P3s can be affected by many experimental factors, and (as is evident here) there are often substantial individual differences. Thus, some P3s are more readily recognisable than others.

2.3. Interacting cognitive subsystems

The general applicability of cognitive frameworks in the context of both cognitive neuroscience and human–computer interaction has been demonstrated by the success of a number of broad cognitive theories, such as SOAR [New90], ACT-R [And93] and EPIC [Mey97]. These are centralised production systems based models. One such cognitive theory is the *Interacting Cognitive Subsystems* (ICS) architecture [Bar99]. It is a highly parallel architecture, in which control of system wide interactions is distributed rather than centralised. This theory assumes that our mental architecture is composed of nine subsystems: (1) sensory subsystems: acoustic, visual, and body-state; (2) effector subsystems: articulatory and limb; (3) central subsystems: morphonolexical, object, propositional, and implicational.

Each subsystem has the same internal structure. Information arrives in a subsystem, is copied into its local memory, called an image record, and is transformed for use in other subsystems. The subsystems differ in their inputs and outputs, that is, they are each specialised for storing and processing a qualitatively different form of mental representation [Bar99]. ICS stresses that mental activity concurrently occurs in multiple domains (realised by their corresponding subsystems). Although each subsystem in ICS has a restricted functionality, much richer mental behaviour can arise out of the interaction among multiple subsystems. It will become clear later that our model can be fitted into the framework of ICS, but a complete description of ICS is outside the scope of this paper. It will also become clear shortly that in order to avoid over-specification in models such as ICS, it is important that modelling techniques supporting distributed control are used. This is exactly what is obtained through using a notation such as a process algebra.

Many broad cognitive frameworks are symbolic in nature, rather than connectionist, for a number of reasons [Bar03, Fod88]. First, symbolic systems are good at manipulating, explaining and reasoning about complex symbolic data structures. Second, the knowledge or rules are visible. Third, large systems can be composed from small structures. Finally, symbolic representations facilitate compositional and hierarchical representation of knowledge.

Broad theories, such as ICS, present a specific challenge for computational modelling in that they are presented using "box and arrow diagrams". Thus, the computational models must capture the interactions between mental subsystems and do it at an abstract level. Barnard and Bowman [Bar03] have argued that the mathematical models developed by computer scientists, such as process algebra [Bow06], offer the prospect to directly model "box and arrow diagrams". For example, in their process algebraic models, the "boxes" are modelled by processes and the "arrows" are modelled by communication channels. Issues in developing such broad models include the *irrelevant specification problem*[New90], which arises when a large number of assumptions are made during implementation. However, it is sometimes unclear what assumptions correspond to known cognitive behaviour. Fortunately, avoiding over-specification is at the heart of formal methods.

We argue that the requirements for modelling theories, such as ICS, are similar to the requirements for modelling distributed computer systems. This is because control is distributed in ICS: subsystems are independent components, which interact through exchange of data representations over communication channels and cognition emerges as the product of the interaction between a set of autonomous subsystems [Bar99]. As a reflection of this match, process algebra have already been applied to modelling ICS [Bow99, Bar03, SBB08]. The research presented here builds upon this earlier work.

3. Cognitive model of human operators

We have provided a concrete account of attentional capture by meaning and the temporal dynamics of that process [SBB07]. Key principles that underlie this account are sequential processing, 2-stages, and serial allocation of attention. We discuss these principles in turn. Note, in order to simplify the presentation, we focus on the modelling of attentional capture in this section. Detailed explanation of the modelling of EEG will be presented in Sect. 4.

3.1. Sequential processing

With any RSVP task, items arrive in sequence and need to be correspondingly processed. 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

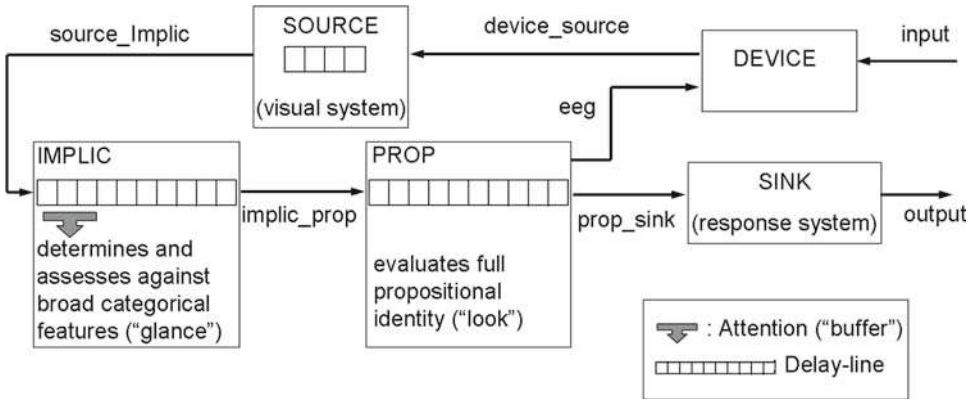

Fig. 5. Top-level structure of the two subsystems model with implicational subsystem attended. Channel labels match their corresponding action names in the specification. Note, the process `CLOCK` is omitted. It connects to every other process in the model in order to provide synchronised clock ticks. Moreover, only data flows are shown here. All control channels including those that set the parameters are also omitted to simplify the presentation

through until they reach the back of the pipeline, where they enter the response system.[2] 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 [ABM93].

A delay-line 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. We shall use this terminology throughout, i.e. a single RSVP item will be modelled by a number of constituents in a (delay-line) representation. We assume six constituent representations comprise one RSVP item/word, which approximates the 110 ms presentation used in the AB experiments [BST04].

As shown in Fig. 5, on every cycle a new constituent representation enters the pipeline and all constituent representations currently in transit are pushed along one place. We shall refer to this as the delay-line update cycle, and assume that one cycle corresponds to 20 ms. This assumption is justified by the observation that underlying neural mechanisms can represent updates on a time scale of tens of milliseconds [Bon99, Rol01]. Thus, in each delay-line update cycle, all delay-lines increment by one slot every 20 ms. Note, the update rate of the model is every 4 ms. This fine grain of time course allows us to be more discriminating with regard to the temporal properties of the AB. However, a higher refresh rate would have implementation costs, in terms of how long simulations would take to run. This assumption of 4 ms is not constrained by neurobiology, but by implementation requirements, i.e. on the one hand, it has to be faster than the update rate of constituent representations, on the other hand, it is also justified by the fact that the sampling rate of EEG for the human participants was 250 Hz. (How the EEG data is modelled will be explained in the relevant section.)

3.2. 2-Stages

Like [Chu95], we have argued elsewhere for a 2-stage model [BST04, Bar03], but this time recast to focus exclusively on semantic analysis and executive processing. In particular, [Bar03] modelled the key-distractor blink task using a 2-stage model. In the context of modelling distributed control, we implemented the 2-stage model as a dialogue between two levels of meaning, see Fig. 5. 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. In this stage 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

[2] Note, visual system and response system do not directly map to ICS subsystems. They are abstract representations of a number of subsystems necessary to perceive stimuli and to make responses.

between the meaning of the incoming item and the target category. These two stages are implemented in two distinct semantic subsystems proposed within a multi-level model for cognition and emotion: the implicational subsystem (which supports the first stage) and the propositional subsystem (which supports the second) [Bar99].

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 [Bar99]. The other is classically "rational", being based upon propositional representation and captures referentially specific semantic properties and relationships. The exchanges between two levels of meaning reflect distributed executive functions rather than central executive control or an homunculus. In the context of the task being considered here, these subsystems can be distinguished as follows:

- Implicational Subsystem (or Implic). This performs the broad "categorical" analysis of items, which might be related to 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 idea introduced earlier of a "glance".
- Propositional Subsystem (or Prop). 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 to test whether the meaning of the incoming item meets the task specification and should therefore be reported. We will describe items that "pass the propositional test" as propositionally salient. That is, this more detailed level of semantics is required to test the specific referential meaning of an incoming item against the specification of the target category.

Semantic errors make clear that sometimes we only have referentially non-specific semantic *gist* information available, e.g. the Noah illusion illustrates implicational meaning [Eri81]. 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, such as "How many animals of each kind did Moses take on the Ark?" most people respond "two" even though they know, when pressed, that it was Noah, not Moses, who took the animals on the Ark. Substitution of Moses for Noah often fails to be noticed, while substitution with Nixon, or even Adam, is noticed. This is presumably because both Moses and Noah fit the generic (implicational) schema "aged male biblical figure" [BST04], but Nixon and Adam do not.

As outlined earlier, 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 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.

3.3. 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. It is worth noting that this is a key aspect of distributed control, i.e. limitations arise because the system can effectively look at only one source at a time. 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 [Bar99]. (In the context of this paper, the term buffer refers to a moving focus of attention.) In this respect, salience assignment can only be performed if the subsystem is buffered and only one subsystem can be buffered at a time, as shown in Fig. 5. The buffer mechanism ensures that the central attentional resources are allocated serially, while items pass concurrently, i.e. all items throughout the overall delay-line are moved on one place on each time step.

Each subsystem assigns salience on the basis of the constituent representations entering it. Salience assignment is performed at the delay-line of the subsystem when it is buffered. As explained previously, an RSVP item (i.e. a word) is composed of several constituent representations, six in the current simulation. Accordingly, the semantic meaning of a word builds up gradually through time. A subsystem accesses the meaning of a word by looking across several of its constituent representations. We assume the meaning of a word emerges from the first few representations. It is important, though, to point out that we are not talking about letter by letter reading here, but the whole word forming an image that builds up gradually through time.

In relation to the time course associated with the extraction of meaning, we assume that three constituent time slots amounting to a 60 ms presentation are required for the extraction of useful meaning. Such an estimate is consistent with early research showing that the number of items reportable from a visual array rises rapidly with exposures up to 50 ms, and plateaus thereafter [Mac63]. A 60 ms integration time also equates closely with a finding recently reported by Grill-Spector and Kanwisher [Gri05]. They show not only that detection increases with exposure durations up to 68 ms but also that at exactly the same time point as the simple detection of an object approximates maximum performance, the ability to report its category also approximates its maximal level. This indicates that accurate generic semantic information can indeed be available on the same time scale as simple detection.

