Do you read me?

An Investigation Into How Expert Users Respond To Dialogue Boxes

Anthony Ioannidis
MSc Human-Computer Interaction with Ergonomics
University College London
2006-2007

Do you read me?

An Investigation Into How Expert Users Respond To Dialogue Boxes

Anthony Ioannidis
MSc Human-Computer Interaction with Ergonomics
University of London
University College London Interaction Centre

Project report submitted in part fulfilment of the requirements for the degree of Master of Science (Human-Computer Interaction with Ergonomics) in the Faculty of Life Sciences, University College London, 2006-2007.

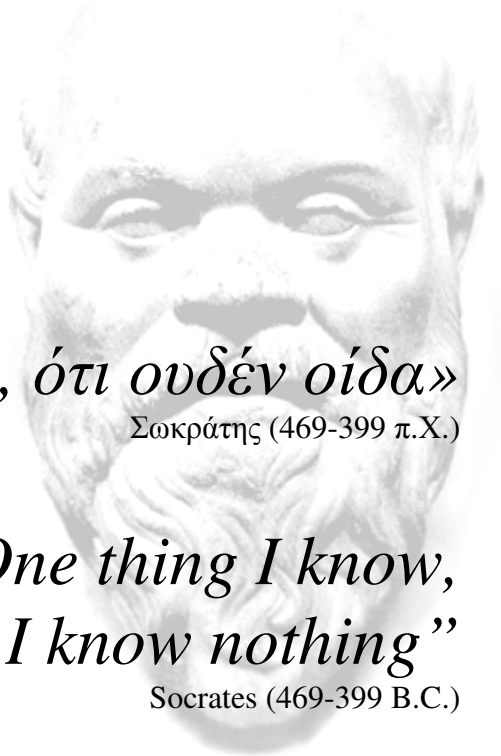
Acknowledgements

None of this would have been possible without the moral and financial support of my parents, Panos and Dora; thank you for challenging me, thank you for all the artichokes you've forced me to eat, thank you for being there for me. Another "ευχαριστώ" goes for my late grandma, who I am sure would keep supporting me in every way possible, without ever understanding a word of my thesis. A "thank you" is also due to those few teachers of mine who managed to keep my interest in sciences alive during school and University years. An even bigger thank you goes to my friends who supported me, put up with me and helped me remain sane. Kate, thanks for inspiring me and all those crazy, freezing nights out and mornings with coffee and cookies. Marina, thanks for your friendship and wisdom. Denia and George, thanks for those long free-of-stress nights under the Acropolis. Kevin, John, Niketa, Fudo, Magda, thanks for not complaining too often while I was nowhere to be seen! Roman, thank you for everything, I'm still not sure the Universe ends somewhere because, you know, it *has* to be infinite.
Special thanks are due to Dr Paul Cairns for kindly enriching my statistics knowledge and, of course, my supervisor, Professor Ann Blandford, for her valuable help, for making sure that I'm always on track and for her witty comments about Greeks; I hope that sacrificing my attendance of a couple of Greek weddings paid off! Finally, to all of those who are not here but believe they should be: Thank you! You know who you are.

ευχαριστώ

"Islands in the stream, that is what we are..."
Rogers, K. & Parton, D. (1983).

As these lines were being written, over 60 people's lives in my country, Greece, were taken away by the biggest forest fires ever seen worldwide during the last century, while several more are still missing. I can't help but think that, however hard we try, it seems that Man will never change.
In memoriam...



«Εν οίδα, ότι ουδέν οίδα»

Σωκράτης (469-399 π.Χ.)

“One thing I know,
that I know nothing”

Socrates (469-399 B.C.)

Abstract

This study is concerned with dialogue boxes and the way experienced computer users respond to them. We are looking at whether users actually read messages communicated via dialogue boxes or whether they respond in an automatic way and what can be done in order to reduce any such automatic responses.

For this we ran two experiments, recruiting a total number of 40 participants, testing the hypotheses that i) Users tend to *not* read the message of an *expected* dialogue box, and that ii) Altering the position of an *expected* dialogue box on the screen will affect an experienced user's automatic action when answering it. Participants were asked to answer simple mathematical problems while being interrupted by dialogue boxes asking them if they wanted to proceed to the next problem, requiring a simple Yes / No response. For some of the dialogue boxes, however, the question was whether they wanted their score to be reset. In the first experiment the position of all dialogue boxes was constant, while in the second the position of the "reset" dialogue boxes was slightly altered. The analysis of the results suggests that both hypotheses hold true, with most users missing the "reset" question in the first experiment and less missing it in the second experiment.

These results suggest that dialogue boxes are *not* a satisfactory safety mechanism against wrong commands as it is widely thought, but that there is room for improvement. Based on these findings, we give several suggestions with regards to designing and testing interactive computer systems and propose ideas for further research.

Table of Contents

List of Figures

List of Images

List of Tables

Introduction

Dialogue boxes and the issue of automaticity

This study is concerned with dialogue boxes (DBs) and the way computer users respond to them. Specifically, we will be looking at the issue of automaticity and whether experienced users actually read messages communicated via DBs and if we can indeed rely on assuming this.

There are several kinds of DBs and we can categorise them in many different ways, depending for example on whether they merely provide information or whether they (also) seek input from the user, on the number of available buttons/options etc. Since we are interested in whether users actually read a DB message before they take any action –whether this action is to reply to a question by pressing the appropriate button or to just dismiss the DB– we are essentially studying all types of DBs, as there is no purpose in displaying a DB without a message at all, unless of course there is a software fault.

We mentioned earlier the issue of automaticity and that we are interested in the way *experienced* users react to DBs. Automaticity, or automatic processing, is the term describing "*the skilled action that people develop through repeatedly practising the same activity, for example driving a car*" (Toft and Mascie-Taylor, 2005). When tasks become automatic, they are faster, require little or no conscious intervention from a person and do not reduce a person's capacity for performing other tasks (i.e. they demand zero attention) (Eysenck and Keane, 2000). It is therefore of no surprise that automaticity is usually described in terms of the benefits it brings to people, with regards to the improvement in their skills and productivity. However, as we will see later on, there is evidence to suggest that, in certain cases, automaticity can also have negative effects (Toft and Mascie-Taylor, 2005, Deutsch, 2005, Resnick, 2001, Barshi and Healy, 1993).

It is evident that automaticity develops as the user becomes familiar with a system and this is why we are interested in experienced users. Langer (1989) notes that one of the dangers of automaticity –or "mindlessness", in her own words– is that "*we take in and use limited signals from the world around us […] without letting other signals penetrate as well*". Similarly, Barshi and Healy (1993) argue that automaticity has "a cost that manifests itself in procedures that are highly routinised but require close attention […] where errors occur because the routine leads to automaticity".

These observations lead us to the next question for this chapter, which is why is studying the effects of automaticity in responding to DBs important.

Study rationale

> *"This is the authors' second attempt at writing this introduction. Our first attempt fell victim to a design quirk coupled with an innocent, though weary and less than attentive, user. The word-processing package we originally used to write this introduction is menu based. Menu items are grouped to reflect their function. The 'save' and 'delete' options, both of which are correctly classified as file-level operations, are consequently adjacent items in the menu. With a cursor controlled by a trackball it is all too easy for the hand to slip, inadvertently selecting delete instead of save. Of course, the delete option, being well thought out, pops up a confirmation box allowing the user to cancel a mistaken command. Unfortunately, the save option produces a very similar confirmation box – it was only as we hit the 'confirm' button that we noticed the word 'delete' at the top... Unfortunately, this is an all too common occurrence."*
>
> (Dix, Finlay, Abowd and Beale, 1998, p. 1)

The above quotation shows the unfortunate side-effects of automaticity in an "everyday" task. We all make mistakes and this is why most computer programmes have security measures in place, the most simple of all probably being a DB asking us to confirm our commands. But just how effective is this measure? Can we safely rely on DBs effectively communicating a question or an event to the user? It is not only a matter of using clear and simple language, but also a matter of whether users actually read the DBs before responding to them. Why, in the example above, did the user first hit the 'confirm' button and then noticed the word 'delete', when it was too late?

It has been argued that even very experienced users rely on the interface display when performing routine tasks (Payne, 1991). While this is understandable and expected to a certain degree since it helps reduce the user's cognitive load, Payne also suggests that "*users have common expectations […] and when (their software) departs from these predictions, they fail to acquire a different model. […] The good performance is an accident; the behaviour of the system happens to conform to the most common guesses*". The fact that users often predict the results of their operations based on their own "common sense" (mental models, expectations etc.) and the fact that they may fail to accordingly adapt their mental models if the system behaves in a way different than expected, can lead to potentially serious issues. In critical situations, systems usually provide feedback in order to enable their users to understand the effects of their actions. If users do not expect an error message but rather a message of success and discard any other message as such, then it is apparent that this particular method of feedback or safety-verification does not work as intended.

Although everyday situations that go wrong, as the one quoted earlier, might not seem disastrous, their effects are probably magnified for the individual user who might have just lost hours of work and sleep. If we take a step further and consider safety-critical systems (as we will do later on), the results of a missed piece of information could, indeed, be immense.

One of the problems with automatic processes is their lack of flexibility, which is likely to disrupt performance when the prevailing circumstances change (Eysenck and Keane, 2000). Apart from the apparent consequences this has in practice, there are also serious issues with testing a system in order to ensure that it will provide the best support possible should exceptional circumstances exist. This is obviously important in safety-critical systems (such as in a nuclear plant or an aircraft control system) but also in systems that are less critical but still prone to encounter many instances of unusual circumstances.

However, studies have shown that tests under controlled conditions in laboratories do not always reflect the way users make decisions in the complex environments and situations often faced in practice (Resnick, 2001). It is also accepted that expert reviews do not identify several types of problems associated with the way users behave in practice (Desurvire, 1994) and as Karat (1994) suggests, testing should be conducted using scenarios with realistic tasks, motivation and *experience*. This includes cases where the user is tired, bored or just lazy but, in addition, if expert users generally interact differently with a system than novice users and especially under stressful situations (Resnick, 2001) then this needs to be taken into account when designing the tests for that system. The knowledge of how and why users react in different ways under different circumstances can help us design not only better systems but also better tests to ensure that these systems support their users in the way they are supposed to.

Obviously, the questions of whether an interface is successful in effectively communicating with the system's users and whether error prevention measures are actually effective are in the very centre of any system's usability evaluation.

Our study in a broader context

There has been extensive research on how automaticity develops and what the underlying cognitive mechanisms are. There is also some research on how it affects human performance during critical tasks, but unfortunately in most cases this only came after accidents had already occurred (Deutsch, 2005; Toft and Mascie-Taylor, 2005). Before we review the relevant literature in the next chapter, we believe it is important to first place this study within it.

So far, most studies on automaticity concentrated on its positive effects on human behaviour (improvement in response time, fewer cognitive requirements etc). Automaticity itself cannot be characterised as good or bad; it is merely a result of training under repeated circumstances. It is the combination of automaticity and exceptional situations that can lead to mistakes. Good interface designs should improve safety and performance by allowing work to progress more fluently, with less interference or interruption from the tools needed to achieve task goals (Harrison and Kurtenbach, 1995).

As early as 1991, Payne had suggested that "*we need to shift attention away from static aspects of the interface such as command language design or screen layout and to start work on the dynamic properties of interactive systems – the properties that make them interactive*" (Payne, 1991). This study is along these lines of investigating the dynamic nature of dialogue boxes; they can continually appear after certain known operations, displaying set and known questions, the answers to which are usually the same (as for example in "are you sure you want to delete this file?"), but they can also appear unexpectedly, either when the user does not expect a DB at all or when he expects a different one. It is obvious that automaticity can be a problem in the latter situation.

It is not our intention to study how or why automaticity occurs, although we will use the theoretical background to understand the way it possibly affects computer users. What we will endeavour to assess is its effect on one of the most simple and basic operations in computer systems: that of responding to dialogue boxes. Such an operation, simple as reading a short text and clicking on a button, is probably considered trivial by most software designers, with the danger of an underlying assumption that simple actions should produce correct results and can not be the source of mistakes. It is our intention to test if this assumption is safe to make and if there are ways to make it even safer, and initiate a discussion on whether we can rely on it for various purposes, including –but not limited to– safeguarding against wrong and potentially destructive user commands.

Literature Review

The theory

As we mentioned earlier, there has been extensive research on the issue of automaticity and its underlying cognitive mechanisms. It is our intention in this chapter to present a brief account of this research on automaticity and closely related issues. This will help us better understand and interpret the results of our experiments.

Automaticity, or automatic processing, is the term describing "*the skilled action that people develop through repeatedly practising the same activity*" (Toft and Mascie-Taylor, 2005). There is reasonable agreement that automatic processes:

- are fast
- do not reduce the capacity for performing other tasks (i.e. they demand zero attention)
- are unavailable to consciousness
- are unavoidable

(Eysenck and Keane (2000)). Few processes are fully automatic, in the sense of conforming to all the above criteria, with most being partially automatic (Eysenck and Keane, 2000), and in some cases the observed effects could also be attributed to other reasons (e.g. lack of interest in the task or laziness). The first step to automaticity is the repetition priming effect; this is the observation that stimulus processing is faster and easier on the second and successive presentations of a specific stimulus, and Logan (1990) suggests that there seems to be a common underlying mechanism between repetition priming and automaticity. It is an interesting fact that even amnesic patients exhibit a variety of repetition priming effects (Eysenck and Keane, 2000): their performance is greatly improved by the prior presentation of stimuli, even when there is an absence of conscious awareness that these stimuli have been previously presented, suggesting a subconscious behaviour which is also observed during automatic processing.

