An Algorithm for Simulating Human Selective Attention

Giovanna Broccia[1], Paolo Milazzo[1], and Peter Csaba Ölveczky[2]

[1] Department of Computer Science, University of Pisa, Italy
{giovanna.broccia,milazzo}@di.unipi.it
[2] Department of Informatics, University of Oslo, Norway
peterol@ifi.uio.no

Abstract. Human cognitive processes have been shown to be very important in human-computer interaction (HCI). In particular, the brain mechanism of *selective attention* plays a key role in determining the success of an interaction with a device. During such interaction, the attention of the user is directed at the specific task to be performed. If the user has to interact simultaneously with more than one device, attention is directed at one of them at a time. Attention can therefore be seen as a shared resource and the attentional mechanisms play the role of a task scheduler.

In this paper we propose an algorithm for simulating the human selective attention. Simulations can then be used to study situations in which a user has to interact simultaneously with more than one device. This kind of study is particularly important in safety-critical contexts in which failures in the main task, such as driving a car or setting an infusion pump, may have serious consequences.

Keywords: Simulation Algorithm, Human-Computer Interaction, Selective Attention, Cognitive Load

1 Introduction

A key goal of interface design is make it easy for the user to perform the necessary tasks at hand by interacting with a device. This may require understanding how the user perceives and interprets the state of the device, recognizes the enabled actions, memorizes information and takes decisions based on such information. These cognitive processes can be modeled [3], and techniques such as simulation and model checking can be applied to analyse and predict user's behaviours.

Reasoning on user's behaviour is nontrivial already with one task, such as withdrawing money from an ATM, but is certainly a complex problem when the user has to perform multiple tasks concurrently. In particular, it might be very hard to reason analytically about how the user distributes attention to the different tasks. Analysing attention is particularly important in order to predict the behaviour of users involved in concurrent tasks exactly like schedulers are important to predict the behaviour of concurrent processes.

Working memory is among the cognitive resources that are mostly involved in interactions with computers and other technological devices. Several models of the working memory have been proposed in the psychological literature, based on different hypotheses about the structure and functioning of such a cognitive system [11, 15, 13, 14, 12]. All of these models agree on the central role of (selective) attention in the regulation of the working memory activity.

According to some psychological studies [21], the cognitive load of the tasks (i.e., the amount of cognitive resources each of them requires) influences the activity of the attentional mechanisms. In particular, focusing attention to a "main" task may be impeded by a secondary "distractor" task having a high cognitive load.

Another factor influencing attention is the fact that some tasks (e.g. driving a car or setting an infusion pump) might be more critical than others (e.g. setting the address in a satellite navigator or resizing the window of the virtual clinical folder application). A critical task should attract the attention of the user more than a non-critical task. However, if the user is involved in different concurrent tasks, one of which is safety-critical and the others non-critical but characterized by a high cognitive load, such a cognitive load of the non-critical tasks could causes the attention of the user to be moved away from the safety-critical task.

We therefore propose an algorithm that allows us to simulate the human selective attention in the case of users involved in multiple tasks, some of which may be safety-critical. Simulations would allow us to get a quick feedback about whether a human can safely perform multiple such tasks, or about which changes should be made to the interface of a device to make interacting with it not too distracting from another (possibily critical) task.

We show that the proposed algorithm simulates human selective attention in accordance with the description of its functioning in the psychological literature.

2 Cognitive Load and Influence on Selective Attention

The *cognitive load (CL)* of a task is a measure of the amount of the user's cognitive resources required for the task completion. The main cognitive resource used during the execution of a task is the *working memory (WM)* [4, 1]. The WM is a cognitive system with a limited capacity, and is responsible for the transient holding, processing, and manipulation of information. It is involved in the accomplishment of cognitive activities such as reasoning, decision making, learning and problem solving [5].

Over the years different models have been proposed to explain how the WM works [11, 15, 13, 14, 12]. Although all of these models are based on different hypotheses, they all agree on two important aspects of the WM: it can store a limited amount of items (that hence dacay over time) and it is responsible of both processing and storage activities. The limited capacity of the WM is thought to be the cause of the phenomenon known as the processing-storage trade-off: under heavy memory load, resources that are devoted to storage are no longer available for processing, and performance deteriorates.