As we have previously mentioned, the model presented here can be placed within the context of ICS, i.e. distributed control is inherent in ICS, and 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 [Tea93].

3.4. LOTOS specification

The top-level model shown in Fig. 5 is an example of a "box and arrow diagram" as previously discussed. Thus, it is informal and not executable. In order to further explain the AB phenomenon and constraints on the interaction of these subsystems, such informal theories have been realised as computational models. The particular specification language used here and by [Bar03] is called LOTOS [Bol88]. The top-level specification is as follows:

```
specification SRRI[para1,para2,para3,para4,input,output]:noexit

library keydisab endlib (* Data structures defined using ACT ONE, and stored
                           in a separate file. *)

behaviour

  hide device_source,source_implic,implic_prop,prop_sink,implic_prop_ctrl,
  prop_implic_ctrl,eeg,tick in
  (* Internal actions to the system. *)

  ( (* See main text for an explaination for the following specification. *)
  (DEVICE[input,device_source,eeg,para4,tick]
    |[device_source,eeg,tick]|
  ((IMPLIC[source_implic,implic_prop,implic_prop_ctrl,prop_implic_ctrl,para1,tick]
      |[implic_prop,implic_prop_ctrl,prop_implic_ctrl,tick]|
  PROP[implic_prop,implic_prop_ctrl,prop_implic_ctrl,prop_sink,eeg,para2,para3,tick])
      |[source_implic,prop_sink,tick]|
  (SOURCE[device_source,source_implic,tick]
      |[tick]|
  SINK[prop_sink,output,tick])))
      |[tick]|
  CLOCK[tick]
  )

where

library     DEVICE,SOURCE,IMPLIC,PROP,SINK,CLOCK endlib

(* Specifications of subsystems are stored in separate files, and are
 omitted here *)

endspec
```

In the above specification, comments are marked between a pair of (* and *), names in uppercase are processes, e.g. IMPLIC, names in lowercase are communication channels, e.g. implic_prop. IMPLIC, PROP, SOURCE, and SINK represent implicational subsystem, propositional subsystem, visual system, and response system respectively. Each process has a set of communication channels attached to it, for example, IMPLIC has six channels. A|[a_1, a_2, ..., a_n]|B denotes a parallel composition of process A and B, which synchronise on channels a_1, a_2, ..., a_n. Unlike mental subsystems, DEVICE denotes the computer systems used in the interface, such as a mini-controller. (Note, some channels, such as eeg, in this specification are only used in the reactive approach, in order to model EEG signals. So, they will be explained in the relevant sections.)

Input channels, such as para1, para2, para3 and para4, set the simulation up with appropriate parameters, many of which were determined from our previous modelling of the AB task [SBB07]. That is, channel para1 sets the delay of buffer movement from Implic to Prop, and channel para2 sets the delay of buffer movement from Prop to Implic. Channels para3 and para4 set the scale factor of the P3 and the criterion for the P3 detection respectively (which will be explained in the relevant section). Their values are generated externally using C programs. Input channel input connects stimuli from the simulation environment to the model. In each run of the simulation, stimuli are randomly generated and fed into the model via input. A run stops by creating a *deadlock* in DEVICE when all items have been presented. The output channel output initiates a subsequent response.[3] All subsystems are synchronised by a global clock CLOCK, which ticks every 4 ms of simulated time.

It can be seen that the LOTOS specification mirrors the overall structure and interaction in our "box and arrow diagram" shown in Fig. 5. The internal structure of each process was also modelled as reusable library files. Note, the following specifications are simplified, but a complete specification can be obtained from the corresponding author on request.

```
process SOURCE[device_source,source_implic,tick,debug] : noexit :=

(* Initialise the delay line with 4 empty constituent representations. *)
let vis_dl : Dline =  (* Specification omitted *) in
Vis[device_source,source_implic,tick,debug](vis_dl)

where

  process Vis[device_source,source_implic,tick,debug](dl:Dline) :noexit :=
    tick?y:Nat!false;
    (* Synchronise with the global clock at the start of each cycle. *)

    ((
    (device_source?ia:Rep; (* Recieve visual stimuli from device and
                              store it in ia (input array). *)
    exit(ia))
        |||
    (source_implic!Get(dl); (* Get() returns the last item in the delay line. *)
    exit(any Rep))
    )

    (* The above processes are performed in parallel, and when both finish,
     the sequential operator >> ensures that stimuli in the input array are
     passed to the next step. *)

    >> accept ia:Rep in

    (
    [(y mod 20) eq 0] ->
     (* Update delay line with the new stimuli when time is a multiple of 20. *)
    (let dl:Dline = Put(ia,Push(dl)) in
```

[3] In reality, the interface has no access to this output signal. Indeed, the output is used by the simulation environment to generate a report. Channels input and output are connected to the simulation environment implemented in C code.

```
    Vis[device_source,source_implic,tick,debug](dl))
      []
    [(y mod 20) ne 0] ->
     (* Otherwise, do noting. *)
    (Vis[device_source,source_implic,tick,debug](dl))
    ))
  endproc

endproc
```

The above specification defines the visual system. The main component in this system is a delay line that holds four constituent representations. This delay line captures the amount of time (i.e. 80 ms) for an item to travel from the retina to Implic. (This parameter is justified in the next section.) This process contains another process `Vis`, which evokes itself recursively. On each cycle, it monitors its input channel `device_source` for constituent representations (with type `Rep`). Simultaneously, the last constituent representation in the delay line is sent to Implic via `source_implic`. After the sequential composition >>, the delay line is updated using the most recently received constituent representation. Note, each cycle takes 4 ms, so all channels update every 4 ms. However, the delay line updates at a slower rate of every 20 ms as previously discussed. This is implemented as a conditional choice, which has the following form:

```
[condition1] -> process1
  []
[condition2] -> process2
```

It is also noteworthy that all mental subsystems have the same overall structure. That is, a process is running recursively within each subsystem. The process is divided into two parts. The first part performs communication between mental subsystems or between the user and the device. The second part updates the internal state of each mental subsystem based on their input data. These two parts may update at different rates. Finally, all subsystems are synchronised by the global clock.

```
process IMPLIC[in1,source_implic,imp_prop,imp_prop_ctrl,prop_imp_ctrl,tick] : noexit :=

  (* initialise the parameter and the delay line with 10 empty constituent
    representations. *)
  let imp_dl : Dline =  (* Specification omitted *) in
  in1?imp_buf_mov:Nat; (* Recieve parameter from external program. *)
  Imp[in1,source_implic,imp_prop,imp_prop_ctrl,prop_imp_ctrl,tick](imp_dl,imp_buf_mov)

  where

  process Imp[in1,source_implic,imp_prop,imp_prop_ctrl,prop_imp_ctrl,tick]
          (dl:Dline,imp_buf_mov:Nat) : noexit :=

    tick?y:Nat!finish;
       (
        (source_implic?ia:Rep; (* Stimuli from visual system. *)
         exit (* values omitted *))
        |||
        (* Input the control signal from Prop. This signal and stimuli
          determine buffer movement. *)
        (prop_imp_ctrl?c:Bool;
         exit (* Values omitted *))
        |||
        ([buffer = true] ->
            (
            (* When Implic is buffered, it assigns salience to all its constituent
               representations according to a predefined threshold.
```

```
                The last item in the delay line is passed to Prop.
                Specification omitted.*)
               imp_prop!Get(dl);
               exit (* Values omitted *)
            )
            []
          [buffer = false] ->
            (
            (* When Implic is in direct mode, i.e. not buffered, it simply
               copies the item to Prop without evaluating salience. *)
              imp_prop!Get(dl);
              exit (* values omitted *)
            )
        )
        |||
        (* Sends the control signal to Prop. *)
        (imp_prop_ctrl!change;
        exit (* Values omitted *))

        >> accept (* Values omitted *) in

        (* It gives up the buffer after a delay when a salient item has been detected.
           This is coordinated with Prop by exchanging a set of control signals.
           The buffer movement delay is also updated via channel in1 once the buffer
           switches. Specification omitted. *)
        (
        [(y mod 20) eq 0] ->
        (
          let dl:Dline = Put(ia,Push(dl)) in
          Imp[in1,source_implic,imp_prop,imp_prop_ctrl,prop_imp_ctrl,
              tick](dl,imp_buf_mov)
        )
           []
        [(y mod 20) ne 0] ->
          (Imp[in1,source_implic,imp_prop,imp_prop_ctrl,prop_imp_ctrl,
               tick](dl,imp_buf_mov))
        )
       )
    endproc

endproc
```

The above specification defines Implic. Its basic structure is similar to the visual system except that it has more communication channels. The main function of this subsystem is salience assessment and, the resulting, switching of the buffer. However, their implementation is omitted in this paper. Prop has the same overall configuration, so its specification is also omitted in this paper. The following specification defines a global clock which ticks every 4 ms, and the variable y stores the current time in milliseconds. The specifications of the response system and mini-controller are not shown in this paper due to space considerations.

```
process CLOCK [tick] (y:Nat) : noexit :=
  tick!y?finish:Bool; (* Broadcast time. *)
  (
  let z:Nat = 4 in  (* Time passes by 4ms every cycle. *)
  CLOCK[tick](y+z)
  )
endproc
```

The LOTOS model was simulated using the CADP toolbox [GLM02]. The specifications of the behaviour and the data types (written in ACT ONE [EFH83]) were transformed to C code using the EXEC/CAESAR environment [GVZ01]. Psychological simulations often obtain good results from the average of thousands of runs. So, C code generated using EXEC/CAESAR can be performed efficiently. This also allows probabilistic behaviour to be modelled in C and fed into the model via externally visible channels, such as `para1`, `para2`, `para3` and `para4` (that are defined in the top-level specification).