Related to the issue of automaticity are the issues of attention, divided attention and action slips. According to Eysenck and Keane (2000), attention generally refers to selectivity of processing. Access to consciousness is controlled by attentional mechanisms and can be either active, based on top-down processes (e.g. specific task goals), or passive, based on bottom-up processes (e.g. as a response to an alarm going off or the appearance of a new dialogue box on a computer screen). Payne (1991) argues that some crucial phenomena can undermine the simplest top-down model of planning and that this suggests that human action is, to an important degree, display-based, responsive to (and even reliant upon) patterns in the external environment.

Attention can either be focused on only one input modality or divided between several input modalities. Performance in this case depends on task similarity, practice and task difficulty. Where full automaticity has been developed for a specific task, performing that task should not impede a person's performance on other tasks (Eysenck and Keane, 2000). Automaticity is a mechanism our cognitive system develops for coping with the demands of divided attention, especially under extreme circumstances.

The term action slips refers to the performance of actions that were not intended (Eysenck and Keane, 2000). Hay and Jacoby (1996) argue that action slips are likely to occur when the following two conditions are satisfied:

1) The correct response is *not* the strongest or most habitual one
2) Attention is not fully applied to the task of selecting the correct response

Eysenck and Keane (2000) also note that action slips result from a failure to shift from automatic to attention-based control at the right time, with the findings of Robertson, Manly, Andrade, Baddeley and Yiend (1997) also suggesting that sustained attention is needed in order to avoid action slips.

There are several theories that attempt to model automaticity and the way it works and it would be far from our intentions to list all of them here. There is, however, the common issue of the encoding of the stimuli that trigger automaticity. Shiffrin and Schneider (1977) suggest that automatic processes develop through practice and indeed during their experiments they observed that after 2100 initial trials with one consistent letter-to-letter mapping, it took participants nearly 1000 trials under reverse mapping conditions before their performance reverted to its level at the start of the experiment (results as reported in Eysenck and Keane, 2000). As Logan, Taylor and Etherton (1996) note, most theories do not indicate clearly how automaticity develops through practice. Their own findings support that this development depends on both encoding and retrieval of past stimuli and related experiences and that if some aspect of a stimulus is important in automatic actions then this suggests that it was encoded. Logan (1988) had earlier described automaticity as being unavailable to conscious awareness and as being constructed…

> "…*as the acquisition of a domain specific knowledge base, formed of separate representations, instances, of each exposure to the task. Processing is considered automatic if it relies on retrieval of stored instances, which will occur only after practice in a consistent environment. Practice is important because it increases the amount retrieved and the speed of retrieval; consistency is important because it ensures that the retrieved instances will be useful.*"

Therefore, with little or even no conscious awareness of automatic tasks, it could be the case that humans can actually carry out certain tasks in a way that does not enable them to be certain, at a later stage, of whether they performed a particular action or not. This is supported by Payne's study (1991), in which 15 experienced word-processor users were asked to describe the steps of finding a specific word in a text. Only 4 out of the 15 participants recalled that they needed to click the "OK" button of a dialogue box in order to be able to proceed during the task.

Finally, Norman and Shallice (1986), also support that fully automatic processes occur with very little conscious awareness. They also introduce the idea of a higher-level supervisory attentional system that is involved in decision making, enabling humans to be flexible in their responses under novel situations.

Under all of the aforementioned theories, a specific stimulus triggers an automatic behaviour irrespective of the way this behaviour was acquired; as explained before, if the stimulus is altered (and therefore presents a novel situation to the user), then this automatic behaviour should not be triggered any more. It is our intention in our first experiment to provide participants with training that will produce automatic reactions to dialogue boxes and divide their attention between the task of answering DB questions and giving answers to mathematical questions as fast as possible. Changing the text of a dialogue box (therefore producing a different stimulus) should affect the way participants respond to the DB, if they indeed notice the change. If their response remains automatic then this is an indication that they did not notice the change and therefore did not read the text. Such a change should, of course, be in a way that does not provide other stimuli of a different nature (e.g. significantly longer text that affects the dialogue box's size etc). More on this reasoning will be presented in the next chapter.

We will also base our second experiment on these theories, to test whether changing the position of a dialogue box is an alteration strong enough for users to notice and therefore exhibit significantly different behaviour than if its position is constant.

As a final note, specific to the issue of reading text (whether it is a DB message or any generic text), research has showed that our expectations often compensate for errors during reading (Dix, Finlay, Abowd and Beale, 1998). An example is shown in Image 2.1, where number 13 can easily be mistaken as the capital letter B, on the first row. This would probably not affect our understanding of a DB's message, since we would hardly base that on a single letter. Another fact is, though, that the probability of reading every word in a serial letter-by-letter manner is extremely low (Dix, Finlay, Abowd and Beale, 1998); words can be identified as fast as single characters and "familiar" words are recognised based on their shape (Dix, Finlay, Abowd and Beale, 1998). This can easily lead to confusion if, for example, the difference between two familiar DB messages is just a word (e.g. as in "Do you want to save the file" and "Do you want to delete the file").

Image 2.1: ABC or A13C?

Environmental factors

When people behave in an automatic way, familiar cues trigger learned actions that have been successful for them in coping with similar situations in the past (Dennis and Schmidt, 2003). When examining human-computer interaction, a cue could be a familiar dialogue box, a familiar alarm sound or anything else that is connected, in their mind, with some specific event. According to Payne (1991) "*the interleaving of planning and action and display-based control apply generally to human action in all kinds of context*". Most accidents and other system failures though occur under less than ideal conditions (Resnick, 2001). While on some occasions negligence (because of fatigue or even laziness) or irrational behaviour may be the cause of a failure, it is generally accepted that system design and environmental factors interact to contribute to errors (Resnick, 2001). In the example from the previous chapter (the quotation from Dix, Finlay, Abowd and Beale (1998)), there were several factors that contributed to the user error, such as the inaccurate input method (trackball), the close position of the menu items, the similar design of the "safety" DBs, and the fatigue of the user who did not read the DB question.

Stress, fatigue and time pressure play an important role in the appearance of automaticity, even if previous training has been given on how to cope under such situations (Deutsch, 2005). Generally, "*time pressure leads to a focus on only the most salient information channels, reduces seeking behaviour and increases the use of recognition-based decision making*" (Amalberti and Deblon, 1992; Klein, 2000 as quoted in Resnick, 2001). This is because our cognitive system, upon recognising a familiar task, will automatically apply the rules of a previous similar task instead of processing the information one step at a time (Toft and Mascie-Taylor, 2005). This reduces the cognitive load by decreasing the demand made on working memory and attention capacities (Eysenck and Keane, 2000).

As Harrison, Kurtenbach and Vicente (1995) note, one of the associated psychological problems is that of focused and divided attention. When there is time pressure and multiple sources of information "*users must make choices as to what to attend to and when. At times, users need to focus their attention exclusively on a single item without interference from other items and […] trade-offs among these attentional requirements must be made*". Unfortunately in such cases, dialogue boxes tend to be intrusive by nature; their purpose is to acquire input or inform the user of an event, but in some cases this might be considered inappropriate by the user with regards to the context or timing of their appearance, and therefore the appropriate attention might not be given to their message.

The issues of time pressure and automaticity, apart from presenting an obvious problem during real and exceptional situations, also play an important part in simulated studies. If they are not replicated during system testing,

participating users may behave in categorically different ways than in actual circumstances (Resnick, 2001). If necessary, "*participants need to be trained sufficiently, in order to reproduce the effects of automaticity that will be reached by expert users under real system use*" (Resnick, 2001).

As mentioned at the beginning of this chapter, all the above observations will be helpful in understanding and interpreting the results of our experiments. But just before that, we will briefly review research similar to our study.

Other similar research

To the best of our knowledge, there is no similar research specifically on dialogue boxes. There have been some efforts on dynamically evolving interfaces based on a user's increasing expertise, but the main literature body is concerned with automaticity merely as a cognitive function.

Payne (1991) must be one of the first to suggest that "*it is time to start work on the dynamic properties of interactive systems – the properties that make them interactive*", rather than just their static features, such as menu wording, colours etc. A few years later, while testing a prototype system based on semi-transparent interface tools, Harrison, Kurtenbach and Vicente (1995) obtain some results that "*suggest new and intriguing possibilities for dynamically evolving interfaces, based on increased expertise. We wish to experimentally test this using more longitudinal studies of skilled users*". The following is, we think, an interesting conclusion in their work:

> "*We believe that interface designers can take advantage of both the intrinsic properties of the task and of an understanding of human visual attention to design new display techniques and systems. We believe that results thus far show promising advantages for creating new user interfaces and interaction techniques. We are exploiting possibilities of new technology in a way that is sensitive to both psychological and task constraints.*"

The only specific reference to dialogue boxes that we could find was a surprising, at the time, side-effect, during the testing of a web development tool for children. Gibson, Newall and Gregor (2003), developed a web authoring tool for children, by using participatory design. Two groups were used for the final evaluation: group A, with children that had previously helped in developing prototypes and during ongoing evaluation of the software; and group B, with children from different schools, who had no prior experience of using the software. Gibson, Newall and Gregor note: "*The first group were often very cautious when presented with an unexpected dialogue box, or if they accidentally moved away from the program. Surprisingly, the second group were not at all hesitant in this situation*". If we consider the difference in expertise and the fact that, for the children of the second group, DBs were unexpected (since they had no previous experience with the programme) then this "surprising" observation could easily be explained, as we will see later on.

Finally, in a broader context, there has also been some research on the effects of automaticity during the verbal challenge-response protocols used in health care and aviation (Toft and Mascie-Taylor, 2005; Deutsch, 2005). Automatic behaviour developed by professionals using verbal checklist techniques can cause them to erroneously believe that a checked procedure is safe, when it is actually not. Although not entirely similar to responding to dialogue boxes, the parallel is obvious. Indeed, the United States Federal Aviation Administration (FAA) warns that the use of verbal safety checklists may on occasions cause crew members to see what they expect to see, rather than what is actually there (Toft and Mascie-Taylor, 2005).

The first experiment

Introduction

The purpose of this experiment is to investigate any differences between the way users who are familiar with a system react to expected and unexpected dialogue boxes. Before describing the experiment we will give some definitions that will be used throughout the rest of the text. The respective abbreviations will be used interchangeably, whenever they make more sense due to space limitations.

Note that we make the assumption that users are familiar with the system they are using, otherwise the distinction between expected and unexpected dialogue boxes would make no sense.

Definitions

System: Any application that displays dialogue boxes in any context, such as (but not limited to) Word processing, Info Kiosks, air-traffic control, nuclear plant monitoring etc.

Expected Dialogue Box (ExpDB): A dialogue box that is expected by a system user to appear at a certain instant each time, as part of a familiar interaction sequence. The term "expected" here refers to the *timing* of the appearance of the dialogue box and not necessarily its message (e.g. when trying to quit Word with an unsaved document open, Word normally asks if the user wants to save the document first; the user therefore *expects* that a dialogue box will appear after clicking on "Quit" – Image 3.1)

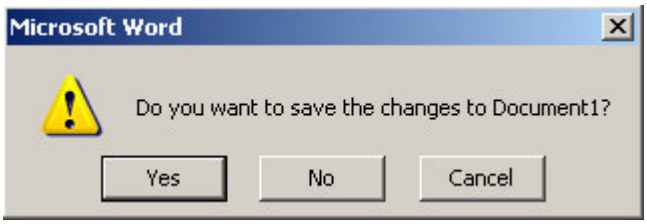


Image 3.1: An expected dialogue box

Unexpected Dialogue Box (UnexpDB): A dialogue box that is *not* expected by a system user to appear at a certain instant as part of a familiar interaction sequence. This is the exact opposite of an Expected dialogue box (e.g. a dialogue box informing the user that there is "not enough storage available" when trying to save a file – Image 3.2)

Image 3.2: An unexpected dialogue box

Expected Message in an (expected) dialogue box (ExpMSG): When an ExpDB appears, the system user has an expectation of what the message will be; i.e. that the message will be the standard message that usually appears in that ExpDB, according to the user's experience (e.g. the question "do you want to save the changes" in Image 3.1) We call such a message an ExpMSG.

Unexpected Message in an (expected) dialogue box (UnexpMSG): When an ExpDB appears but has a message different to the one that would normally be expected by the user at that instant of the interaction, then we say that the dialogue box was expected, but its message was unexpected. (e.g. when trying to connect via a modem, instead of getting an expected "Connected" message a user gets a "no dial tone" message – Image 3.3 and Image 3.4)

Note that there is no sense in defining an unexpected message in an UnexpDB, since the user does not expect anything anyway.

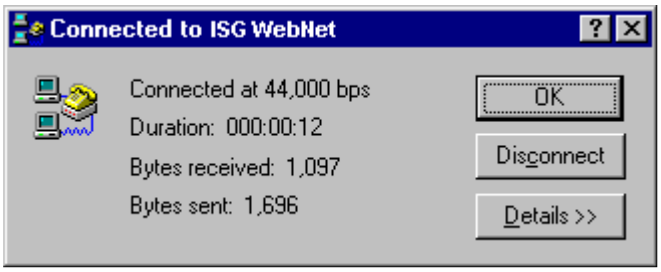


Image 3.3: An expected dialogue box with an expected message

Image 3.4: An expected dialogue box with an unexpected message

Hypothesis

Our research hypothesis is the following:

H_1: Users tend to *not* read the message of an *expected* dialogue box

As a consequence of our hypothesis, we also expect that:

Because: Users tend to not read the message of an expected dialogue box…

…therefore: their answer (choice of button to click) is to the question they expect to be asked.