There are several hypothesis regarding the nature of the items decay. One is that memory traces in WM decay within a few seconds, unless refreshed through rehearsal, and because the speed of rehearsal is limited, we can maintain only a limited amount of information. This is the "Task Switching Model" by Towse and Hitch [16]. Their memory decay hypothesis is that the strength of the memory traces in the short-term storage space weakens as the temporal interval between storage and recall increases.

The theory most successful in explaining experimental data on the interaction of maintenance and processing in WM is the "Time-Based Resource Sharing Model" [17]. This theory assumes that representations in WM decay unless they are refreshed and that refreshing them requires an attentional mechanism.

The attentional mechanism is also needed for any processing task executed concurrently to memory refreshing, especially when the processing components require retrieval from long-term memory. Both processing and maintenance of information in WM therefore share the same resource: the attention. When there are small time intervals in which the processing task does not require attention, this time can be used to refresh memory traces. When attention is switched away from the items to be recalled, their activation suffers from a time-related decay. This effect would be particularly pronounced when the processing component involves memory retrieval from long-term memory.

The amount of forgetting therefore depends on the temporal density of attentional demands of the processing task. Such a temporal density corresponds to what in [17] is defined as CL. The CL is a function of the time during which a task captures attention impeding the refreshing of decaying memory traces. As this time increases, the fewer and the shorter are the pauses during which the attention can be addressed to the refreshing of the decaying items.

When different retrievals (that could be of different types) are performed at a constant pace, the CL would correspond to the following value:

$$CL = \sum a_i n_i / T \tag{1}$$

where n_i corresponds to the number of retrievals of type i, a_i to a parameter that represents the difficulty of such retrievals (i.e., the time during which these retrievals totally capture attention), and T is the total duration of the activity.

Figure 1 shows a schematic representation of a portion of a WM span task in which the goal of the interaction is to remember a sequence of letters (in white boxes) while performing some processing activities (e.g. reading aloud some digits or solving some operations) that require successive retrievals (gray boxes marked R). The three panels in the figure illustrate variants of the same task that differ in the amount of cognitive load.

Several studies show how the attentional limitations could cause trouble when performing concurrent tasks [18–21]. In particular, [21] describes the roles of WM, of the CL, and of the attentional mechanism, in the interaction with two concurrent tasks (a "main" task and a "distractor" task). It is shown that when the CL of the "distractor" task increases, the interaction with the "main" task could be impeded.

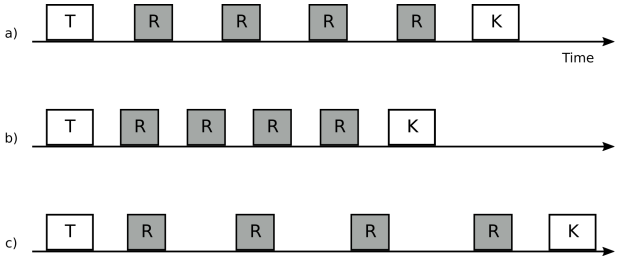

Fig. 1. Working memory tasks with different values of CL: b) has the highest, c) has the lowest and a) has a value in the middle of the other two.

3 The Simulation Algorithm

A task to be completed can be seen as a sequence of *basic tasks*: single actions that can not be further decomposed. For example, in the WM task described in the previous section, the basic tasks are the single actions such as read and "put in mind" a letter, solve an operation, read aloud a digit, and so on. In the task of sending an email, the basic tasks would be: typing a character, looking for a button in the interface, clicking on the button, etc.

The time between two basic tasks could correspond to the time necessary to switch from one basic task to the next, but also to the time required by the used device to process the received input and to enable the execution of the next basic task.

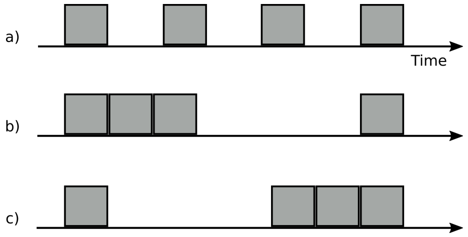

Fig. 2. Three different tasks with basic tasks denoted as gray boxes.