4. Modelling the attentional blink and electrophysiology

4.1. How the model blinks

The general idea that attention deployment is governed by an initial glance at generic meaning and then is optionally pursued by more detailed scrutiny of referentially specific propositional meaning is captured here by two stages of buffering with distributed control. The subsystem that is buffered decides when the buffer moves and where it moves to. In the majority of real life situations, stimuli do not arrive as rapidly as in AB experiments, so Implic and Prop will normally interpret the representation of the same item 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 buffer movement dynamic provides the underlying mechanism for the blink, i.e.

- When in response to the key-distractor being found to be implicationally salient, the buffer moves from Implic to Prop, salience assessment cannot be performed on a set of words (i.e. a portion of the RSVP stream) entering Implic following the key-distractor. Hence, when these implicationally uninterpreted words are passed to Prop, propositional meaning, which builds upon coherent detection of implicational meaning, cannot be accessed. If a target word falls within this window, it will not be detected as implicationally salient and thus will not be reported.
- There is normally lag-1 sparing in key-distractor AB experiments, i.e. a target word immediately following the key-distractor is likely to be reported. This arises in our model because buffer movement takes time (i.e. after processing three constituent representations (60 ms), the buffer takes 180–220 ms to move from Implic to Prop,[4] as shown in Fig. 6). Hence, the word immediately following the key-distractor may be implicationally interpreted before the buffer moves to Prop.
- When faced with an implicationally uninterpreted item, Prop is no longer able to assign salience and the buffer has to return to Implic to assess the implicational meaning. (There is also a delay of buffer movement from Prop to Implic in the range between 20 and 740 ms after detecting three uninterpreted items at Prop, as shown in Fig. 6.) Then, Implic assigns salience to its constituent representations again. After this, targets entering the system will be detected as both implicationally and propositionally salient and thus will be reported. Hence, the blink recovers.

The distribution from which the buffer movement delay was sampled sets the shape of the blink onset and recovery. The results of the simulation were compared to human performance in order to verify our theories of temporal attention [SBB07]; see Fig. 3.

4.2. Modelling the electroencephalogram

In the context of cognitive neuroscience, computational modelling is a critical method for the validation of theories. To a large extent, models have been used to reproduce observable human behaviour, e.g. reaction times. (In fact, our model was initially designed to accurately reproduce the behaviour data obtained from the AB experiment [SBB07].) However, the ability to reproduce electrophysiological recordings has been overlooked by the cognitive modelling community. In addition to modelling behavioural data, our model also simulates eletrophysiological data and compares it with human data. This could further verify the cognitive theory encapsulated in the model.

[4] 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. Hence, the overall delay of a subsystem is the sum of basic salience assessment delay and buffer movement delay.

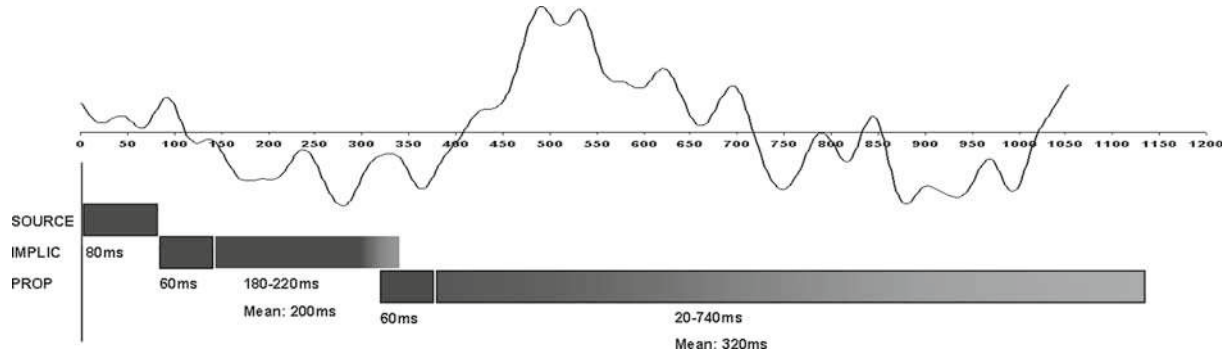

Fig. 6. Time course of the model. This figure compares the time course between a typical human EEG (*top*) with the model (*bottom*). For the model, the passage through the model of a single item is depicted; *zero* indicates stimulus onset. The *solid filled boxes* represent periods that are determined, i.e. fixed in each run of the simulation. (In the current model, they are also constant across runs.) The *gradient filled boxes* represent periods that are stochastic, i.e. randomly generated for every single simulation run. The *gradient areas* approximately illustrate the distributions of their corresponding parameters

Moreover, models can be modified or given different inputs in order to predict the results of the underlying neural physiological activities and the corresponding electrophysiological recordings.

In the context of human–computer interaction and particularly where brain–computer interfaces are under consideration, simulating electrophysiological data is potentially useful in order to investigate designs of the system. Ultimately, such techniques may provide another application of the model. That is, it allows rapidly testing parameters of BCIs using computational models rather than running expensive and time consuming human experiments.

In the context of this paper, we are interested in generating *virtual* EEG signals (the counterpart of the human EEG) from the model. (The term virtual EEG or ERP was firstly introduced by Craston et al. [CWC08].) The model simulates EEG responses, and computes the size of the P3 component using the same method as described in [WCB06]. That is, the P3 size is calculated by the total area under the curve for a time window from 300 to 700 ms after the onset of the target. (P3 size was also normalised to a baseline level, which will be explained in the relevant section.) Then, the interface can use the P3 size to predict if the user has seen a target. If the P3 is big enough when compared to a predefined criterion, the interface displays the next item, otherwise, it repeats the previous one. We intend to simulate the same mechanism using our cognitive model of the human operator and evaluate the performance of the interface.

In our model, the EEG signal is generated from Prop and used by a mini-controller, i.e. `DEVICE`, which sits outside the brain, see Fig. 5. There are three basic principles related to the generation of the EEG signal: P3 time course, P3 amplitude, and EEG noise. We discuss each in turn.

- **Time Course**. In general, the P3 starts when Prop is buffered, since working at Prop can be equated with referential short-term memory[5] updating. In the model, the buffer moves to Prop when a salient item is detected at Implic. As we have discussed in the previous section, the buffer moves to Prop at around 260 ms after a salient item has arrived at Implic (the time to extract implicational meaning plus the time to relocate attention), as shown in Fig. 6. There is also neurophysiological evidence that signals related to visual processing take around 80 ms to travel from the retina to occipital areas in the brain [SWH98], as shown in Fig. 6. (As previously explained, this delay is modelled in the visual system in the model.) Thus, in our model, the P3 starts at around 340 ms, as shown in Fig. 6, which matches the human P3 very well. The P3 duration is set by the amount of time that the buffer stays at Prop, i.e. around 380 ms, as shown in Fig. 6, which also matches the duration of the P3 observed in humans. See Figs. 4 and 6 for a comparison.
- **P3 Amplitude**. At any point of time, the amplitude of the P3 is set by the number of constituent representations within the propositional delay-line that are both implicationally and propositionally salient. Basically, the more salient representations being processed by Prop, the stronger the P3. The maximum amplitude of the noise free P3 is directly determined by the speed of buffer movement, i.e. the faster the buffer moves to Prop, the more salient constituent representations can be processed at Prop, consequently the bigger the P3. Once

[5] In many situations, the two concepts: short-term memory and working memory are interchangeable.

the salient constituent representations have left Prop, the P3 starts to decay. Thus, the P3 terminates when the buffer shifts away from Prop. The speed of the decay is set to the reciprocal of the amount of time that the buffer stays at Prop, multiplied by the maximum P3 amplitude:

$$v = \frac{m}{t}$$

where m denotes the maximum amplitude of the noise free P3; t denotes the buffer movement delay from Prop to Implic, and it ranges from 20 to 740 ms, as shown in Fig. 6; v denotes the speed of the decay.

- **EEG Noise**. Broadly speaking, there are a number of sources of noise in EEG. Some are generated by participant's neural physiological processes and others are generated externally, e.g. EEG recording instruments or other environmental influences. After the EEG has been recorded, systematically induced artefacts from eye movements and electrical influences can be removed. However, other kinds of noise that are not systematically related remain in the EEG signals. In many cases, such as using EEG in the practical domain, reliable conclusions cannot be drawn from EEG without carefully considering the characteristics of such noise (or background activities) since it always presents in a single trial.[6] Thus, any model should simulate such noise in the virtual EEG too. We use a simple approach to introduce physiologically plausible noise. Specifically, we took large amounts of EEG recordings from participants watching an RSVP stream containing only distractors, removed artefacts such as those caused by eye movement, and superimposed these data onto the noise free virtual EEG.