Of course if the available choices (e.g. Yes/No) and layout of an ExpDB with an UnexpMSG differ significantly from those of the ExpDB with the ExpMSG, then the above hypothesis would be problematic, only because the difference in the layout and available options would probably *force* the users to read the message. A good example of such a scenario is shown in Images 3.3 and 3.4, where the layout and options are significantly different. ("OK/Disconnect/Details…" vs. "Redial=50/Cancel/More Info" and vertical vs. horizontal layout). Therefore…

…for the purposes of our working hypothesis, we make the assumption that the available options and layout of the ExpDB remain similar, regardless of whether the message is expected or unexpected.

The variables

These are the independent and dependent variables for H_1:

Independent variable: Whether the dialogue box is Expected or Unexpected, with two possible values (ExpDB / UnexpDB)

Dependent variable: Whether users read the dialogue box message before responding, with two possible values (Read / Not read)

It is obvious that there is no *direct* way of measuring the dependent variable; i.e. we cannot somehow *directly* measure whether a user actually read a message or not, but we can certainly try and infer this. Directly asking users themselves would not be suitable in this case, for at least two reasons: not only people often forget or imagine things, but also the mere fact of asking

would interfere with their behaviour. Therefore we have to find another way of indirectly measuring (inferring) the value of the dependent variable across our experiments.

To this extend, we will use two "intermediate" variables which we accept show whether the user read the text or not:

1) **MSEC:** the time in msec elapsed from the moment a dialogue box appears until a response is given (by clicking on one of its buttons). This is a ratio scale variable.
2) **ANS:** whether the answer given was correct or not, with two possible variable values (Correct / Incorrect). We assume that, out of all the possible choices (buttons) of a dialogue box, only one is correct and the rest are incorrect. We define as correct the one that will bring the user closer to achieving their goal.

Obviously, there are some underlying assumptions that need to be made explicit:

1) with regards to MSEC, we assume that:

> The higher MSEC is, the more chances are that the user actually read the dialogue box message

- We expect any irrelevant delays to be evened out as statistically insignificant.

2) with regards to ANS:

> If users actually read the message, we expect them to give the correct answer.

- We assume that users have a goal when using the system (either an ultimate goal or intermediate one).
- We also assume that, even if several answers to the dialogue box might lead closer to that goal one way or the other, only one of them leads to the *faster* way to achieve that goal and that this is obvious to the user from the dialogue box message.
- We need to make clear that we are not claiming a two-way relationship between reading the message and giving the correct answer (i.e. *not* reading the message does *not* necessarily lead to a wrong answer). A wrong answer would be given only if the question asked (UnexpMSG) was different from the one expected (ExpMSG) and the right answer was different too (e.g. the correct answer was

"yes" in one case and "no" in the other). If we "construct" the questions and answers in this way, then obviously giving an incorrect answer would mean that the user didn't actually read the message.

- Finally, we assume that all messages are clear to the users and not ambiguous in any way.

An illustration of the way that we use the MSEC and ANS variables in order to deduce our dependent variable is shown in Figure 3.1. Note that we are *not* assuming *nor* trying to prove that MSEC and ANS are a direct result of our independent variable (i.e. we are not trying to prove that if a dialogue box is unexpected then the user will always give an incorrect answer). In our case, MSEC and ANS are merely a way of deducing our dependent variable, as explained above (i.e. whether users actually read the message or not).

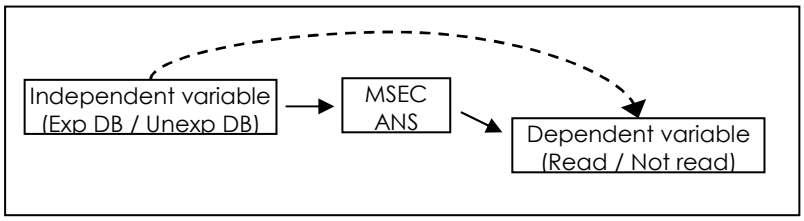


Figure 3.1: Using MSEC & ANS to deduce the dependent variable

Another point to note here is that our hypothesis refers to any type of dialogue box. Nevertheless, because we need to get a meaningful value for the ANS variable, we will only assume the cases of dialogue boxes asking a question and expecting an answer, as opposed to dialogue boxes that merely give information and have only one button (like "OK" or "Close"). The issue of generalising our results will be examined later, under the discussion section.

Designing the experiment

In order to design an experiment that will successfully test our H_1 with the help of MSEC and ANS, we need to carefully consider our dependent and independent variables. Table 3.1 summarises the way we use our variables to test our predictions.

H_1 independent Variable		Observations (Experiment dependent variables)		H_1 dependent variable
ExpDB (ExpMSG)	→	Short time in MSEC Correct ANS	→	User did not read the MSG
ExpDB (UnexpMSG)		Short time in MSEC Wrong ANS		
UnexpDB (MSG is by default Unexp)	→	Long time in MSEC Correct ANS	→	User read the MSG

Table 3.1: The variables and our predictions

Note that our hypothesis independent variable is whether the DB is Exp or Unexp. The MSG variable though needs to be changed from Exp to Unexp in order to observe changes in the ANS variable. Again, as explained above, this is because we have no other way of directly observing whether the user read the MSG or not. Notice how in Table 3.1 the main two categories on the leftmost column are ExpDB and UnexpDB and that under the ExpDB category, ExpMSG and UnexpMSG are mere subcategories. Finally, when we have an UnexpDB the MSG is by default Unexp too.

In our experiment the observations (MSEC and ANS) become the dependent variables and we have already discussed how changes in their values signify changes in the dependent variable of H1.

For our experiment we recruited 20 participants (participants' statistics will be presented in the next chapter), who were asked to use a program on their computer whenever and wherever they felt most comfortable. They were asked to make sure that they wouldn't be interrupted and they were all trained to the same standard. Training was required in order to ensure that they were familiar with the task that they were asked to do and also to shape their expectancies for the appearance of ExpDBs.

As a task we chose the easy assignment of performing simple additions and subtractions. The participants' goal was to perform correctly in as many as they could, as fast as possible, in order to gain as high a score as possible. The score was calculated as the sum of their correct mathematical answers. Table 3.2 presents the expected structure of the task and Image 3.5 shows screenshots of the corresponding sequence.

#	User action	System response
1	N/A	Present mathematical question
2	Answer question	N/A
3	Click OK	Present dialogue box asking if they want to proceed to the next question
4	Click Yes	Go to step #1

Table 3.2: The expected structure of the task

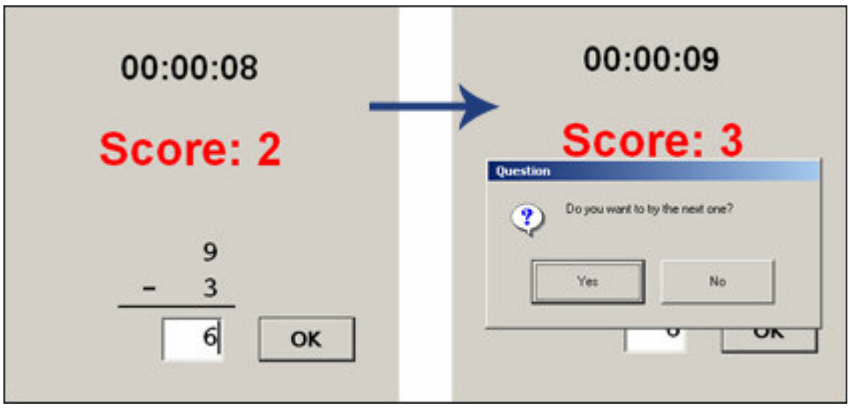


Image 3.5: Screenshots from the task sequence

As is immediately obvious, the expected dialogue box here is the one asking the user whether they "want to try the next one". The training session included 25 mathematical questions and only presented ExpDbs with ExpMSGs.

After the training session, participants were informed that the actual experiment would begin and that they should not be disturbed during that. For each participant, the system recorded the following data:

- Gender
- Year of birth
- Computer expertise (on a subjective scale of 1 - 5)
- Whether each mathematical answer was right or wrong
- The MSEC for each dialogue box
- The ANS to each dialogue box

During the actual experiment, participants were presented with 50 mathematical questions and 46 ExpDBs with ExpMSGs, exactly as in the training session. In 4 cases though (after the 15th, 25th, 30th and 45th mathematical question) they were presented with an ExpDB that had an

UnexpMSG. The UnexpMSG asked them if they want to reset their score (Image 3.6). Obviously, as correct ANS here we define "No", instead of "Yes" that participants had to click in the usual ExpDBs with the ExpMSGs. This variation enables us to measure differences in the ANS variable and infer if they read the message or not. After clicking "yes" or "no", users where presented with the usual dialogue box which even though it has a familiar message ("do you want to try the next one?"), in this case is unexpected because it was preceded by another dialogue box. If participants don't notice the UnexpMSG of the first dialogue box, then the second one will probably come as a surprise to them (remember that it is an UnexpDB), because they will probably think they already answered that question. This sequence is presented in Table 3.3.

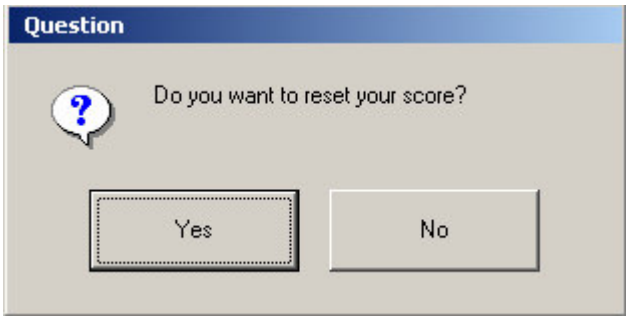


Image 3.6: An expected dialogue box with an unexpected message

#	User action	System response
1	N/A	Present mathematical question
2	Answer question	N/A
3	Click OK	Present dialogue box asking if they want to reset their score
4	Click Yes / No	Present dialogue box asking if they want to proceed to the next question
5	Click OK	Go to step #1

Table 3.3: The appearance of unexpected dialogue boxes

To conclude, there are 54 dialogue boxes in total:

- 46 ExpDBs with an ExpMSG
- 4 ExpDBs with an UnexpMSG, directly followed by…
- … 4 UnexpDBs (all succeeding the above 4 dialogue boxes)

We wanted to keep the task simple in order for the users to not deviate from their goal and also to enable us to train participants to a level where certain dialogue boxes would be expected. If the task was long and

complicated we would need so much more time in order to train users that would be inappropriate for an experiment. Apparently this resulted in a repetitive task and we can possibly expect some sort of decline of interest from the participants along the way. In order to make sure that this is not the case, although we are not interested in the correctness of the participants' mathematical results, we can use them to verify that participants were still paying attention to the task and not just making random selections.

In the next chapter we will describe the analysis of the results and make the necessary statistical tests that will ensure our findings are valid and reliable.

Analysis of the first experiment

Participants' demographics

We recruited a total of 20 participants for our experiment, 15 male and 5 female. Although we do not expect any significant differences based on age, we wanted to recruit as much a random and representative sample of computer users as possible. For this we didn't restrict ourselves to just University students and recruited a random sample of computer users; the mean age is 32.15 years, with a median of 28 (positive skew), mode (peak) at 25, standard deviation of 11.094, a minimum of 22 and a maximum of 55 years. The above match quite well the age trends reported by Kehoe, Pitkow, Sutton, Aggarwal and Rogers (1999), based on their studies on European users. Our participants were also asked to subjectively rate their computer expertise, on a scale of 1 to 5, where 1 is "Beginner" and 5 is "Expert". Again, no significant differences are expected due to expertise. All demographics are summarised in Table 4.1.

Total	20	
Gender	Male: 15 (75%) Female: 5 (25%)	
Expertise (1-5)	3: 6 (30%) 4: 8 (40%) 5: 6 (30%)	
Ages	22, 23, 24(2), 25(4), 27(2), 29(3), 30, 32, 36, 50, 52, 54, 55	Mean: 32.15 Median: 28 Mode: 25 Std. Dev: 11.094

Table 4.1: Summary of participants' demographics

The experiment variables

Table 4.2 shows the names of the experiment variables that we will use and their meaning; we will use these names throughout the analysis. Remember that the first group of ExpDBs with ExpMSGs is followed by an ExpDB with an UnexpMSG (the "Reset" dialogue box) and then immediately after that follows an UnexpDB (see section "Designing the experiment" of the previous chapter for a more detailed explanation).

Figure 4.1 shows the sequence of the dialogue boxes throughout the experiment, with the "Reset" DBs coming up in positions 15, 25, 30 and 45.