In [17], the time between two basic tasks is not explicitly taken into account since the definition of the CL of a task assumes that the single actions are performed at a constant pace (see Eq.1). According to such a definition, the CL of the three tasks depicted in Figure 2 would be the same. However, if the three tasks in Figure 2 were potential "distractors" of another "main" task, they would interfere with the main task differently over time: the first one would constantly attract the attention of the user, the second one would attract the attention mostly at the very beginning, and the third one mostly after some time.

We propose an algorithm for simulating the selective attention in which the cognitive load of the concurrent tasks is recomputed each time a basic task is completed. Actually, in order to choose which of the tasks to execute, the algorithm does not compute the cognitive load of each task, but the difficulty and duration of the next basic task of each task. In a sense this corresponds to computing an instantaneous CL rather than the average CL of the whole task, and it allows tasks like the ones in Figure 2 to be differentiated.

The free time between two tasks is modeled as a waiting time before the execution of a basic task (i.e., information is included in the basic task that will follow such a free time). Moreover, each tasks is associated with a parameter describing its *criticality level*.

Definition 1. *A* basic task *is defined as a triple* $\langle w, t, d \rangle \in \mathbb{R}^3$ *where:*

- w *is the* waiting time *before the basic task is enabled;*
- t *is the time corresponding to the basic task* duration*; and*
- d *is the* difficulty *of the basic task, with* $0 < d \leq 1$.

Definition 2. *A* task *is a sequence of basic tasks, denoted* $t_1.t_2.\ \ldots\ .t_n$*, associated with a criticality level* $C \in \mathbb{R}$ *such that* $0 < C \leq 1$.

A task is denoted as a pair enclosed between square brackets $[t_1.t_2.....t_n, C]$. Moreover, we denote an empty sequence of basic tasks as ε. Consequently, the pair $[\varepsilon, C]$ will represents a *completed task*.

The state of the simulation is given by a *configuration* $\mathcal{C}$, essentialy the set of active tasks, and by a global clock gc that will be used to increase the probability of choosing a task that has been ignored for a long time. For this reason, also the timestamps of the last executions of all the tasks are stored in the configuration.

Definition 3. *A* configuration $\mathcal{C}$ *is a set of triples* (tid, T, ts) *where:*

- tid *is a* task identifier *(of any type and not repeated in the configuration);*
- T *is a* task $[t_1.t_2.....t_n, C]$*;*
- ts *is a* timestamp *storing the last time the task with identifier* tid *has been executed.*

We define a few auxiliary functions that are used by the simulation algorithm.

Given a task T, functions $hd(T)$ and $tl(T)$ give its first basic task and the sequence of the other basic tasks, respectively. Formally:

$$hd([t_1.t_2.....t_n, c]) = t_1 \qquad tl([t_1.t_2.....t_n, c]) = t_2.....t_n$$

Moreover, $criticality(T)$ gives the criticality level of T.

Given a configuration $\mathcal{C}$, $enabled(\mathcal{C})$ gives the set of the tasks that are enabled at time gc (the global clock). A task is enabled if the waiting time of its first basic task has passed since the execution of the previous basic task:

$$enabled(\mathcal{C}) = \{(tid, T, ts) \in \mathcal{C} \mid \langle w, t, d \rangle = hd(T) \wedge gc - ts \geq w\}$$

Algorithm 1 Algorithm for simulating selective attention

```
 1: while not $completed(\mathcal{C})$ do
 2:   if $enabled(\mathcal{C}) \neq \emptyset$ then
 3:     for all $(tid, T, ts) \in enabled(\mathcal{C})$ do
 4:       $\alpha_{tid} := c \cdot t \cdot d \cdot (1 + (gc - ts))$
 5:       where $\langle w, t, d\rangle = hd(T)$ and $c = criticality(T)$
 6:     end for
 7:     choose $(\overline{tid}, \overline{T}, \overline{ts}) \in enabled(\mathcal{C})$ with probability $\frac{\alpha_{\overline{tid}}}{\sum_{(tid,T,ts)\in enabled(\mathcal{C})} \alpha_{tid}}$
 8:     $gc := gc + \bar{t}$ where $\langle w, t, d\rangle = hd(T)$
 9:     $\mathcal{C} := (\mathcal{C} \setminus (\overline{tid}, \overline{T}, \overline{ts})) \cup (\overline{tid}, tl(\overline{T}), gc)$
10:   else
11:     $gc := min\{ts + w \mid (tid, T, ts) \in \mathcal{C} \wedge \langle w, t, d\rangle = hd(T)\}$
12:   end if
13: end while
```

Furthermore, $completed(\mathcal{C})$ is true if and only if all tasks in C are completed:

$$completed(\mathcal{C}) = \forall (tid, T, ts) \in \mathcal{C}.T = [\varepsilon, c]$$

Now, we define the simulation algorithm of selective attention.