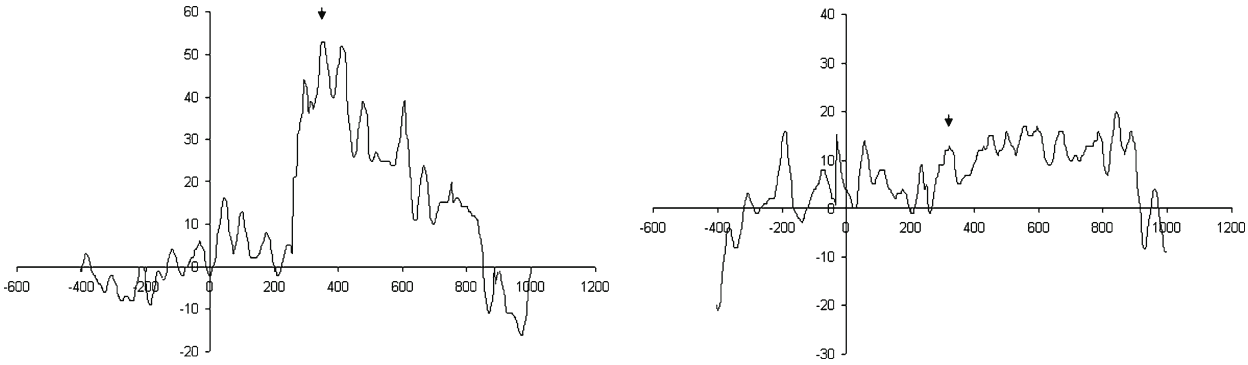

Fig. 7. Examples of virtual P3s generated from model simulations (targets were presented at time 0, the unit for the y-axis is mV and the x-axis is ms). Note that the simulation only models the P3, and so other components of a typical EEG, such as the N1 and the N2, are notably absent from these simulations. As a result, the virtual EEGs generated here do not necessarily look like real EEGs other than in the presence of a large positive deflection at around 300–700 ms

As previously discussed, the P3 varies over individual people and as a function of many other experimental factors. So, the model also needs to capture variation in P3 amplitude in order to reflect individual differences and other sources of variance. To deal with this, we introduce a parameter to the final EEG signal eeg. This parameter is a scale factor that modulates the amplitude of the P3, see the following equation for how the model computes the EEG.

$$\forall x \in [-400, 1000] \cdot \ eeg(x) = \gamma \times P3(x) + \omega(x)$$

where $P3$ denotes the noise free EEG generated by the model, γ is the scale factor, ω denotes the EEG noise, and $[-400, 1000]$ are time points of a single EEG trial. Figure 7 shows two examples of virtual P3s generated from the model simulations (targets were presented at time 0). The diagram on the left hand side was generated using a scale factor of 6, and the one on the right-hand side was generated using a scale factor of 2. When compared with Fig. 4, we can see that the virtual P3s have similar characteristics to the human P3s. Moreover, the variability of the P3 is captured by the scale factor. Note a scale factor of zero gives a noise only EEG signal.

[6] After averaging across multiple trials, a technique called ERP, such noise could be removed or reduced. However, real world applications often work with single trial EEG data.

5. Modelling stimuli rich environments

The previous section suggests that our model captures attentional deployment and some aspects of the underlying electrophysiological activities of humans in the laboratory environment. As previously argued, we believe that the AB paradigm is generalisable to practical settings and the model could well provide insights into interface design issues. Stimuli in the environment can be classified by three types: high salient (i.e. targets), low salient, or non-salient stimuli (i.e. background items). In many situations, only the human operator can decide what is truly significant, however, the interface designates targets as those items that are expected to be high priority. We simulate a situation in which information is only briefly available (e.g. 120 ms per item presentation rate) which allows for a greater density of information to be presented. We assume the simulation only presents a fixed number of items (maximum of 200 in the simulation) in order to ensure that the user perceives stimuli in a timely fashion. In addition, all stimuli appear at the same spatial location and have observable bursts on all time scales. Such assumptions are necessary to enable us to link psychological theory to practical simulation. Nonetheless, it is a starting point to evaluate more complex systems in the future when one or more assumptions made here are violated. We will revisit and discuss those assumptions in the concluding section.

To simulate the arrival of uncoordinated stimuli to the operator (sometimes called self-similar traffic[7]), we use an algorithm with parameters that control the pattern of stimulus arrival. There are a number of methods to generate self-similar traffic, such as the multiple ON/OFF source aggregation process [Gom98]. However, many of these are complex and hard to implement. We choose a particular model, called the *b*-model [WMC02]. It is a simple model with very few parameters, but it can generate self-similar and bursty traffic for any given time scale. Its parameters include the number of high salient items (i.e. targets), the number of low salient items, and the burstiness of the traffic, which is characterised using a single parameter b that ranges between 0 and 0.5. The smaller b is, the more bursty the stimuli. A b parameter of 0.5 results in items being evenly distributed in the stream. A complete description of the model is presented in [WMC02]. Intuitively speaking, the *b*-model randomly divides items into two halves and distributes them into two equal periods of time. This process repeats on each of these two parts of the presentation until all items have been distributed.

We would like to use *b*-models to simulate how stimuli appear in stimulus rich environments. However, such stimuli are different from those arising from RSVP streams, since multiple events/stimuli can arrive simultaneously. Thus, we assume that the interfaces have a buffer,[8] which stores all events and presents them to the user serially. The total number of high and low salient stimuli generated is set to 2^n, where n is a parameter in the *b*-model, called the aggregation level. The aggregation level determines the length of the non-background stimuli, i.e. an aggregation level of 2 generates 4 items; an aggregation level of 6 generates 64 items, and the stimulus stream contains both high and low salient items. An example of such stimuli is shown in Fig. 8a. Note, the interface outputs background stimuli when all salient items in the environment have been presented. Equipped with this presentation algorithm, we vary a number of key parameters of the environment, and measure the probability of reporting target identity after applying either a constructive or a reactive approach to improve the interface.

6. Evaluating the constructive approach

Our model of the AB task suggests that the ability to report the subsequent item is significantly impaired between 100 and 600 ms after the onset of the previous target. Hence, a constructive approach to improve performance is to separate targets, so that the interference between them is minimised. Such an approach ensures that interfaces leave a window between stimuli that are known to be of high salience. This is done by rearranging the order of stimuli so as to insert specified lengths of less important elements of information, which we term background stimuli, in between the more salient items.

The window size is measured by the amount of time between the arrivals of sequential targets. Figure 8b illustrates an example of an AB-aware system with blink window size 5. That is, between every two adjacent targets, there are minimum five intervening slots (which, if possible, are filled with background items). A typical window size is 5, which corresponds to 600 ms. We call such interfaces AB-aware systems, and the original one AB-unaware; see Fig. 8.

[7] There are many models of traffic proposed by computer scientists, such as Poisson arrivals and self-similar models. The traditional Poisson arrivals is not used here, since it is only bursty within a short time scale, but it will smooth out if it is applied to large time scales.

[8] Using the standard computer science meaning here.

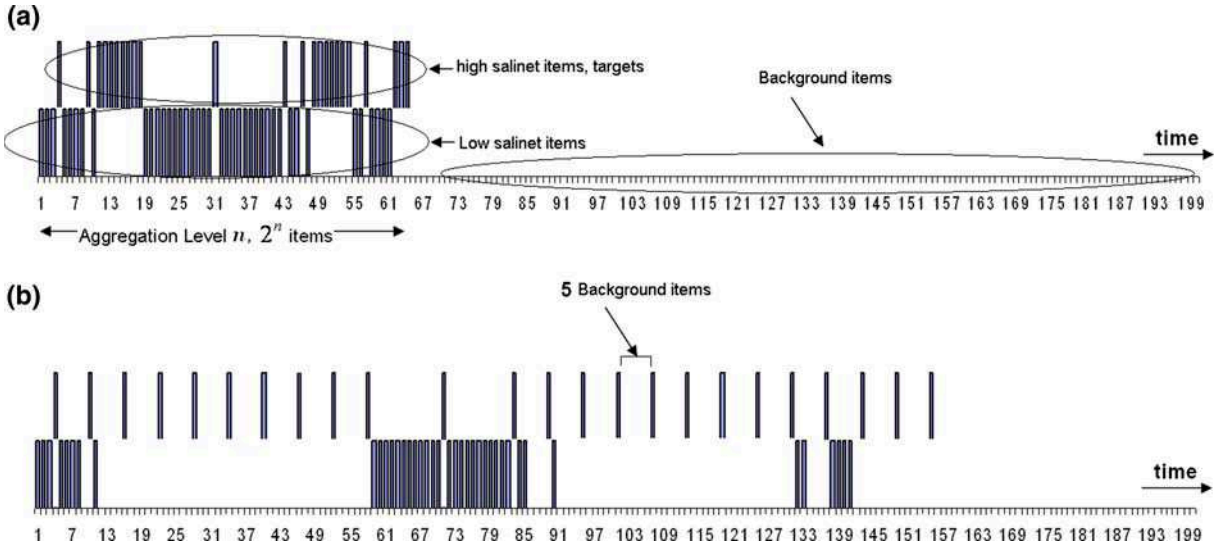

Fig. 8. Examples of stimulus rich environments, with time expressed in terms of the number of items. *Vertical bars* on the top half of each graph represent targets. *Bars* on the bottom half represent low salient items. **a** an AB-unaware system, **b** an AB-aware system. Note, the slightly larger intervals between the 10th, 11th and 12th target in the AB-aware system is due to the inherited burst from the AB-unaware system

We now present the simulation results, focusing on the performance of the interface measured as the average probability of seeing a target. In the first experiment, both the burstiness and the number of targets were varied. In the second experiment, both the burstiness and the aggregation level (total number of high and low salient stimuli) were changed. Finally, we explore how the window size affects the efficiency of the interface.

6.1. Experiment 1

In this experiment, we fixed the aggregation level to 6, window size to 5, and varied the burstiness and the number of targets. Figure 9a shows that the performance of both systems drops as the number of targets increases. This is because targets tend to be close together when the number of targets is high. In general, the performance of the AB-unaware system but not the AB-aware system also decreases when the traffic becomes burstier, since targets are closer to each other and interfere with each other more often. In order to compare these two systems in detail, we subtracted the performance of the AB-unaware system from the AB-aware one. As shown in Fig. 9a, the AB-aware system is less affected by the burstiness because the AB-aware system can reduce bursts and smooth the traffic. It effectively acts as a buffer,[9] maintaining space between salient items. When the number of targets is low (below 20), targets are more separated, and the difference between the two systems is less significant. However, we can see that the AB-aware system is significantly superior when the number of targets is in a medium range (between 20 and 40). The benefits of AB-aware systems start to drop when the number of targets is 40, since the AB-aware system is unable to present all targets within the 200 item's simulation time limit, and keep a sufficient gap (5 intervening items in the current setting) between targets. Moreover, the AB-aware system starts to perform worse than the AB-unaware system when both the number of targets and the burstiness are high.