Figure 4.1: The sequence of the dialogue boxes throughout the experiment

Variables	Description	Corresponding DB text
MSECXX where XX = 01-14, 16-24, 26-29, 31-44, 46-50.	The time in msec the participant took to respond to an ExpDB with an ExpMSG.	"Do you want to try the next one?"
MSECR01 - MSECR04	The time in msec the participant took to respond to an ExpDB with an UnexpMSG	"Do you want to reset your score?"
MSECXX where XX = 15, 25, 30, 45.	The time in msec the participant took to respond to an UnexpDB.	"Do you want to try the next one?"
ANSXX where XX = 01-14, 16-24, 26-29, 31-44, 46-50.	The answer (choice of button) the participant clicked on in an ExpDB with an ExpMSG.	"Do you want to try the next one?" (Yes/No)
ANSR01 - ANSR04	The answer (choice of button) the participant clicked on in an ExpDB with an UnexpMSG	"Do you want to reset your score?" (Yes/No)
ANSXX where XX = 15, 25, 30, 45.	The answer (choice of button) the participant clicked on in an UnexpDB.	"Do you want to try the next one?" (Yes/No)

Table 4.2: All the variables measured throughout the experiment

The null hypothesis

Just before we begin our analysis we need to state our null hypothesis (H_0), which is set to be refuted, in order to support H_1. As a reminder, here is H_1, followed by H_0:

H_1: Users tend to *not* read the message of an *expected* dialogue box

H_0: Users *always* read the message of an *expected* dialogue box

According to what we discussed in the previous chapter, if our null hypothesis is correct, then:

1) If "users always read the message of an ExpDB" we should expect: *no difference* between the MSEC of an ExpDB and the MSEC of an UnexpDB, provided that the messages are of similar size and cognitive processing requirements (it would obviously be irrational to compare the MSEC of a trivial to question to the MSEC of a question requiring a complex mathematical answer). Even more specifically, there should be no statistically significant differences if the message is exactly the same. Remember that as discussed in the previous chapter, comparing the MSEC of an ExpDB to that of an UnexpDB provides an indication of whether a participant read the message or not.
2) If "users always read the message of an ExpDB" we should expect: *the correct ANS*, regardless of whether the MSG is expected or unexpected, provided that the correct answer is obvious to the user (i.e. the question is not deceiving)

On the other hand, if the null hypothesis is wrong then we should observe statistically significant differences and in this case we would drop H_0 in favour of our original hypothesis (H_1).

The analysis

We will start our analysis by presenting some descriptive statistics that provide an idea of how participants generally behaved throughout the experiment. Further to that, we perform a Wilcoxon signed-rank test in order to test H_0.

First of all, as we noted in the previous chapter, because of the repetitive nature of the experiment task, there is a danger that there could be a decline of interest from the participants along the way. In order to ensure that this is not the case, our first statistic will be the number of participants that gave a wrong mathematical answer per each of the 50 questions (minimum: 0, maximum: 20). If participants became less interested in the experiment there should be an increase in the number of wrong answers along the way. As Figure 4.2 shows this is not the case, and the maximum number of wrong answers (which is 3) is spread quite evenly, occurring in questions 3, 15, 17, 25, 26 and 41.

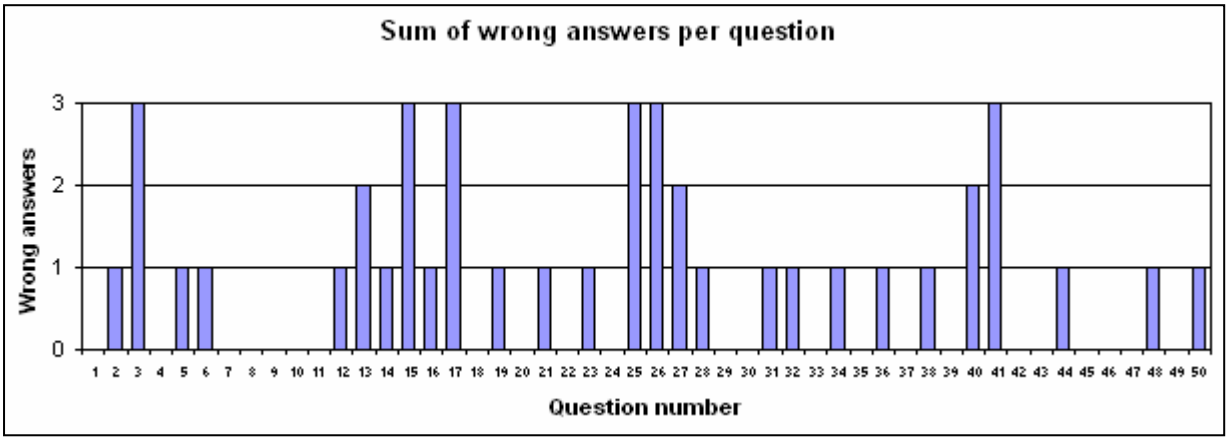


Figure 4.2: Number of participants that gave a wrong answer for each question (min. possible: 0 / max. possible: 20)

As we have now established that participants were focused throughout the experiment, we can start comparing our results. Figure 4.3 on the next page shows the average time (MSEC) each participant spent on all 46 ExpDBs with ExpMSGs compared to the average time (MSEC) they spent on the 4 ExpDBs with the UnexpMSGs (the "reset" questions) and Figure 4.4 shows the corresponding differences. Since the dialogue boxes are in both cases expected, we predict that the alteration of the message should make no difference to the participant and therefore the MSECs should have no significant difference for the same participant (obviously there will be differences between the participants; no two people are the same).

Most cases follow our prediction, with only about 5 exceptions (depending on where we set our threshold). Not surprisingly at all, these 5 participants are the only ones who gave at least one correct response to the 4 "reset" questions (ExpDBs with UnexpMSG). Figure 4.5 shows the total answers (out of 4) that each participant gave to the ExpDBs with the UnexpMSGs. Comparing the participant numbers, we can see that the 5 participants who gave at least one correct answer are the same as the 5 previous exceptions.

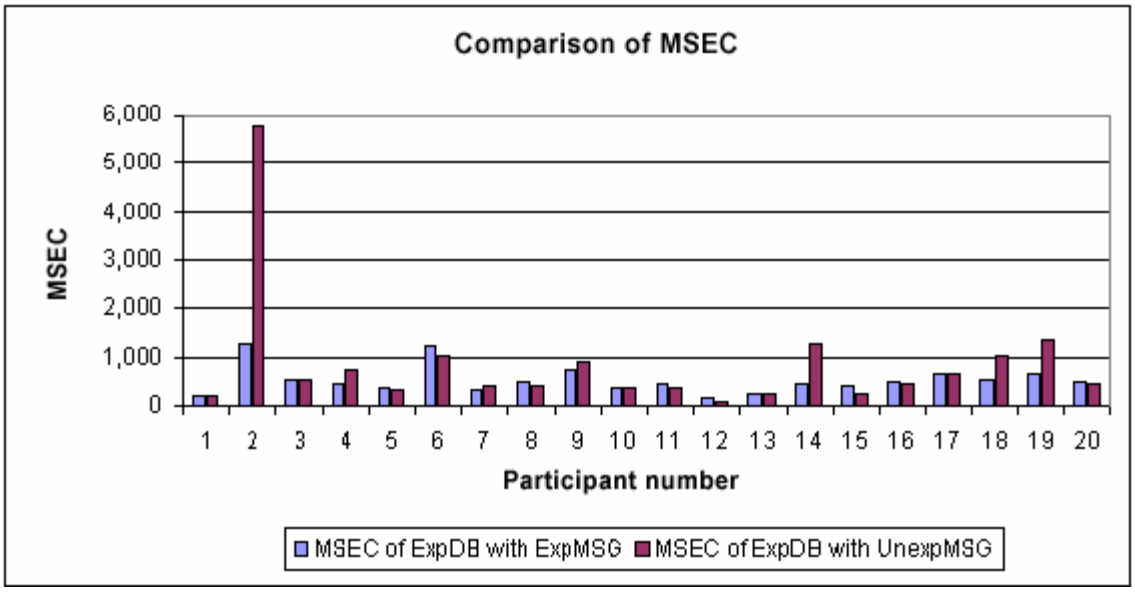


Figure 4.3: Comparison of mean times between ExpDBs with ExpMSG and ExpDBs with UnexpMSG

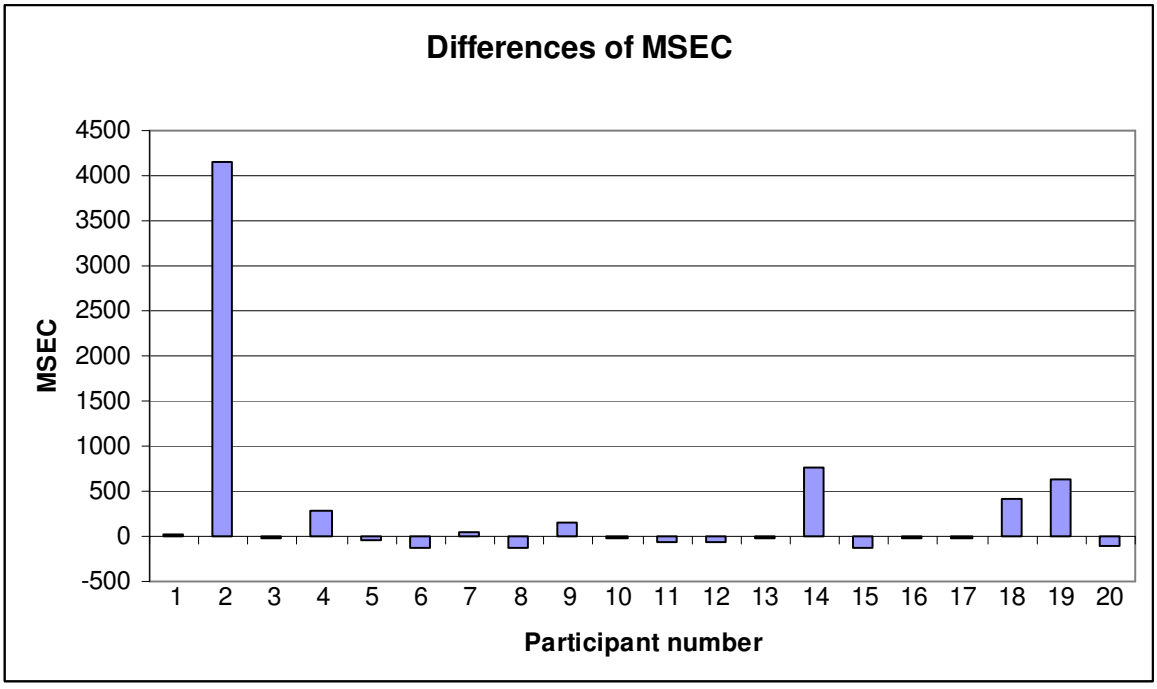


Figure 4.4: Corresponding differences for Figure 4.3

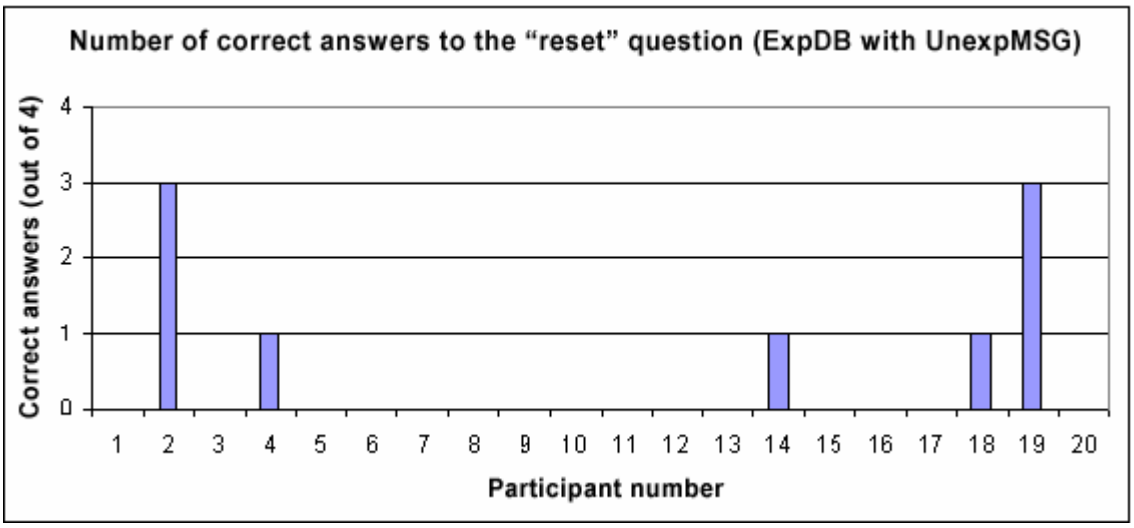


Figure 4.5: Correct answers per participant for each of the ExpDBs with an UnexpMSG

Another promising comparison is presented in Figure 4.6, where for each participant we show the difference between the average MSEC of ExpDBs with ExpMSG and the MSEC of the UnexpDBs. As we expected, the differences are significant, providing evidence that participants spent considerably more time reading the UnexpDBs.

A further statistic worth mentioning is the amount of time participants spent reading ExpDBs with ExpMSGs. The average was 533.30 msec, with a minimum of 161.63 msec, a maximum of 1,272.13 msec and a standard deviation of 289.07 msec. According to Bonnie and Newell (1989), the average time required to read and perceive a 6-letter word is 340 msec; therefore we believe that the average 533 msec from the moment a dialogue box appears until an answer is given is another indication that participants did not actually read the question before answering.