The algorithm performs a main loop that essentially executes one basic task at each iteration. The basic task to be executed is the first basic task of one of the enabled tasks. For each of such candidate basic tasks, an *attention attraction factor* α_{tid} is computed as the product of the criticality level, duration, difficulty and time from the last execution. Each of the candidate basic tasks then has a probability of being chosen that is proportional to α_{tid}. Once a basic task has been chosen, it is removed from the configuration and the global clock gc is updated. If the algorithm reaches a configuration in which no task is enabled, the main loop performs an iteration in which only the global clock gc is updated.

In order to show that the proposed simulation algorithm simulates selective attention in accordance with relevant literature, let us consider a variant of the algorithm that does not take the task timestamp into account when computing the attention attraction factor α_{tid}. This corresponds to modifying line 4 of the algorithm into $\alpha_{tid} := c \cdot t \cdot d$.

Let us consider two concurrent tasks with the same criticality level and each consisting of k identical basic tasks:

$$T_1 = \langle w_1, t_1, d_1\rangle.\langle w_1, t_1, d_1\rangle.\ldots.\langle w_1, t_1, d_1\rangle$$
$$T_2 = \langle w_2, t_2, d_2\rangle.\langle w_2, t_2, d_2\rangle.\ldots.\langle w_2, t_2, d_2\rangle$$

In order to complete both tasks, the simulation algorithm performs exactly $2k$ steps (where a step represents the execution of a single basic task). Since the two tasks have the same criticality level, the probability of completing task T_1 at step n, with $k \leq n \leq 2k$, is

$$P(T_1, n) = \left(\frac{t_1 d_1}{t_1 d_1 + t_2 d_2}\right)^k \left(\frac{t_2 d_2}{t_1 d_1 + t_2 d_2}\right)^{(n-k)} \binom{n}{n-k}$$

Now, we can compute the expected number of steps necessary to complete task T_1, namely $E[T_1] = \sum_{n=k}^{2k} P(T_1, n)n$, that corresponds to

$$E[T_1] = \left(\frac{t_1 d_1}{t_1 d_1 + t_2 d_2}\right)^k \sum_{n=k}^{2k} \left(\frac{t_2 d_2}{t_1 d_1 + t_2 d_2}\right)^{(n-k)} \binom{n}{n-k} n$$

The formula shows that the expected number of steps for the completion of T_1 increases with the difficulty and duration of the basic tasks of T_2, namely it increases when the CL of T_2 increases. Hence, the algorithm simulates the switching of attention in agreement with what described in [17] and [21].

Let us now discuss what changes when the task timestamp is taken into account, namely when line 4 is exactly as in the algorithm definition. Formula $E[T_1]$ becomes more complex since the repeated probabilistic events are no longer independent. However, the structure of the formula remains the same, with a result that is still increasing with the difficulty and duration of the basic tasks of T_2 (in agreement with [17] and [21]). In addition to this, the timestamps tend to favor at each step the task that has not been choosen in the previous step. As a consequence, the regular alternation of T_1 and T_2 is promoted with, as a result, a reduced variance in the distribution of the number of steps necessary to complete T_1. Hence, the use of timestamps reduces the probability of innatural starvation phenomena among the tasks.

4 Conclusion

We proposed a (probabilistic) algorithm for simulating the human selective attention, based on the current knowledge of this cognitive mechanism. The algorithm takes into account the cognitive load and the criticality level of the tasks to be executed. The algorithm could be used to simulate the interaction of a user with more than one device. Simulating this kind of situations is particularly interesting when one of the devices is associated to a safety-critical task such as driving a car or an infusion pump. By mean of simulations it could be possible to identify situations in which the non-critical devices could represent a too high distraction for the user and could lead to failures in the safety-critical task.