6.2. Experiment 2

In this experiment, the effect of aggregation level on the performance is investigated. We fixed the number of targets to 20, window size to 5, and varied the burstiness and the aggregation level. Figure 9b shows that the performance of the AB-unaware system improves as the aggregation level increases. This is because the targets

[9] Using the standard computer science meaning here.

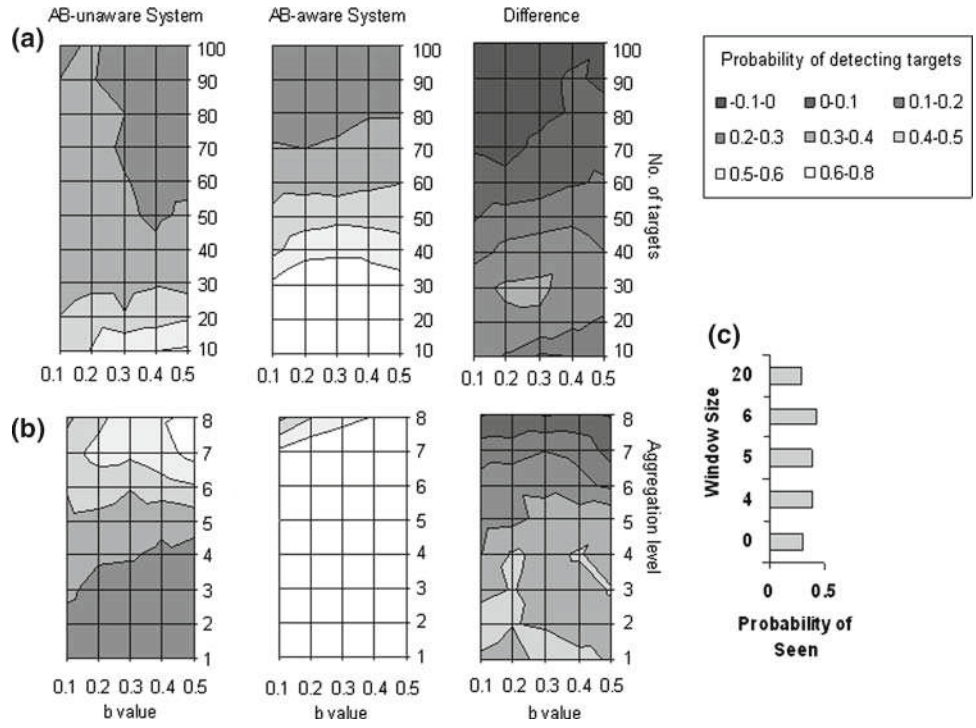

Fig. 9. a, **b** Performance (measured as probability of detecting targets) of AB-unaware and AB-aware systems by varying the number of targets, the aggregation level, and the burstiness (i.e. the b value, with burstiness increasing as b decreases). Note that the x-axis in both **a** and **b** is the b value. The only difference between **a** and **b** in this figure is their y-axis. **c** Performance of the interface with different window sizes of the stimuli. AB-unaware system is a special case of AB-aware system with a window size of 0

are more separated when the aggregation level is high. Performance drops when the stimuli are more bursty. However, the performance of the AB-aware system stays high while increasing either the aggregation level or the burstiness. The difference in performance of these two systems is shown in Fig. 9b. It can be seen that the AB-aware system consistently performs better than the AB-unaware system. Note that the performance did not improve continuously as the aggregation level increases. Accuracy starts to drop at aggregation level 8. This is because high aggregation levels result in targets being separated by larger intervals in general. In addition, our simulation only presents a maximum of 200 items and the rest will be discarded. Thus, increasing the total number of salient stimuli (by increasing the aggregation level) reduces the background period at the end of the traffic (see Fig. 8 for how the AB-aware system uses this period), leaving the AB-aware system less capable to maintain five intervening slots between targets.

6.3. Experiment 3

In this experiment, we repeated the previous two experiments, but this time using different blink window sizes of 4, 5, 6 and 20. Figure 9c shows the performance of the AB-unaware system (which can be seen as a special case of the AB-aware system with a window size of 0) and AB-aware systems using different window sizes. A trade-off can be found. That is, if we want to ensure that the human user perceives targets as accurately as possible by avoiding interference between targets, we should increase the window size. However, large window sizes potentially result in fewer targets being presented within a bounded time. Hence, it becomes necessary to consider the balance between accuracy and urgency.

6.4. Summary

We varied a number of parameters in the interface, and evaluated the performance trade-offs that arise in AB-aware systems using a constructive approach. Such an interface could reduce the effect of burstiness by

smoothing the traffic, and improve the probability of reporting targets by sacrificing urgency. The disadvantage is that this interface could potentially delay the presentation of targets thereby impairing performance (in particular when the aggregation level is high), and affect the absolute timing of events. We also noticed that the blink window size affects the performance of the AB-aware system, and a medium size is suitable for the type of stimuli used here. Using these parameters, target presentation in AB-aware systems is highly regular. In practice, humans would detect such regularity, and then anticipate targets if salient stimuli occur frequently in the environment. However, our model does not capture this. Hence, it is not unreasonable to think that real human performance with AB-aware systems would be even better. Note, when salient stimuli rarely occur in the environment, human's ability to detect the underlying regularity of targets is restricted.

Figure 9 shows that the highest probability of detecting a target is around 0.6 in all experiments, which is close to the baseline performance of AB experiments and reflects a lack of interference between targets. The smallest probability of detecting a target is around 0.2, which is similar to the worst performance during the blink. This reflects the fact that constructive approaches can adjust presentation to correspond with the rhythm of a user's attention, therefore improving the performance, but only up to a certain limit defined by the probability of reporting a singly presented target. In other words, constructive approaches cannot eliminate the noise of biological and mental processes as reflected by the baseline performance in the AB.

The next section considers the reactive approach. An SRRI is shown to improve overall performance by using EEG recordings to determine whether a target has been detected by the human operator.

7. Evaluating the reactive approach

Interfaces use many techniques to increase the inherent salience of stimuli for a human user, such as using warning lights and sounds. However, accidents still occur due to critical information being ignored. Using computer networking as a metaphor, this situation is akin to transmitting information over a communication channel without acknowledgement of receipt. The computer is transmitting information to the user with the intent that it be received, but the system has no way of verifying that this transmission was successfully completed. However, computer–computer interfaces do not generally operate this way. Acknowledgement of packet receipt is a core element of modern networking technology.

The goal of the reactive approach is to record an acknowledgement signal from scalp recorded brain waves to allow re-presentation of missed communications. Electrophysiological neuroscience has identified electrical signals (e.g. P3) produced by the brain when information is encoded into working memory. The P3 could serve this acknowledgement role. In this paper, we explore a novel use of recorded brain signals, to enhance the reliability of information transmission from computer to user.

Such an approach should provide faster feedback than asking a user to explicitly acknowledge individual signals. Wyble et al. [WCB06] have performed a feasibility investigation on SRRIs using EEG. They discovered that two potential EEG measures (reduced EEG power in the alpha band at posterior brain areas and a P3-like deflection over parietal areas as shown in Fig. 4) may be correlated with whether a stimulus has been perceived. Critically, they investigated the possibility of extracting these signals in real-time using compact devices implemented and integrated in a head-mounted display. This raises the possibility that a reactive approach could improve performance above the baseline level of accuracy observed for presentation of a single target. This is because such systems can identify missed targets and re-present them, while in constructive approaches, the user has only one chance to perceive each stimulus.

However, due to the noisy nature of the signal, such EEG feedback may not always accurately predict whether the user has perceived targets. Hence, we could extend our formal model to produce feedback in a similar fashion to EEG signals and use them to investigate the feasibility of such interfaces. In this section, we will evaluate how accurate an acknowledgement should be, in order to make such an approach worthwhile, and search for suitable parameter settings that result in the best performance given realistic noise.

7.1. Real-time P3 detection

As previously explained, the model described here simulates virtual EEG signals, and we can use the virtual P3s [CWC08] as an indication of target detection. In our model, the mini-controller monitors EEG for the presence of a P3 after presenting each target to the user. On detection of a sufficient P3, the system will present the next item, otherwise, it will repeat the previous item. While the interface is waiting for the P3 as an acknowledgment, it

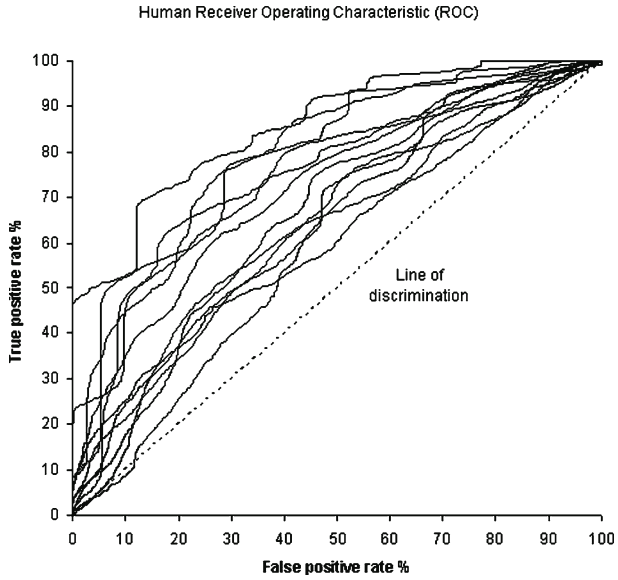

Fig. 10. Receiver operating characteristic (ROC) curves for 12 human participants with variability in the P3 detection criterions. Note, true negative rate and false negative rate do not need to be plotted, as they are fully defined by the true positive and false positive rate (respectively), e.g. true negative rate + false positive rate = 100%

presents background items to the user. In signal detection theory [Sno88], there are four possible outcomes from a binary classifier such as the mini-controller discussed here. A true positive (or hit) occurs if the mini-controller predicts that the user has seen a target, and the user has also correctly reported the target. However, if the user has not correctly reported the target, the mini-controller produces a false positive (or false alarm). Conversely, when the mini-controller predicts that the user did not see a target, and the user did in fact miss it, the response is a true negative (or correct rejection). However, if the user has actually seen the target, the controller's response is called a false negative (or miss).