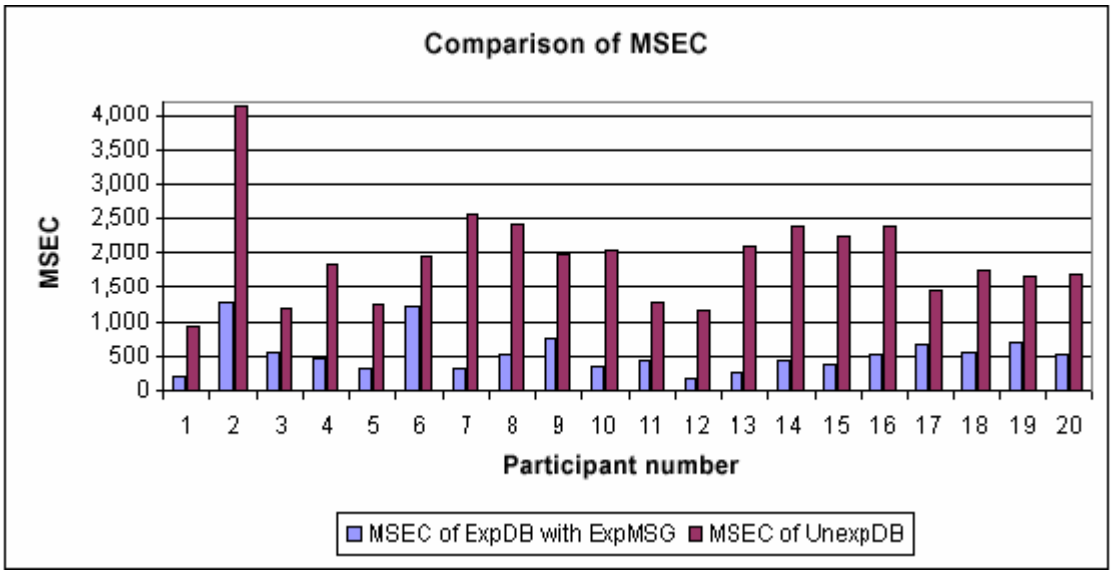


Figure 4.6: Comparison of mean times between ExpDBs with ExpMSG and UnexpDBs

We will now try to get an "overview" of the time that participants took to respond to each of the 54 dialogue boxes. For this we calculated the median MSEC of all participants, separately for each dialogue box and present this in Figure 4.7. As previously, we obviously expect no major differences in the MSEC between ExpDBs with ExpMSGs and ExpDBs with UnexpMSGs, but we expect to see significant differences for the UnexpDBs. In this case we have used the median because we want an overview that is not skewed by individual times that were exceptionally high for any reason (e.g. in one case, one of the participants took 12,641 msec(!) to respond to a dialogue box, which suggests that they were distracted). The results show that the time to answer unexpected dialogue boxes was three to four times higher than the time to answer expected dialogue boxes.

The descriptive statistics for MSEC have been a good start. As we expected, the statistics for ANS are along the same lines. Figure 4.8 shows the total *incorrect* answers given by all users for each dialogue box. The minimum possible is 0 and the maximum possible is 20. If our predictions are right, the incorrect answers should be low for the ExpDBs with the ExpMSG and the UnexpDBs, but high for the ExpDBs with an UnexpMSG, and this is verified in Figure 4.8.

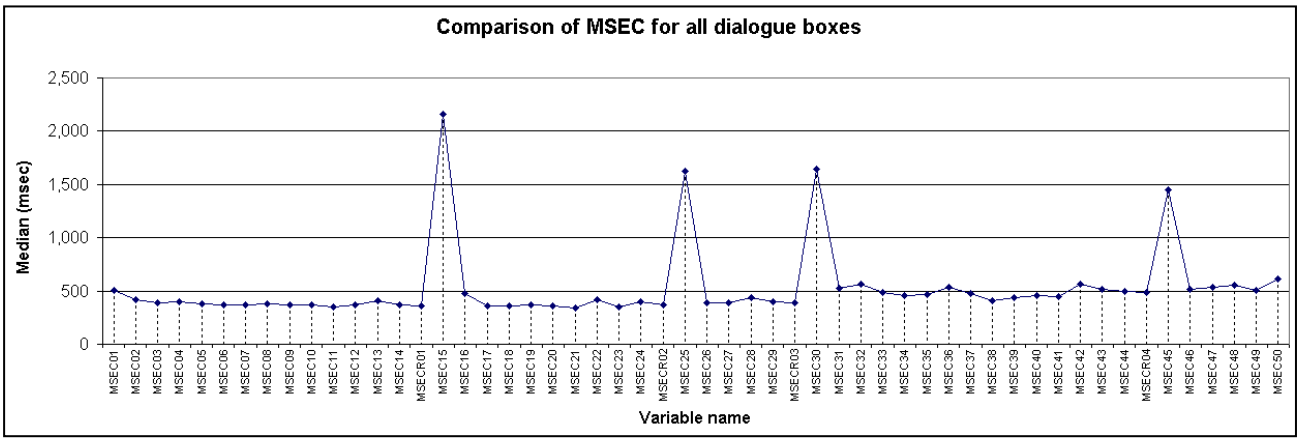


Figure 4.7: Overview of MSEC for all dialogue boxes

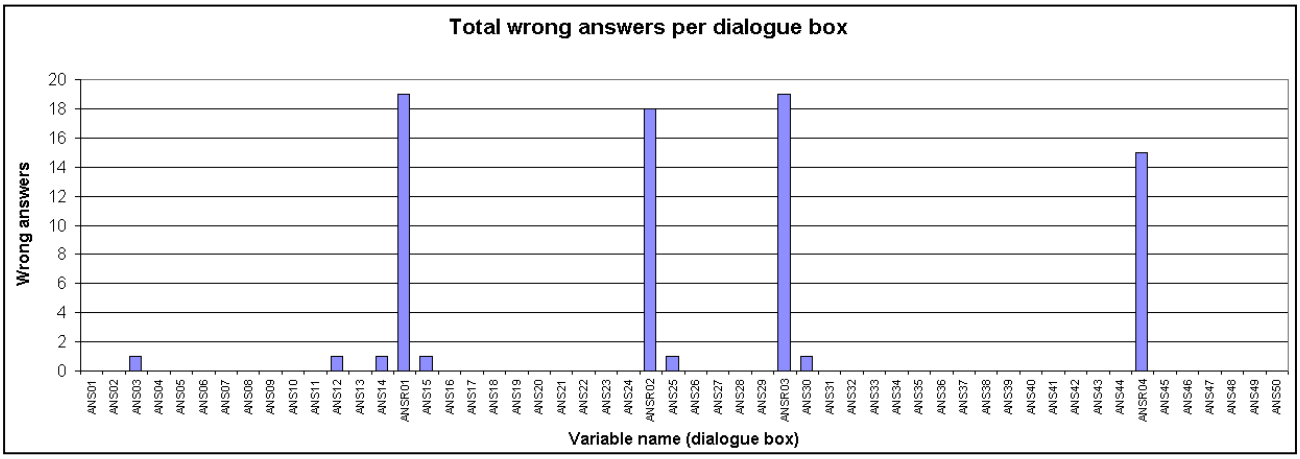


Figure 4.8: Total number of wrong answers per dialogue box (min: 0 / max: 20)

Although in all 4 cases of the "reset" message there was at least one answer which was correct, this was not from the same participant at all times. In other words, none of the participants answered correctly all four times. Table 4.3 shows which participants answered correctly to which ExpDBs with UnexpMSG (the "reset" dialogue boxes).

Participant	RESET01	RESET02	RESET03	RESET04
2	✗	✓	✓	✓
4	✗	✗	✗	✓
14	✗	✗	✗	✓
18	✗	✗	✗	✓
19	✓	✓	✗	✓

Table 4.3: Answers of participants who got at least one correct "reset" answer

Although not of any statistical significance, Table 4.3 shows that only the fourth (and last one) ExpDB with UnexpMSG was answered correctly by all participants that gave at least one correct answer to any of the four ExpDBs with UnexpMSG. Neither of them though answered correctly to all four, not even the one (participant #19) who answered correctly the very first such dialogue box. Participant #19 noted during the follow-up questionnaire: "I noticed it the first time, but missed it when it came back again". An informal qualitative analysis of the short follow-up questionnaires follows in the discussion chapter; as a quick note here, we could possibly infer that participants, after noticing the different DB, started adapting their automatic reactions, realising that the stimulus was not the same in all cases. It may be possible to predict that if there were more ExpDBs with UnexpMSGs after the fourth one, adaptation rates would get progressively higher, with more correct answers, but further research would be needed to confirm this.

Finally, we will now test our H_0 and if we find no significant evidence to support it, we will drop it in favour of H_1. As explained above, our data distribution is skewed and therefore several of the commonly used statistical tests (such as the Student's t-test) which assume a normal distribution are not the most appropriate in our case. Instead, we will use the Wilcoxon signed-rank test (Wilcoxon, 1945), which is non-parametric and therefore does not require assumptions about the form of the distribution of the measurements. All tests were run on SPSS, where all results in this thesis come from.

If H_0 holds and "users *always* read the message of an *expected* dialogue box", then there shouldn't be any statistically significant differences between the MSEC observed for different kinds of dialogue boxes. To test this we will first compare in pairs the average MSEC per user for the ExpDBs with ExpMSGs, ExpDBs with UnexpMSGs and UnexpDBs. For this comparison we will use the four ExpDBs with ExpMSGs that come up just before the four ExpDBs with UnexpMSGs (since users do not know the sequence of expected and unexpected dialogue boxes, it makes no difference which 4 we choose).

We expect to find no difference when ExpDBs with ExpMSGs and ExpDBs with UnexpMSG are compared to each other, but we expect significant differences when these two are compared with UnexpDBs. We accept the typical statistical significance level of $a=0.05$, but because we are testing 3 independent cases using the same data, something that leads to inflation of the alpha level (Abdi, 2007), we need to first apply the Bonferroni correction to the a level (Benjamini and Hochberg, 1995). In our case, this gives $a= 0.05/3 = 0.0167$. Therefore, we now accept $a=0.0167$ for all 3 comparisons, which is even more strict than the initial value of 0.05 and gives a confidence interval of 98.33%. The test was run on SPSS and the results are presented in Tables 4.4 and 4.5, on the next page. The last column of Table 4.5 is of interest to us and shows exactly what we expected, with a $p<0.001$ for pairs 2 and 3, while the first pair fails the test. In other words, there are no significant differences in the way our participants treated ExpDBs with ExpMSGs and ExpDBs with UnexpMSGs. The different message did not cause any change in their reaction time, while the reaction time for UnexpDBs was significantly different.

Variables		N	Mean Rank	Sum of Ranks
MSEC of ExpDB with UnexpMSG - MSEC of ExpDB with ExpMSG	Negative Ranks	8 [a]	9.31	74.50
	Positive Ranks	11 [b]	10.50	115.50
	Ties	1 [c]		
	Total	20		
MSEC of UnexpDB - MSEC of ExpDB with ExpMSG	Negative Ranks	1 [d]	1.00	1.00
	Positive Ranks	19 [e]	11.00	209.00
	Ties	0 [f]		
	Total	20		
MSEC of UnexpDB - MSEC of ExpDB with UnexpMSG	Negative Ranks	1 [g]	14.00	14.00
	Positive Ranks	19 [h]	10.32	196.00
	Ties	0 [i]		
	Total	20		

(a) MSEC of EXPDB with UnexpMSG < MSEC of ExpDB with ExpMSG
(b) MSEC of EXPDB with UnexpMSG > MSEC of ExpDB with ExpMSG
(c) MSEC of EXPDB with UnexpMSG = MSEC of ExpDB with ExpMSG
(d) MSEC of UnexpDB < MSEC of ExpDB with ExpMSG
(e) MSEC of UnexpDB > MSEC of ExpDB with ExpMSG
(f) MSEC of UnexpDB = MSEC of ExpDB with ExpMSG
(g) MSEC of UnexpDB < MSEC of EXPDB with UnexpMSG
(h) MSEC of UnexpDB > MSEC of EXPDB with UnexpMSG
(i) MSEC of UnexpDB = MSEC of EXPDB with UnexpMSG

Table 4.4: Wilcoxon signed ranks test ranks for MSEC

Variables	Z	Asymp. Sig.
MSEC of ExpDB with UnexpMSG - MSEC of ExpDB with ExpMSG	-.825	.409
MSEC of UnexpDB - MSEC of ExpDB with ExpMSG	-3.883	.000
MSEC of UnexpDB - MSEC of ExpDB with UnexpMSG	-3.397	.001

Table 4.5: Wilcoxon signed ranks test results for the MSEC means

Finally, we apply the exact same test to the ANS variable. In order to calculate meaningful values, we coded the variable as 0 for incorrect answer and 1 for correct and calculated the mean over the 4 dialogue boxes as explained above. We expect to find no statistically significant difference between the ANS means for ExpDBs with ExpMSG and UnexpDBs but a statistically significant difference for ExpDBs with UnexpMSG and the other two. Tables 4.6 and 4.7, on the next page, show the results as obtained from SPSS, which prove the above, with a $p<0.001$ for pairs 2 and 3, with pair 1 failing the test as expected.