We plan to implement the algorithm and to evaluate it empirically on a number of examples of concurrent tasks. This will probably lead us to refine the algorithm by introducing weights for the considered parameters. The implementation will be made available here [22]. We plan also to validate the algorithm against data collected by running some experiments with real users involved in more than one (web-based) task.

Acknowledgements. This work has been supported by the project "Metodologie informatiche avanzate per l'analisi di dati biomedici (Advanced computational methodologies for the analysis of biomedical data)" funded by the University of Pisa (PRA_2017_44).

References

1. Atkinson, R. C., Shiffrin, R. M.: Human memory: A proposed system and its control processes. Psychology of learning and motivation, Elsevier, 2, 89–195 (1968)
2. Atkinson, R. L.: Hilgard's introduction to psychology (Vol. 12). Philadelphia PA: Harcourt Brace College Publishers, (1996)
3. Cerone, A.: A cognitive framework based on rewriting logic for the analysis of interactive systems. International Conference on Software Engineering and Formal Methods. Springer International Publishing, (2016).
4. Miller, G. A., Galanter, .E, Pribram, K. H.: Plans and the Structure of Behavior. Holt, Rinehart and Winston. 65 (1960)
5. Diamond, A.: Executive functions. Annual review of psychology, 64, 135–168 (2013)
6. Daneman, M., Carpenter, P. A.: Individual differences in working memory and reading. Journal of verbal learning and verbal behavior, Elsevier, 19, 4, 450–466, (1980)
7. Miyake, A., Just, M. A., Carpenter, P. A.: Working memory constraints on the resolution of lexical ambiguity: Maintaining multiple interpretations in neutral contexts. Journal of Memory and Language, Academic Press, 33, 2, 175, (1994).
8. Barrouillet, P.: Transitive inferences from set-inclusion relations and working memory. J. of Experimental Psychology: Learning, Memory, and Cognition, American Psychological Association, 22, 6, 1408, (1996)
9. Kyllonen, P. C., Christal, R. E.: Reasoning ability is (little more than) working-memory capacity?! Intelligence, Elsevier, 14, 4, 389–433, (1990)
10. Shute, V. J.: Who is likely to acquire programming skills?. Journal of Educational Computing Research, SAGE Publications, 7, 1, 1–24, (1991)
11. Baddeley, A. D., Hitch, G.: Working memory. Psychology of Learning and Motivation, Elsevier, 8, 47–89, (1974)
12. Baddeley, A.: The episodic buffer: a new component of working memory?. Trends in cognitive sciences 4.11, 417–423 (2000)
13. Ericsson, K. A., Kintsch, W.: Long-term working memory. Psychological Review. 102 (2): 211–245.(1995)
14. Cowan, N.: Attention and memory: an integrated framework. Oxford University Press (1995)
15. Just, M. A., Carpenter P. A.: A capacity theory of comprehension: individual differences in working memory. Psychological Review 99.1, 122 (1992)
16. Towse, J. N., Hitch, G. J., Hutton, U.: On the interpretation of working memory span in adults. Memory & Cognition, Springer, 28, 3, 341–348, (2000)
17. Barrouillet, P., .Bernardin, S., Camos, V.: Time constraints and resource sharing in adults' working memory spans. J. of Experimental Psychology: General 133.1, 83, (2004)
18. Pashler, H.: Dual-task interference in simple tasks: data and theory. Psychological Bulletin, American Psychological Association, 116, 2, 220, (1994)
19. Lavie, N., Hirst, A., De Fockert, J. W., Viding, E.: Load theory of selective attention and cognitive control. Journal of Experimental Psychology: General, American Psychological Association, 133, 3, 339, (2004)
20. Wickens, C. D.: Processing resources and attention. Multiple-task performance, 1991, 3–34, (1991)
21. de Fockert, J. W., Rees, G., Frith, C. D., Lavie, N.: The role of working memory in visual selective attention. Science, American Association for the Advancement of Science, 291, 5509, 1803–1806, (2001)
22. Simulator webpage http://www.di.unipi.it/msvbio/software/AttentionSim.html