The mini-controller makes use of an algorithm designed by Wyble et al. [WCB06] for detecting P3s in raw EEG. That is, the mini-controller reads in the EEG data and locates the point of time when the target was presented (or target onset). For each trial, the algorithm computes a baseline level of EEG by averaging electrical activity over an interval between the target onset and the start of the P3. This baseline average is used to normalise the P3 on that trial by subtraction. The amplitude of the normalised P3 is computed as average positive activity within a temporal interval approximately 300–600 ms after target onset. The temporal interval of the baseline and P3 measurements vary for different subjects as configured by pilot work.

This P3 activity can be compared to a threshold to determine whether a P3 did or did not occur on a given trial. The threshold is computed on a per subject basis by averaging individual P3's from a number of trials in which that subject successfully reports a target, and then using one half of the normalised, average P3 amplitude as a detection criterion. When used in real-time, those trials in which the P3 has an area equal or above the criterion are classified by the mini-controller as target seen and the rest as target miss.

The choice of criterion affects the rate of these four different outcomes. For example, a greater criterion results in lower false alarm and hit rates, but higher miss and correct reject rates. However, the sensitivity of the algorithm is independent of the criterion. We can use receiver operating characteristic (or ROC) curves [Met78] to describe how sensitive the algorithm is for detecting P3s as the criterion varies. A ROC curve plots the true positive rate against the false positive rate. Figure 10 shows the ROC curves for 12 human participants who were watching an RSVP stream while the EEG signal was recorded simultaneously. From this graph, we can see that there is considerable variance in the sensitivity of SRRIs due to individual differences. The closer the curve to the line of discrimination, the less sensitive the algorithm will be, thus, the less accurate the acknowledgment will be. However, the closer the curve to the top-left corner of the ROC space, the more sensitive the algorithm will be, thus, the more accurate the acknowledgment will be. Fundamentally, the differences in sensitivity are determined by how far apart the two probability distributions, P3 present vs. P3 absent, are.

Table 1. Comparison of experimental results across 12 human participants with model simulations

Subject	Human d′	Model d′	Scale factor
11	0.39	0.44	3
1	0.48	0.44	3
7	0.53		
6	0.56		
2	0.59		
3	0.61	0.65	4
12	0.73	0.85	5
10	0.93	0.85	5
9	1.07	1.02	6
4	1.09	1.02	6
8	1.21		
5	1.69	1.7	10

The table is ordered by human d′ scores

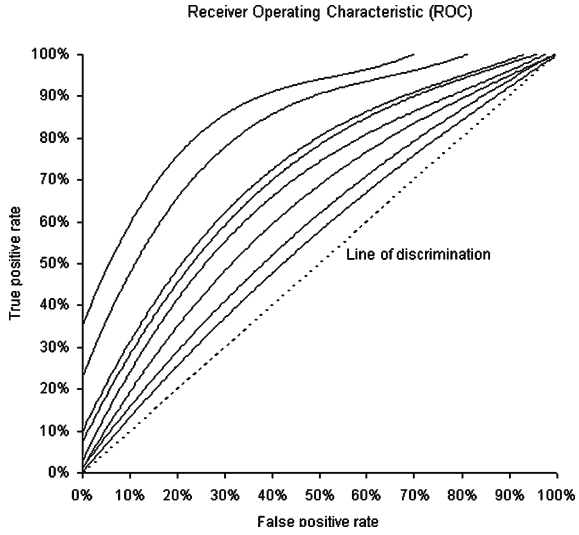

Fig. 11. Receiver operating characteristic (ROC) profiles generated by model simulation with variability in the P3 detection criterions. Each set of data has a different scale factor, which captures differences in discrimination. (From the bottom, the scale factors are 1, 2, 3, 4, 5, 6, 10, 15 and 20.) The ROC curves are actually trend lines fitted to the distribution of data points generated by runs of the simulation

Psychologists often use a measurement called d′ (pronounced d prime) to describe the ability to discriminate these two probability distributions. The relationship between d′ and ROC is that the line of discrimination corresponds to a d′ of zero. When the ROC curve tends toward the top-left corner of the ROC space, d′ tends toward infinity. The d′ scores for 12 human participants are shown in Table 1.[10] They range from 0.39 to 1.69, and the average d′ across all subjects was 0.82.

In this paper, we seek to reproduce similar ROC curves using our cognitive model of the human user as feedback to the SRRI. The interface is configured such that after a target is presented, presentation of further targets is prevented until there has been successful acknowledgment from the virtual EEG signal. In each case, the interface waits for 700 ms before making a decision to either continue normally or to repeat presentation of a "missed target". In our simulations, the baseline and P3 intervals are fixed to ranges of [0–300 ms] post target and [300–700 ms] post target respectively.

Note, as a side effect of this algorithm, there is an implicit window of six items, where no target can be presented. Such an interval between targets ensures that the P3s from multiple items do not overlap. As we vary the criterion for P3 detection from 0.5 to 29 mV, ROC profiles were generated from the model in the same fashion as in human participants. The sensitivity of the SRRI is influenced by the scale factor, see Fig. 11. Table 1

[10] When calculating the d′, we assume that the P3 amplitude is drawn from a normal distribution with a standard deviation of 1.

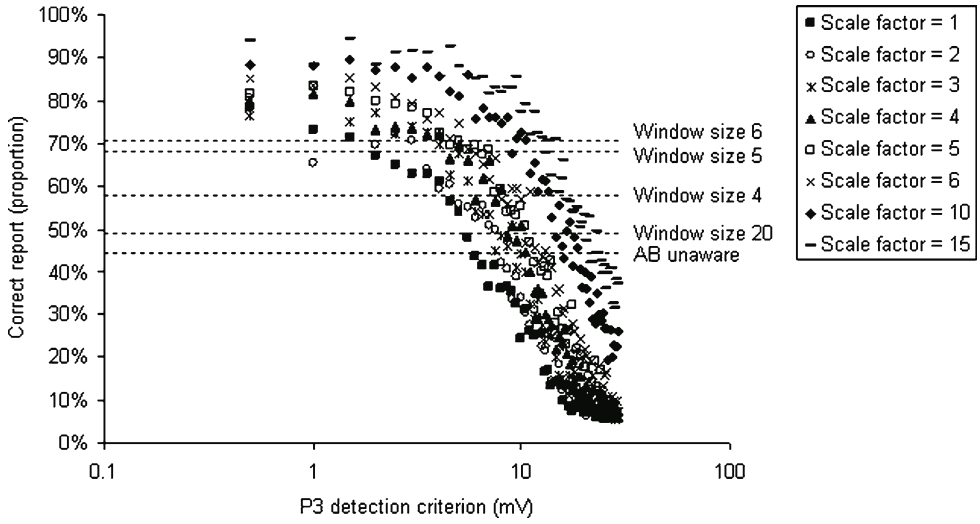

Fig. 12. Performance (measured as probability of detecting the targets) of the reactive approach using EEG feedback with variability in the P3 detection criterion. Each set of data has a different scale factor, which captures differences in discrimination. For comparison, absolute performance of constructive approach is shown for different window sizes

compares model simulations with human d′. It can be seen that human participants can be matched to the model with scale factors between 3 and 10. We have simulated scale factors of 1, 2, 3, 4, 5, 6, 10, 15 and 20. However, scale factors can be any positive real value. Although not shown in this paper, the model could in fact fit human users very accurately, if the full range of the scale factor were explored using suitable search algorithms. Hence, the match between human d′ and model d′ can be improved, e.g. Subjects 2, 6 and 7 could be accurately modelled by a scale factor around 3.5, and Subject 8 could be modelled by a scale factor between 7 and 9.

7.2. Performance analysis

Having reproduced the electrophysiological data of human participants engaged with an SRRI, the model is able to predict the performance trade-offs that would arise when parameters in the model are varied. Figure 12 shows the simulation results measuring the performance of our reactive SRRI as the proportion of targets that have been reported within a bounded time. Note, repeated items are only counted once. In simulations, the aggregation level was set to 6, the number of targets was 20, and the b value was randomly chosen between 0 and 0.5.

Figure 12 shows that when the scale factor is large, the reactive approach is able to perform better than the baseline performance, which cannot be exceeded by the previously described constructive approach. However, the performance drops when the criterion increases. So, it can be seen that a suitable criterion is critical to balance the trade-off between accuracy and urgency when working with deadlines. In our simulation, the miss and false alarm rates influence the urgency and accuracy respectively. Moreover, they are modulated by the criterion, i.e. a greater criterion results in a lower rate of false alarms but higher rate of misses. When a high criterion is used, targets are more likely to be re-presented, increasing the likelihood that they are detected. However, more seen targets may be re-presented unnecessarily. Hence, the interface runs out of time before most of the targets are ever presented.

7.3. Summary

Our simulation accurately reproduced the electrophysiological recording (i.e. P3s in the EEG signal) from human participants while they are interacting in an SRRI. This suggests that the model can be used to investigate interface design with properties close to those discussed here. We have simulated the reactive approach with different parameter settings, and evaluated the performance trade-offs that arise when EEG signals are used

as an acknowledgment of the user having attended to information presented at an interface. The simulation confirmed our prediction that a reactive approach may improve the performance of SRRIs beyond baseline performance of AB experiments.

8. Comparison of constructive and reactive approaches

In this section, we compare the performance of constructive and reactive approaches. As we have previously explained, increasing the scale factor results in larger P3s in the EEG, thus, they are easier to detect. Consequently, performance improves as we increase the scale factor as shown in Fig. 12. Another observation is that in general the reactive approaches perform much better than constructive approaches (shown as horizontal dashed lines in Fig. 12) when a low criterion is used. (Lines corresponding to the performance of an AB-aware system with blink window sizes of 4, 5, 6, 20 and the performance of the AB-unaware system are also shown in the same graph.) As previously explained, a reactive approach with a very low criterion value and a low scale factor is similar to a constructive approach with a blink window size of 6. So, a small scale factor does not improve SRRIs dramatically from constructive approaches when the criterion is low. Moreover, if the scale parameter is small and the criterion is high, the system performs much worse than constructive approaches because many target-seen trials are classified as misses, and the system repeats the same targets many times.