Variables		N	Mean Rank	Sum of Ranks
ANS of ExpDB with UnexpMSG - ANS of ExpDB with ExpMSG	Negative Ranks	20 [a]	10.50	210.00
	Positive Ranks	0 [b]	.00	.00
	Ties	0 [c]		
	Total	20		
ANS of UnexpDB - ANS of ExpDB with ExpMSG	Negative Ranks	1 [d]	2.00	2.00
	Positive Ranks	1 [e]	1.00	1.00
	Ties	18 [f]		
	Total	20		
ANS of UnexpDB - ANS of ExpDB with UnexpMSG	Negative Ranks	0 [g]	.00	.00
	Positive Ranks	20 [h]	10.50	210.00
	Ties	0 [i]		
	Total	20		

(a) ANS of EXPDB with UnexpMSG < ANS of ExpDB with ExpMSG
(b) ANS of EXPDB with UnexpMSG > ANS of ExpDB with ExpMSG
(c) ANS of EXPDB with UnexpMSG = ANS of ExpDB with ExpMSG
(d) ANS of UnexpDB < ANS of ExpDB with ExpMSG
(e) ANS of UnexpDB > ANS of ExpDB with ExpMSG
(f) ANS of UnexpDB = ANS of ExpDB with ExpMSG
(g) ANS of UnexpDB < ANS of EXPDB with UnexpMSG
(h) ANS of UnexpDB > ANS of EXPDB with UnexpMSG
(i) ANS of UnexpDB = ANS of EXPDB with UnexpMSG

Table 4.6: Wilcoxon signed ranks test ranks for ANS (0=incorrect / 1=correct)

Variables	Z	Asymp. Sig.
ANS of ExpDB with UnexpMSG - ANS of ExpDB with ExpMSG	-4.089	.000
ANS of UnexpDB - ANS of ExpDB with ExpMSG	-.447	.655
ANS of UnexpDB - ANS of ExpDB with UnexpMSG	-4.087	.000

Table 4.7: Wilcoxon signed ranks test results for the ANS means (0=incorrect / 1=correct)

Based on the tests performed above, we can drop the null hypothesis, with a probability that we have made a type I error (that we dropped it when we shouldn't have) which is less than 0.1%!

Therefore, in light of all the evidence presented above, we can now confidently conclude that our original hypothesis, H_1 is correct and that "Users tend to not read the message of an expected dialogue box".

The second experiment

Introduction

The purpose of the second experiment is to investigate whether placing all the ExpDBs with ExpMSGs in the same position (at the centre of the screen) and all the ExpDBs with the UnexpMSG in a slightly different position will produce different results compared to the first experiment. In particular, and based on our literature review, we expect that the different position of the ExpDbs with the UnexpMSG will be encoded as a different stimulus and trigger an action different than the prevailing one (Eysenck and Keane, 2000). We predict that participants will eventually associate (consciously or not) the position of the dialogue boxes with their question and therefore we expect to see more correct ANSs to the ExpDBs with the UnexpMSGs compared to the first experiment.

All definitions remain exactly as we have already used them in the previous chapters. As before, we make the assumption that users are familiar with the system they are using, otherwise the distinction between expected and unexpected dialogue boxes would make no sense.

Hypothesis

Our hypothesis for this experiment is the following:

> H_1: Altering the position of an *expected* dialogue box on the screen will affect an experienced user's automatic action when answering it.

The null hypothesis (H_0) for this experiment, which is set to be refuted in order to support H_1 is the following:

> H_0: Altering the position of an *expected* dialogue box on the screen will have *no effect* on an experienced user's automatic action when answering it.

Finally, because we will compare the data gathered from this experiment to that gathered from the first one, we need to make the same assumptions. Therefore…

> …for the purposes of our working hypothesis, we make the assumption that the available options and layout of the ExpDB remain similar, regardless of whether the message is expected or unexpected.

The variables

Although the setup of the experiment was kept the same as before in order to allow comparison of the results (and therefore we have both expected and unexpected dialogue boxes), here we only concentrate on the position of *expected* dialogue boxes. Our variables here are the following:

Independent variable: The position of expected dialogue boxes (Centre of the screen / Different position)

Dependent variable: Whether users read the dialogue box message before responding, with two possible values (Read / Not read)

The position of the dialogue box is either on the centre of the screen or 150 pixels to the right and 130 pixels lower (the choice of numbers was such that the position of the test dialogue boxes is obviously distinct, but not very far from the same area of the screen as the rest).

As we have discussed, there is no *direct* way of measuring the dependent variable. In our previous experiment we measured MSEC and ANS, but because of the different positions on the screen of some of the dialogue boxes and of the different ways users make their selections (either using the mouse or their keyboard), we cannot directly compare the MSECs. We still measured both in case we observe any interesting changes / trends, but we will get our results by only comparing the ANSs between the two experiments. Just as a reminder:

> **ANS** measures whether the answer given was correct or not. We assume that, out of all the possible choices (buttons) of a dialogue box, only one is correct and the rest are incorrect. We define as correct the one that will bring the user closer to achieving their goal.

For any clarifications and all the assumptions related to ANS, please refer to its definition in the previous chapters.

As before, our hypothesis refers to any type of dialogue box. Nevertheless, because we need to get a meaningful value for the ANS variable, we will only assume the cases of dialogue boxes asking a question and expecting an answer, as opposed to dialogue boxes that merely give information and have only one button (like "OK" or "Close").

Designing the Experiment

The design of the second experiment was very similar to the first one. We held everything constant as far as possible and although obviously we cannot control things like where the participants run the experiment or their comfort

level, we don't expect any such differences to be statistically significant since all participants were randomly selected for both experiments. The relative difference in the position of the DBs is shown in Image 5.1.

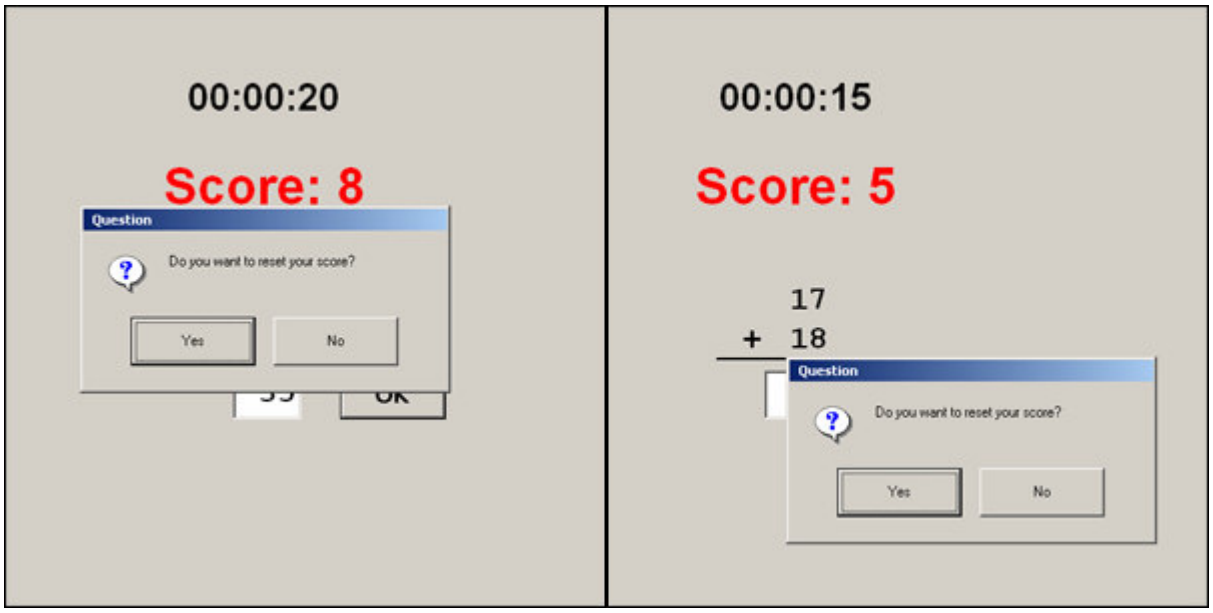


Image 5.1: The relative difference in the position of the DBs

All participants received the same training as in the first experiment and the system recorded the following data:

- Gender
- Year of birth
- Computer expertise (on a subjective scale of 1 - 5)
- Whether each mathematical answer was right or wrong
- The MSEC for each dialogue box
- The ANS to each dialogue box

The 4 ExpDBs with the UnexpMSG are, as before, the 15th, 25th, 30th and 45th. Again, as correct ANS for those dialogue boxes we define "No" instead of "Yes" that subjects have to click in the usual ExpDBs with the ExpMSGs.

The next chapter describes the analysis of the results that we got.

Analysis of the second experiment

Participants' demographics

For the second experiment we recruited again a total of 20 participants, 10 male and 10 female. During the experiments, one of the participants noticed that their score was being reset and decided to rerun the experiment a total of 5 times, until they got all trials right. Therefore we had to disqualify their results and another participant was recruited so that the total remained 20.

This sample has a mean age of 34.2 with a median of 33 (slightly positive skew) and a standard deviation of 8.4. Minimum age was 24 and maximum 53 years. These statistics differ only slightly from those of the first experiment and we don't expect any significant differences because of this. Participants also subjectively rated their computer expertise, on a scale of 1 to 5, where 1 is "Beginner" and 5 is "Expert". As before, no significant differences are expected due to expertise. All demographics are summarised in Table 6.1.

Total	20	
Gender	Male: 10 (50%) Female: 10 (50%)	
Expertise (1-5)	3: 9 (45%) 4: 8 (40%) 5: 3 (15%)	
Ages	24, 25(2), 26, 27(2), 29(2), 31, 33(2), 34, 35, 38(2), 40, 43, 45, 49, 53	Mean: 34.2 Median: 33 Std. Dev: 8.402

Table 6.1: Summary of participants' demographics

The experiment variables

The variables' names have been kept exactly the same as in the first experiment. As a reminder, the first group of ExpDBs with ExpMSGs is followed by an ExpDB with an UnexpMSG (the "Reset" dialogue box) and then immediately after each "Reset" dialogue box follows an UnexpDB (see section "The experiment variables" of the "Analysis of the first experiment" chapter for a more detailed explanation).

The null hypothesis

Here is a quick reminder of H_1 and H_0:

> H_1: Altering the position of an *expected* dialogue box on the screen will affect an experienced user's automatic action when answering it.

> H_0: Altering the position of an *expected* dialogue box on the screen will have *no effect* on an experienced user's automatic action when answering it.

If H_0 is correct, then we should expect no statistically significant difference between the ANS given to ExpDBs with UnexpMSGs (the "reset" DBs) in the first experiment and those given in the second. On the other hand, if H_0 is wrong, then we should observe a statistically significant difference and in this case we would drop H_0 in favour of H_1.

The analysis

We will start our analysis by providing some descriptive statistics, in order to get a general idea of how participants generally responded in the second experiment. Further to that, we perform a Mann-Whitney test for H_0.

As we also noted in the previous experiment, because of the repetitive nature of the task, there is a danger that there could be a decline of interest from the participants along the way. In order to ensure that this is not the case, we will use the number of participants who gave a wrong mathematical answer per each of the 50 questions (minimum: 0, maximum: 20). If participants became less interested in the experiment there should be an increase in the number of wrong answers along the way. As Figure 6.1 shows this is not the case, and the sums of wrong answers are spread quite evenly throughout the experiment. The slightly higher error rate for question 25 can probably be attributed to the fact that that particular mathematical problem was somewhat more difficult to calculate than the rest.

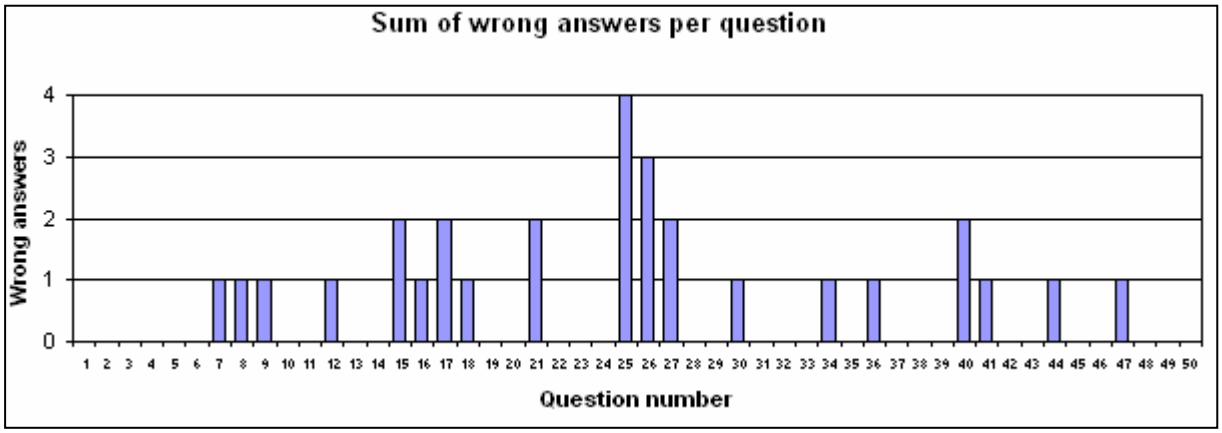


Figure 6.1: Number of participants that gave a wrong answer for each question (min. possible: 0 / max. possible: 20)

We have now established that participants were focused throughout the experiment and we can start comparing our results. As we mentioned in the previous chapter, there is no point in comparing the MSEC between the two experiments. We will just note here that there don't seem to be any worthy differences in the MSEC trends between the two experiments.

Figure 6.2 shows the total answers (out of 4) that each participant gave to the ExpDBs with the UnexpMSGs. For an easier comparison we also include the relevant figure from the first experiment, as Figure 6.3; the differences are obvious.

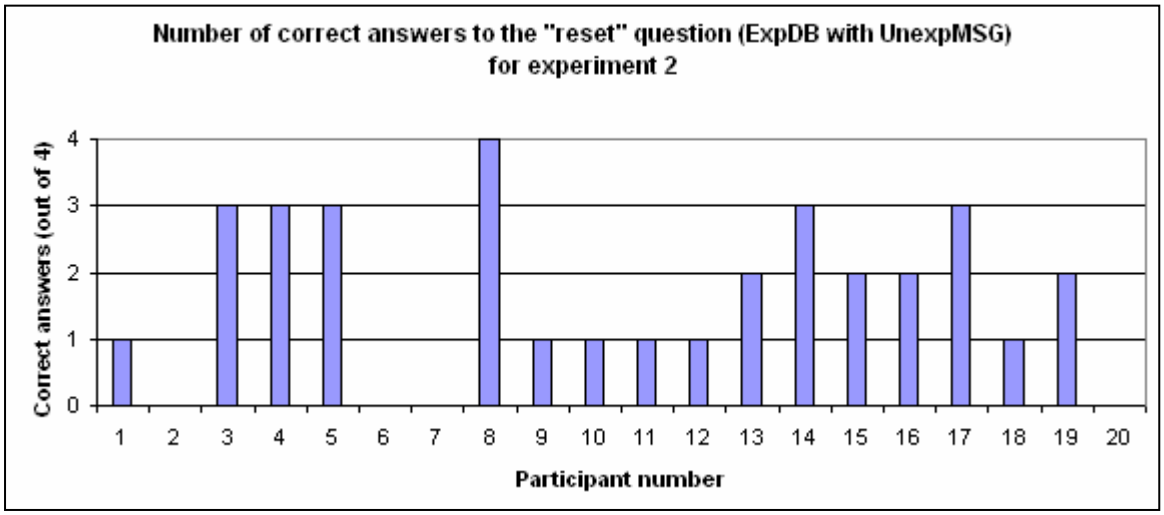


Figure 6.2: Correct answers per participant for every ExpDBs with an UnexpMSG

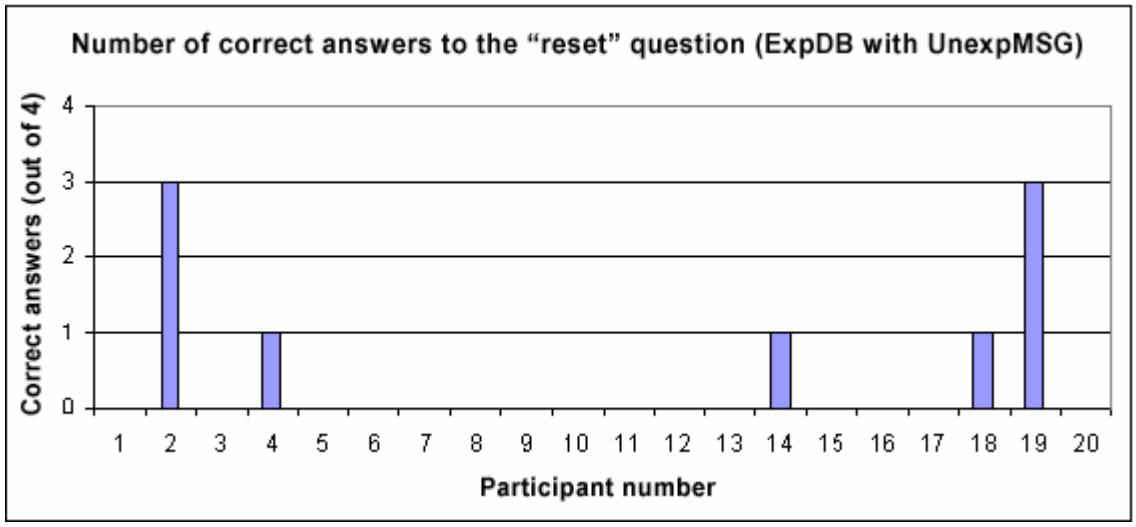


Figure 6.3: The corresponding figure from the first experiment

A direct comparison shows that in the second experiment 16 participants gave at least one correct answer to the "reset" question, compared to just 5 in the first experiment. The total correct answers for all "reset" dialogue boxes were 33 in the second experiment (out of a possible total of 4 x 20 = 80) compared to a mere 9 in the first one. On average, 4 participants answered the first "reset" question right, 9 the second one, 7 the third one and 13 the fourth one. Although the small number of reset dialogue boxes does not allow for definite inferences, there seems to be an increasing trend which would be consistent with the theories about the development of automaticity presented in the second chapter.

Table 6.2 shows a breakdown of the correct answers for each of the participants who answered at least one of the "reset" questions right.

Participant	RESET01	RESET02	RESET03	RESET04	Average
1	×	✓	×	×	0.25
3	✓	✓	×	✓	0.75
4	×	✓	✓	✓	0.75
5	×	✓	✓	✓	0.75
8	✓	✓	✓	✓	1
9	×	✓	×	×	0.25
10	×	×	×	✓	0.25
11	×	×	×	✓	0.25
12	×	×	×	✓	0.25
13	×	×	✓	✓	0.5
14	×	✓	✓	✓	0.75
15	×	✓	×	✓	0.5
16	✓	×	✓	×	0.5
17	×	✓	✓	✓	0.75
18	×	×	×	✓	0.25
19	✓	×	×	✓	0.5

Table 6.2: Answers of participants with at least one correct "reset" answer

For the statistical analysis we will use the non-parametric Mann-Whitney test (which tests the null hypothesis that two independent samples come from the same population) to compare the answers to the ExpDBs with UnexpMSG between the first and the second experiment. If H_0 correct, then we should not see any statistically significant differences between the first and the second experiment results. The test was run on SPSS and results are presented in Tables 6.3 and 6.4.

Variable	Group	N	Mean Rank	Sum of Ranks
MeanOfResets	1st exp.	20	14.70	294.00
	2nd exp.	20	26.30	526.00
	Total	40		

Table 6.3: Mann-Whitney test ranks

	MeanOfResets
Mann-Whitney U	84.000
Z	-3.353
Asymp. Sig.	**.001**

Table 6.4: Mann-Whitney test results

The significance (last row in Table 6.4) of 0.001 allows us to safely reject the null hypothesis, in favour of H_1.

In light of all the evidence presented above, we can now confidently conclude that our original hypothesis, H_1 is correct and that "altering the position of an *expected* dialogue box on the screen will affect an experienced user's automatic action when answering it".

Conclusions and Discussion

Conclusions and Discussion

From our two experiments, we got the following straightforward conclusions, as expressed in their respective hypotheses:

1) Users tend to *not* read the message of an *expected* dialogue box
2) Altering the position of an *expected* dialogue box on the screen will affect an experienced user's automatic action when answering it

These conclusions support our predictions based on the literature review that was presented in the second chapter. Our findings challenge the established convention that dialogue boxes work sufficiently as a safety mechanism against unintended or miscalculated user actions (experiment 1) and suggest that there is room for further improvement (experiment 2). Although the results from the second experiment are statistically significant compared to those of the first one, several participants still failed to even notice that there were differences in the dialogue boxes. Based on our findings and the relevant literature, we predict that a more obvious change in the position of the "reset" DBs would produce a better response rate. In the next section, we will provide some suggestions for design based on the above results.

According to the relative theory, there seems to be an issue with users switching between focused and divided attention. Eysenck and Keane (2000) note that people typically decide whether to engage in focused or divided attention and therefore this is often determined by goal-driven (top-down) attentional control mechanisms. In our case (and in many similar real-life situations), participants' attention shifted to what they considered to be their primary goal (answering the mathematical questions correctly). The task of responding to the dialogue boxes, which also affected their score, gradually became automatic and as we have seen automatic processes are difficult to adapt when the prevailing circumstances change (Eysenck and Keane, 2000). Even though some users reported noticing the different dialogue boxes and their score changing, they still had difficulty giving the correct answer to subsequent ExpDBs with UnexpMSGs. This was partly caused by the fact that both tasks (reading and answering the mathematical questions and reading and answering the dialogue boxes) used the same modality (vision), thus hindering the participants' ability to "multitask" (Eysenck and Keane, 2000).

With regards to our second experiment, we noticed that the different position of the dialogue box can affect the automatic action of an experienced user. This is in accordance to the theory that automaticity is a

memory phenomenon that depends on the relationship between encoding of a stimulus and retrieval of the corresponding learned action (Eysenck and Keane, 2000). By presenting a different stimulus and causing its encoding along with an action different to the prevailing one, we can affect the way a user reacts to expected dialogue boxes. This knowledge is useful in designing systems, as we will see in the next section.

As part of our discussion, we will quote the most interesting comments, as written by participants after they were debriefed, with regards to the above issues.

Comments from participants of the first experiment:

- *"Clicking certain buttons is a habit for me. I also usually don't read the questions very carefully, I just skim them, and unless something unusual hits me, I'll click the standard button"*
- *"I have overwritten documents by accident. Most of the times I don't read the question, because I believe I know what it is"*
- *"For instance once when using an unusual piece of software in France I clicked on 'supprimer' (delete) when I wanted 'imprimer' (print), partly because the icons were nothing like I expected, and the words were similar. A few times I have also chosen 'No' when closing down a word processor (for instance) that was asking if I wanted my changes saved. I meant 'no' to closing down the program, which is actually 'cancel' "*
- *"I have a tendency to respond too quickly without reading properly first"*
- *"I never read the labels of these things after the first go"*
- *"I just click and hope for the best"*
- *"I am more careful when a different dialog box rises, stop and scan its content for a while"*
- *"I noticed that some questions were followed by two dialogue boxes, but I didn't bother to read the text in the second box. My first reaction was to assume it was a bug in the program. I realised it was intentional the second or third time it happened"*
- *"I understood that my score was reset due to the box, but I answered too fast to correct my answer from 'yes' to 'no'. I make often the same mistakes, because sometimes, I tend to not pay attention to some dialogue boxes, i.e. licence agreement boxes"*
- *"No, I really thought this was a program fault. It got me confused, but I didn't get more careful"*

Comments from participants of the second experiment:

- *"Often because I hit return, I tend to go by box shapes and formats. Some are the usual boxes you become familiar with, and the answers of*

yes or no become automatic. I learned some time ago to read three option boxes as they are the 'are you sure you don't want to save your work before closing' boxes"

- *"I wondered why sometimes I had to click OK twice and then I started to notice that the dialogue box appeared in a different place on the screen when it had a different question. Then I looked at my score as I knew I had gotten all the maths correct and noticed it was only on 4! After that I realised that the score was being reset somehow so started paying more attention to the dialogue boxes. Clicking on OK is a nasty habit. I often don't understand the errors that come up on computers so I just click OK to get rid of it... that is what I was doing here. Clicking OK to move on faster without paying any attention to what I was saying OK to. Lesson learned!"*
- *"I was focusing on the mental arithmetic! I can't say it prompted me to be more careful with my answers, I was more worried about the ticking clock to be honest! I usually read the first few words and then remember the question and go for the standard answer ('save changes?' 'yes')"*
- *"I noticed that it was in a different place but took a couple of attempts to realise what it was doing. […] When purchasing things on Amazon I have pressed the 'one-click' button. I never really read pages, and tend to do things out of habit"*
- *"Clicking OK is habit, it's the usual answer to everything"*
- *"I noticed it after the score was reset, and thought 'why did I do that?'. I was more careful the next time. I do usually read the question in a dialogue box, but it does become a matter of habit when you are trying to rush through something"*
- *"When I forgot to save the file, and the computer asked 'are you really sure that you want to quit the programme?', sometimes, I accidentally clicked yes and lost the work. I think it's because I felt that yes is the correct answer for everything, such as 'do you want to save the file?' "*
- *"Once you know what the dialog box is saying then you don't expect it to change!"*

There seems to be a very strong expectation by users that dialogue boxes "do not change" and that if something doesn't work according to their expectations it is a programme fault. Again, this is supported both by the theory and other similar experience as noted in the second chapter (Eysenck and Keane, 2000; Hay and Jacoby, 1996; Toft and Mascie-Taylor, 2005). The implications of these expectations can be detrimental, especially under safety critical circumstances, if the system is trying to warn the user of an exceptional state and the user perceives this as a system malfunction. If we consider the fact that, as technology progresses, most safety systems' feedback tends to be concentrated in a few computer screens and monitored by only a few operators, important readings and system messages might be getting more and more likely to be missed.

On a more everyday basis, accidents like overwriting and deleting important files are quite common as suggested by the participants' comments mentioned above and in the text by Dix, Finlay, Abowd and Beale (1998) quoted in the first chapter. It is apparent that dialogue boxes cannot always prevent such accidents from happening and, even worse, as some participants' comments suggest, sometimes they even provide the false illusion that things went "as expected", only because an expected dialogue box appeared (regardless of its message!) and the user clicked on the "correct" button, as they always do. The mistake is only later discovered by the user, when it is too late to reverse their actions.

Another example comes from installing computer software with the help of "wizards" that guide the user step-by-step through the setup process, with the most common options pre-selected. Newman (as quoted in Norman, 1983) points out that the normal response to requests for confirmation is "*something like this: 'yes, yes, yes, yes. Oh dear!'*". This appears to be a well-known fact to software vendors who (sometimes maliciously) install additional software without the user noticing, just because this was the default option in a dialogue box that appeared at some point during the interaction. Apart from obvious problems like the user not remembering where they installed a programme or what options they included, this provides a "backdoor" to malicious software for altering a user's system in a way the user did not intend to.

Finally, as computer systems are now embedded in devices that were previously purely mechanical, similar interaction issues can arise in areas where they were not previously expected. Familiar examples are photocopier systems that incorporate touch-screens which occasionally display dialogue boxes, requiring input or confirmation from the user. Although the results of a wrong input because the user did not read a message are not detrimental, they can be quite frustrating and, when interpreted as a malfunction of the system, they can make it more difficult to recognise the actual problem and lead to even more frustration.