More importantly, the simulation also illustrates the relative value of constructive and reactive approaches. When the sensitivity of P3 detection or d′ is low, there is no significant improvement for a reactive approach relative to a constructive one. The much more complicated and expensive SRRI would only be worthwhile when the accuracy of P3 detection is sufficiently high. There is no doubt that an SSRI is capable of improving accuracy, however, it does not necessarily improve urgency compared to constructive approaches for the following reasons. First, in the current setting, SRRIs require a minimum of 700 ms interval between any two targets to examine their P3's in succession. As a result, the presentation rate of SRRIs cannot exceed about one target per second. Second, if a target was missed by the user and the SRRI re-presented the item, the urgency of the interface is further reduced. These limitations could be potentially overcome by developing more advanced EEG recording technologies to achieve a higher d′, and reducing the time required to assess the P3.

Despite these limitations, SRRIs may be useful for many applications, such as the cockpit environment, where the losses caused by missing critical targets can be catastrophic. Another question for designing such a system is the choice between various strategies, e.g. constructive or reactive approaches. When making such a choice, the model is able to evaluate the relative value of different interfaces. In particular, it can model the user at an individual level using the scale factor. Hence, we could, in principle, determine whether SRRIs should be used for a specific person depending on the quality of their EEG signals. Taking subject 5 (see Table 1) as an example, he scored the highest d′ among all 12 participants. As shown in Fig. 12, the model predicts that his performance (corresponding to scale factor 10) will reach over 90% accuracy when a suitable criterion is used. However, subject 11 with a quite low d′ is not a good candidate for the use of a reactive interface, which would perform similar to constructive approaches with a blink window size of 6.

9. Discussion and future work

9.1. Formal methods: process algebra

It is clear that the choice of modelling notation dramatically affects the value of (even the ability to complete) a model of a psychological theory. In a very general sense, selection of an appropriate modelling notation can be a major enabler of problem solving. It is thus natural to believe that the identification of appropriate modelling notations, which offer a suitable level of abstraction, can aid the progress of cognitive modelling. The most attractive feature of process algebras, in this respect, is that they allow the formal specification and computational exploration of the properties of psychological theories formulated in terms of interaction between subsystems. As we have previously argued, it has been difficult to accurately express such theories in production systems or neural networks. Furthermore, models based upon hierarchical control theory [Car98] or "box and arrow diagrams" can be given an explicit and direct computational realisation in process algebras. To be more explicit, we would emphasise the following benefits of such approaches:

- A pragmatic reason for preferring distributed control is, we would argue, that the resulting computational model more directly reflects the component structure of the psychological theory. For example, in EPIC's model of Baddeley and Hitch's Working Memory theory there is no direct analogue of the central executive [KMM99]. The central executive function arises from programming the cognitive processor with particular productions. This places a level of indirection in the computational modelling of the working memory theory, which can be a barrier to psychologist's interpretation of the resulting model. In contrast, the component structure of the working memory model could be directly represented in a process algebra.
- Although our work to date has not directly confirmed this, we postulate that another advantage of the modular (distributed control) approach is that neuropsychological evidence can be more easily modelled. In simulating lesion studies, a more direct mapping between the structure of computational models and psychological and brain-level theories, should make it easier to localise the appropriate site of damage in the model.
- Our hypothesis is that for many forms of psychological-level modelling, the sort of techniques we have been advocating are at a similar level of abstraction to that at which psychologists work. In particular, not only is distribution of control valuable, but other aspects of modelling techniques are also important means of developing more adequate computational models of mind and brain. For example, process algebras come with rich combinatorial symbol systems and data structuring mechanisms. We have also argued that hierarchical decomposition of components is important in supporting macro-level modelling and that abstract specification can (potentially) be used to tackle the irrelevant specification problem.

The history of computational modelling in psychology has been an extremely rich one. In particular, traditional AI techniques and connectionism have made profound contributions. The former of these still offers the most complete architectural-level explanation of the mind, and is still the only realistic modelling technique when macro-level constraints on the mind need to be brought into play. In contrast, the connectionist revolution in psychology has yielded many influential models, which have elucidated specific cognitive phenomena. That revolution has lead to a profound re-evaluation of theories of innate cognitive capacities [EBJ96] and helped to bridge the gap between modelling the mind and modelling the brain [Rol98]. However, despite the strengths of these approaches, we nonetheless believe that new modelling techniques from computer science can also advantageously be employed in computational modelling of the mind. These would complement existing techniques, being specifically targeted at psychological level modelling, in which it is advantageous to directly represent distributed control of mental mechanisms.

9.2. Modelling human–computer interaction

In summary, our model offers a detailed psychological description of human temporal attention. It has been demonstrated elsewhere [SBB07] that this model accurately reproduces a large set of behavioural data in Barnard's key-distractor AB task. Hence, we believe that this model should, to some degree, help us to understand and predict the effect of the temporal bottleneck of attention when humans operate in stimulus rich environments. In addition, the model is capable of accurately simulating both human behaviour and electrophysiology data, such as P3s in the EEG signal. This enables rapid testing of interface designs. Many issues can be investigated directly using model simulations, e.g. determining suitable parameter settings and choosing feasible strategies for improving the interface. The methodology of generating EEG data introduced here could be generalised beyond SRRIs, and applied to other types of brain–computer interfaces.

A further contribution of this paper has been to give a concrete illustration of the use of process algebra in computational modelling of a mental process and its associated eletrophysiological data. Furthermore, through our use of distributed control, attentional constraints (in particular, its sequentiality) have been modelled as emergent properties of inter-subsystem interaction with no recourse made to a centralised focus of attentional or supervisory control. Attentional resources can be allocated and moved around amongst modules. However, this movement is constrained since the buffer can only be at one subsystem at one instant.

We should also note that both this model and the simultaneous type, serial token model of temporal attention and working memory (or the STST model) [Bow07, Wyb05] can generate virtual EEG or ERP data. These two mechanisms differ dramatically: Craston et al. [CWC08] obtained ERP signals by summing up the (weighted) neural activities from specific layers in the neurally detailed STST model, but our approach is directly to convolve human EEG noise to the activation of a mental subsystem, i.e. Prop. An advantage of STST, in terms of generating virtual EEG or ERP data, is that the modelling notation, i.e. neural networks, directly mirrors the biological mechanism underlying human EEG. Moreover, there are relatively few hidden parameters that affect the virtual

EEG. However, currently the STST model is limited in its capacity to generate an EEG signal that closely resembles the human EEG in terms of noise and other variations. This is not an issue for ERP, but cannot be ignored for modelling the brain–computer interface, which requires the analysis of single trial EEG in real-time. The method in our model overcomes this limitation by directly extracting noise from human EEG. It captures the complex variations in humans and allows accurate predictions to be made from model simulations.

9.3. Future work

In Sect. 5, assumptions were made for the stimulus rich interfaces we considered, in order that the AB paradigm would be directly applicable to our investigation. Now, we will discuss to what extent these assumptions hold in real life applications. It should be noted that most AB experiments have very small numbers of targets. However, in some cases, human users have to cope with much larger numbers of targets with the likelihood that working memory capacity will be exceeded. We have set this issue aside in our recent studies, but it could have a dramatic influence on performance. The effect of such cognitive load awaits future investigation. Also, effects of context or of schematic fit may be important. Note, for example that AB experiments use discrete items, whereas in many situations visually presented items are going to be part of a more or less predictable knowledge structure. Indeed, this may be a really interesting area to investigate in the future. Do humans pick up schema discrepant material more readily than schema conforming stimuli? Although we tried to minimise our assumptions, many aspects of system design have been ignored in the current study, for example, spatial attention, contextual influence, multimodal input, and so on. Hence, we hope future studies will address these interesting issues. However, a complete validation of our theory and predictions from such abstract models would require intensive experiments with real participants.

Although we have not so far performed any formal verification of the model, the notation and tools used in this paper support bisimulation checking and model checking of various properties. For example, we could potentially verify that given sufficient accuracy in the P3 detection and enough time to repeat the target until it has been detected, the model is equivalent to a stop and wait protocol in terms of bisimulation. Alternatively, liveness properties, such as that every presented target is eventually detected, could also be checked using the CADP tool. However, due to the large data set and the random/non-deterministic behaviour embedded in the current model, we can postulate that the verification can only be performed on a somewhat reduced abstraction/approximation of the current model in order to avoid the pitfall of state space explosion. A possible solution to such problems may be using stochastic process algebra [BBD01], or timed process algebra [BMM02]. These are important research topics to be explored in the future.

Acknowledgements

We are indebted to anonymous reviewers for their comments on the previous version of this manuscript. We also thank Patrick Craston, Srivas Chennu and Dell Green for their contribution to the collection and analysis of the EEG data. The UK Engineering and Physical Sciences Research Council supported this research (grant number GR/S15075/01). The participation of Philip Barnard in this project was supported by the Medical Research Council under project code U.1055.02.003.00001.01.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

[ABM93] Abeles M, Bergman H, Margalis E, Vaadia E (1993) Spatiotemporal firing patterns in the frontal cortex of behaving monkeys. J Neurophysiol 70:1629–1638

[And93] Anderson JR (1993) Rules of the mind. Erlbaum, Hillsdale

[And05] Anderson AK (2005) Affective influences on the attentional dynamics supporting awareness. J Exp Psychol General 134(2): 258–281

[BMM02] Baeten JC, Middelburg CA, Middelburg K (2002) Process algebra with timing. Springer, New York

[Bar99] Barnard PJ (1999) Interacting cognitive subsystems: modelling working memory phenomena within a multi-processor architecture, Models of Working Memory: Mechanisms of active maintenance and executive control, pp 298–339
[BST04] Barnard PJ, Scott S, Taylor J, May J, Knightley W (2004) Paying attention to meaning. Psychol Sci 15(3):179–186
[Bar03] Barnard PJ, Bowman H (2003) Rendering information processing models of cognition and affect computationally explicit: distributed executive control and the deployment of attention. Cogn Sci Q 3(3):297–328
[BRB05] Barnard PJ, Ramponi C, Battye G, Mackintosh B (2005) Anxiety and the deployment of visual attention over time. Vis Cogn 12(1):181–211
[BPS03] Beverina F, Palmas G, Silvoni S, Piccione F, Giove S (2003) User adaptive BCIs: SSVEP and P300 based interfaces. Psychnology 1(4):331–354
[Bol88] Bolognesi T, Brinksma E (1988) Introduction to the ISO specification language LOTOS. Comput Netw ISDN Syst 14910:25–29
[Bon99] Bond AH (1999) Describing behavioural states using a system model of the primate brain. Am J Primatol 49:315–338
[BBD01] Bowman H, Bryans JW, Derrick J (2001) Analysis of a multimedia stream using stochastic process algebra. Comput J 44(4):230–245
[Bow99] Bowman H, Faconti G (1999) Analysing cognitive behaviour using LOTOS and Mexitl. Formal Aspects Comput 11:132–159
[Bow06] Bowman H, Gomez RS (2006) Concurrency theory, calculi and automata for modelling untimed and timed concurrent systems. Springer, Berlin
[Bow07] Bowman H, Wyble B (2007) The simultaneous type, serial token model of temporal attention and working memory. Psychol Rev 114(1):38–70
[Car98] Carver CS, Scheier MF (1998) On the self-regulation of behaviour. Cambridge University Press, Cambridge
[Chu95] Chun MM, Potter MC (1995) A two-stage model for multiple target detection in rapid serial visual presentation. J Exp Psychol Hum Percept Perform 21(1):109–127
[Cil93] Cilliers PJ, Van Der Kouwe AJW (1993) A VEP-based computer interfcae for C2-Quadriplegics. Engineering in Medicine and Biology Society, 1993. In: Proceedings of the 15th Annual International Conference of the IEEE, vol 15(3), pp 1263–1624
[CFS96] Cooper R, Fox J, Farringdon J, Shallice T (1996) A systematic methodology for cognitive modelling. Artif Intell 85:3–44
[CWC08] Craston P, Wyble B, Chennu S, Bowman H (2008) The attentional blink reveals serial working memory encoding: evidence from virtual and human event-related potentials. J Cogn Neurosci (in press)
[Don81] Donchin E (1981) Presidential address, 1980. Surprise!...Surprise?. Psychophysiology 18(5):493–513
[DBD98] Duke DJ, Barnard PJ, Duce DA, May J (1998) Syndetic modelling. Hum Comput Interact 13(4):337–393
[EFH83] Ehrig H, Fey W, Hansen H (1983) ACT ONE—an algebraic specification language with two levels of semantics, ADT
[EBJ96] Elman JL, Bates EA, Johnson MH, Karmiloff-Smith A, Parisi D, Plunkett K (1996) Rethinking innateness: a connectionist perspective on development, a Bradford book. MIT Press, Cambridge
[Eri81] Erickson TD, Mattson ME (1981) From words to meaning: a semantic illusion. J Verbal Learn Verbal Behav 20:540–551
[Fod88] Fodor JA, Pylyshyn ZW (1988) Connectionism and cognitive architecture: a critical analysis. Cognition 28:3–71
[GLM02] Garavel H, Lang F, Mateescu F (2002) An overview of CADP 2001. EASST Newsl 4:13–24
[GVZ01] Garavel H, Viho C, Zendri M (2001) System design of a CC-NUMA multiprocessor architecture using formal specification, model-checking, co-simulation, and test generation. Springer Int J Softw Tools Technol Transf (STTT) 3(3):314–331
[Gom98] Gomez ME, Santonja V (1998) Self-similiary in I/O workload: analysis and modeling. In: Workshop on Workload Characterization
[Gri05] Grill-Spector K, Kanwisher N (2005) Visual recognition: as soon as you know it is there, you know what it is. Psychol Sci 16(2):152–160
[Hoa85] Hoare CAR (1985) Communicating sequential processes. Prentice-Hall, London
[KMM99] Kieras DE, Meyer DE, Mueller S, Seymour T (1999) Insights into working memory from the perspective of the EPIC architecture for modelling skilled perceptual-motor and cognitive human performance, Models of Working Memory, Mechanisms of Active Maintenance and Executive Control. Cambridge University Press, New York, pp 183–223
[Lev79] Leventhal H (1979) A perceptual-motor processing model of emotion. In: Pilner P, Blankstein K, Spigel IM (eds) Perception of emotion in self and others, vol 5. Plenum, New York, pp 1–46
[LHB00] Levine SP, Huggins JE, BeMent SL, Kushwaha RK, Schuh LA, Rohde MM (2000) A direct brain interface based on event-related potentials. IEEE Trans Rehabil Eng 8(2):180–185
[Mac63] Mackworth J (1963) The duration of the visual image. Can J Paychol 17(1):62–68
[MFP97] Maki WS, Frigen K, Paulsen K (1997) Associative priming by targets and distractors during rapid serial presentation. J Exp Psychol Hum Percept Perform 23:1014–1034
[MKH03] Meinicke P, Kaper M, Hoppe F, Heumann M, Ritter H (2003) Improving transfer rates in brain computer interfacing: a case study. Adv Neural Inform Process Syst 15:1107–1114
[Met78] Metz CE (1978) Basic principles of ROC analysis. Semin Nuclear Med 8:283–298
[Mey97] Meyer DE, Kieras DE (1997) A computational theory of executive cognitive processes and multiple task performance: Part 1. Basic mechanisms. Psychol Rev 104:3–65
[Mil89] Milner R (1989) Communication and concurrency. Prentice-Hall, Hemel Hempstead
[New90] Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge
[Rei00] O'Reilly RC, Munakata Y (2000) Computational explorations in cognitive neuroscience: understanding the mind by simulating the brain, a Bradford book, MIT Press, Cambidge
[RSA92] Raymond J, Shapiro K, Arnell KM (1992) Temporary suppression of visual processing in an RSVP Task: an attentional blink. J Exp Psychol Hum Percept Perform 18(3):849–860
[Rol01] Rolls ET, Stringer SM (2001) A model of the interaction between mood and memory. Netw: Comput Neural Syst 12:89–109
[Rol98] Rolls ET, Treves A (1998) Neural networks and brain function. Oxford University Press, Oxford
[RMP86] Rumelhart DE, McClelland JL, the PDP Research Group (1986) Parallel distributed processing, explorations in the microstructure of cognition. Volume 1: Foundations and Volume 2: Psychological and Biological Models, a Bradford book. MIT Press, Cambridge

[SWH98] Schmolesky M, Wang Y, Hanes D, Thompson K, Leutgeb S, Schall J, Leventhal AG (1998) Signal timing across the Macaque visual system. J Neurophysiopl 79(6):3272–3278

[SCS97] Shapiro KL, Caldwell JI, Sorensen RE (1997) Personal names and the attentional blink: the cocktail party revisited. J Exp Psychol Hum Percept Perform 23:504–514

[Sha99] Shapiro KL, Luck SJ (1999) The attentional blink: a front-end mechanism for fleeting memories. In: Fleeting memories, cognition of brief visual stimuli, A Bradford book. MIT Press, Boston, pp 95–118

[Sno88] Snodgrass JG, Corwin J (1988) Pragmatics of measuring recognition memory: applications to dementia and amnesia. J Exp Psychol Gen 117(1):34–50

[SSH75] Squires NK, Squires KC, Hillyard SA (1975) Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man. Electroencephalogr Clin Neurophysiol 38(4):387–401

[SBB07] Su L, Bowman H, Barnard PJ (2007) Attentional capture by meaning: a multi-level modelling study. In: Proceedings of the 29th Annual Cognitive Science Society. Cognitive Science Society, Austin, pp 1521–1526

[SBB08] Su L, Bowman H, Barnard PJ (2008) Performance of reactive interfaces in stimulus rich environments, applying formal methods and cognitive frameworks. In: The 2nd International Workshop on Formal Methods for Interactive Systems FMIS2007 (held in conjunction with HCI2007), Electronic Notes in Theoretical Computer Science, vol 208. Elsevier, Amsterdam, pp 95–111

[Tea93] Teasdale JD, Barnard PJ (1993) Affect, cognition and change: re-modelling depressive thought. Lawrence Erlbaum Associates, Hove

[Vid73] Vidal JJ (1973) Toward direct brain-computer communication. Ann Rev Biophys Bioeng 2:157–180

[VLS98] Vogel EK, Luck SJ, Shapiro KL (1998) Electrophysiological evidence for a postperceptual locus of suppression during the attentional blink. J Exp Psychol Hum Percept Perform 24(6):1656–1674

[WMC02] Wang M, Madhyastha T, Chan NH, Papadimitriou S, Faloutsos C (2002) Data mining meets performance evaluation: fast algorithms for modeling bursty traffic. In: 18th International Conference on Data Engineering

[Wyb05] Wyble B, Bowman H (2005) Computational and experimental evaluation of the attentional blink: testing the simultaneous type serial token model. In: Bara BG, Barsalou LW, Bucciarelli M (eds) CogSci 2005, XXVII Annual Conference of the Cognitive Science Society. Cognitive Science Society, Cognitive Science Society through Lawrence Erlbaum, Austin, pp 2371–2376

[WCB06] Wyble B, Craston P, Bowman H (2006) Electrophysiological feedback in adaptive human–computer interfaces, Technical Report 8-06, Computing Laboratory, University of Kent, Canterbury, UK

Received 1 March 2008

Accepted in revised form 6 October 2008 by A. Cerone, P. Curzon and D.A. Duce

Published online 19 November 2008