Implications and Suggestions for design

All the above observations have some direct implications for the design of computer systems and also, as mentioned in previous chapters, experiments that test such systems.

To begin with the latter, as Resnick (2001) notes: "*two types of tasks must be included (in systems' tests). Typical tasks […] that users will be performing on a regular basis while using the system […] when users perform a task repeatedly, parts of the task can become automatic. This reduces the amount of attention that users give to the task, possibly increasing the risk of error or altering the kinds of errors that do occur. If necessary, test participants must be trained sufficiently to reproduce the automaticity that will be reached by expert users*

under real system use. Extreme tasks should also be tested […] requiring users to perform at the limit of perception, strength, motor control or information processing." Our findings from both experiments nothing but support the above recommendations. We believe that it is vital that a system's test participants must be representative of the actual system users' population, especially with regards to experience. Although there are already several guidelines about the participants being representative of the system users' population, most practitioners only consider aspects such as age, gender, social class, ethnic composition etc. In fact, in many cases novice users are being preferred, since they are less expensive and easier to get hold of, and because of a possible perception that they might discover more usability issues than expert users who can circumvent such issues more easily.

On the issue of systems' design, Dix, Finlay, Abowd and Beale (1998) offer some guidelines to make reading any text (obviously including that of dialogue boxes) easier. They mention that the probability of reading a text in a serial way is very low and that experiments have shown that words can be identified as easily as single letters, based on their shape. This means that removing those familiar aspects of a word that help us recognise it (such as converting all characters to capitals) can have a significant effect on the speed and accuracy of reading. Nevertheless, they also mention that words without a meaning are clearer when constantly presented with capital characters. For example flight numbers (EZY123 and not ezy123) and more importantly when stating keys that the user must press (e.g. "Press Q to exit" rather than "press q to exit"). Keeping the above in mind and based on the relevant literature review, we can suggest that making different DBs actually look different as much as possible can help suppress automatic reactions. This can be achieved by using unique icons or splitting text that looks similar (e.g. "do you want to save this document?" and "do you want to delete this document?") in more than one lines, in order to make the relevant DBs have a different layout.

Based on the above discussion, we can appreciate the need for different systems to conform to what is considered "standard" by users; while software creators have used in the past certain "tricks" such as reversed logic in DB questions in order to gain users' attention, this can in fact create more problems when users don't actually read the questions. The fact that unfortunately different programmes have used different DB questions to address the same problem does not help. For example, when closing down a word processor with an unsaved document open, the system could ask two different questions, with two different "correct" answers: either "do you want to save this document" (correct answer is "Yes") or "are you sure you want to quit" (correct answer is "no"). Users experienced in different systems have different expectations and this obviously makes mistakes more likely to happen. Although it is hard to know what the expectations of each and every

user are, it is useful to study a system user's background and relate it to the wording of DBs, especially if the target group is quite specific and with certain training and experience (e.g. pilots, air-traffic and train controllers, nuclear plant operators, emergency services dispatchers etc).

Even more important is the issue of upgrades of a specific system: In no case should the logic of an expected dialogue box be reversed between system upgrades. Users have already formed expectations about the way it operates and expect it to continue operating in the same way. If there is the need for a new dialogue box to be introduced, which could potentially come up where another dialogue box is expected, then these two DBs should be significantly different in their appearance, in order to reduce the possibility of an automatic response.

A solution that has been recently adopted by a few systems (e.g. Mozilla Firefox) is enforcing a "reflection" time, during which the dialogue box is visible but the user cannot select any option. In most cases this can be annoying to an experienced user, especially if it happens for every trivial DB. It is therefore important to only enforce such a pause when it is truly important (e.g. in an ExpDB with an UnexpMSG), rather than adopt it as a constant standard which defeats its own purpose.

Another danger from responding to an expected DB too quickly is that the user could miss the opportunity to mentally or physically record and process critical information that were presented in that DB about the system's state or the outcome of an operation. As Payne (1991) notes, learning to use a device successfully is not a matter of internalising goals but rather a matter of interpreting the state of the device and working out the implications of that state for future actions. It is therefore important that a persistent indicator of a system's state is always provided, or at least upon request. The user should never be at a disadvantage because they missed an important piece of information on a dialogue box, if not only to reduce their memory load.

Another simple but quite often overlooked point in DB design is the actual message wording, especially when messages are created dynamically. A very characteristic example is presented in Image 7.1 on the next page, where the user is prompted to check that their "dial-up networking" settings are configured properly, when the actual error, which is only displayed further down, is that the remote line is busy. As our participants' comments suggest, even if users start reading the error message they are likely to stop before reaching the final part of it. This could cause them the unnecessary trouble of trying to "correct" settings which are not in fact related to the actual problem, possibly causing even more problems. Therefore, the most important part of the message should be displayed as early as possible in a dialogue box.

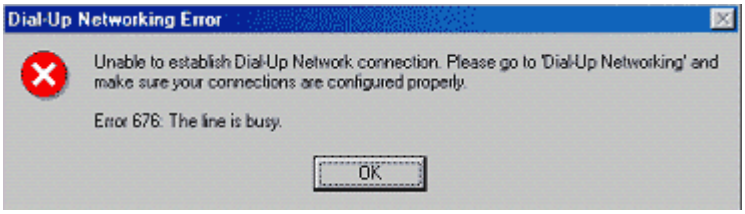


Image 7.1: The line is busy...

Finally, although "guessability" (the user being able to guess the results of a system action) is an important characteristic of good systems' design, it is virtually impossible to provide it for every system and every possible action, especially since different users have different expectations. The problem is that as we have seen, users behave according to their expectations, and sometimes can even perceive error messages as merely expected DBs. Therefore, we should not always rely on users understanding that their "guess" was wrong, especially if feedback is not appropriate. There is the need to improve safety measures since we have shown that dialogue boxes are not always appropriate in giving feedback, as users do not usually expect an error message, but rather the usual informative message of success. Altering the position of an ExpDB with an UnexpMSG or even delaying its appearance (if it can be delayed) and displaying it when it is actually not expected, could improve its effectiveness. A wrong action plan might be triggered and continue to be applied by a user if they fail to recognise that their actions do not have the desired effects, especially in modern software systems where menus and layouts dynamically change. This can be a problem in multitasking environments, where DBs can be triggered by programmes running in the background but considered by the user to be from the active programme. It should be the operating system's responsibility to provide visual stimuli strong enough to distinguish DBs that come from a programme other than the active one, like dimming the screen colours or opening up the programme that triggered the DB; there is still the need for further study on these issues.

To summarise, here is a list of all our proposed suggestions, based on our findings and discussion:

- A system's test participants must be representative of the actual system users' population, especially with regards to experience.
- Make different DBs look different: use unique icons and split similar messages in more than one lines, in order to give the DBs a different layout.
- Different systems' DBs should, as much as possible, conform to what is considered "standard" by the target population, based on their background and experience with similar systems.
- In no case should the logic of an expected dialogue box be reversed between system upgrades.

- If a new dialogue box must be introduced in an existing system design where another dialogue box is expected, then it should be significantly different in appearance, in order to reduce the possibility of an automatic response.
- Enforcing a "reflection" time should be done only when needed and not as a default for every DB; otherwise it defeats its own purpose.
- A persistent indicator of a system's state should always be provided, or at least upon request. A user should never be at a disadvantage because they missed an important piece of information on a dialogue box.
- The most important part of the message should be displayed as early as possible in a dialogue box.
- We should not always rely on users understanding that their guess was wrong, especially if feedback is not appropriate. DBs do not always provide the most appropriate feedback, but altering the position of an ExpDB with an UnexpMSG or even delaying its appearance (if it can be delayed) and displaying it when it is actually not expected, could improve its effectiveness.
- Strong visual stimuli should be provided to the user when a DB is triggered by a programme running in the background and not the one that is currently active, such as dimming the screen colours or switching to the programme that triggered the DB.

Limitations of this study and further work

It is important to note the limitations of our experiments and therefore of our conclusions. As we have previously seen, both of these conclusions refer to the way *experienced* users react to *expected* dialogue boxes. As Resnick (2001) notes, system experts generally interact differently with a system than novices and it is obvious that if a user is not experienced he has no expectations about dialogue boxes. We expect novice users to be more cautious and this is in accordance with Gibson, Newall and Gregor's (2003) findings, as discussed in the "other similar research" section of the second chapter. As one of our participants noted: *"I am usually more careful when it comes to a program I am not familiar with"*.

Another limitation is that, in both experiments, all dialogue boxes were visually similar to each other; this includes the layout and the available options (Yes/No). We expect that if an ExpDB with an UnexpMSG has a layout that is very different from its corresponding ExpDB with an ExpMSG, then users will be "forced" to notice the difference and read the message (good examples are the dialogue boxes in Images 3.3 and 3.4). Nevertheless, if a user's automatic reaction to an expected dialogue box is to press the "Enter" key on their keyboard to respond using the default option, then there is still the possibility

that their automatic reaction will be faster than consciously processing the visual differences. This is consistent with Shiffrin and Schneider's (1977) findings, as discussed in the literature review section; even if users are aware of a change in context, it is still very difficult to suppress or adapt automatic reactions. Disabling the alternative of responding using the "enter" is an option that should be considered only for exceptional circumstances, as discussed above with regards to enforcing a "reflection" time, for the same reasons.

Although automatic processes have a lot to do with the way users' react to DBs, we can never rule out the effects of tiredness, boredom or just plain laziness. Although the effects are similar (users do not read the DB messages), the underlying causes are significantly different and should be treated as such. There is still the need for extensive studying of the combined effects of all the above issues that usually co-exist when an accident happens.

The last issue to note here is one that applies to all similar research; these experiments were designed to train participants in a very specific and repetitive task and measure the required variables over a short period of time. Although we reduced the effects of being under experimental conditions by enabling participants to run the experiment in their own computer, in an environment that they are used to, we still need to consider that everyday practice is probably rather different and more complicated.

New system designs that incorporate dynamic menus and layouts have not been extensively tested and there is yet a lot of work to be done on this field. The effects of reversed logic in critical DB questions have not been extensively measured and there are also issues with experienced users who switch from a system that they are familiar with to a new one. The reluctance of several organisations to upgrade their systems because of such issues and the interrelated issues of training and safety is understandable, but it is a fact that newer generations will need to be trained to use new systems that cannot work in conjunction with the old ones. Gradually, the need to shift to different systems becomes more and more apparent and we need to know the effects in advance and be prepared as much as possible.

We have shown that dialogue boxes are *not* a satisfactory safety mechanism against wrong commands as it is widely thought, but that there is room for improvement. We have emphasised the fact that consideration needs to be given in how we present dialogue boxes to experienced users and we hope that this study will lead to further research and a greater understanding of the issues of suppressing automaticity when it is an undesired effect of repeated exposure to the same stimuli.

References

Abdi, H. (2007). The Bonferonni and Šidák corrections for multiple comparisons. In Neil Salkind (ed), *Encyclopædia of measurement and statistics*. Thousand Oaks (CA): Sage.

Amalberti, R. & Deblon, F. (1992). Cognitive modelling of fighter aircraft process control: a step towards an intelligent on-board assistance system. *International Journal of Man-Machine Studies*. 36, 639-671.

Barshi, I. & Healy, A. F. (1993). Checklist procedures and the cost of automaticity. *Memory & Cognition*, 21(4), 496-505.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57(1), 289-300.

Bonnie, E. J. & Newell, A. (1989). Cumulating the Science of HCI: From S-R Compatibility to Transcription Typing. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Wings for the mind*, p. 109-114. New York: ACM Press.

Dennis, I. & Schmidt, K. (2003). Associative processes in repetition priming. *Journal of Experimental Psychology: Learning, Memory, and Cognition*. 29(4), 532-538.

Desurvire, H. W. (1994). Faster, cheaper!! Are usability inspection methods as effective as empirical testing? In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 173-202). New York: John Wiley & Sons Inc.

Deutsch, S. (2005). *Reconceptualising expertise explaining an expert's error*. In J.M.C. Schraagen (Ed.), *Proceedings of the seventh International NDM conference*. Amsterdam, June 2005.

Dix, A. J., & Finlay, J. E., & Abowd, G. D., & Beale, R. (1998). Human-Computer Interaction. U.S.A.: Prentice Hall

Eysenck M. W., & Keane M. T. (2000). *Cognitive Psychology, A student's handbook*. East Sussex: Psychology Press Ltd.

Gibson, L. & Newall, F. & Gregor, P. (2003). Developing a web authoring tool that promotes accessibility in children's designs. In *Proceedings of the 2003 conference on Interaction Design and children*. Preston, England, 2003.

Harrison, B. L. & Kurtenbach, G. & Vicente, K. J. (1995). An experimental evaluation of transparent user interface tools and information content. In *Proceedings of the 8th annual ACM symposium on User interface and software technology*. Pittsburgh, Pennsylvania, 1995.

Hay, J. F. & Jacoby, L. L. (1996). Separating habit and recollection: Memory slips, process dissociations, and probability matching. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22, 1323-1335.

Karat, C. M. (1994). A comparison of user interface evaluation methods. In J. Nielsen & R. L. Mack (Eds.), *Usability inspection methods* (pp. 203-233). New York: John Wiley & Sons Inc.

Kehoe, C. & Pitkow, J. & Sutton, K. & Aggarwal, G. & Rogers, J. D. (1999). Results of GVU's tenth world wide web user survey. Graphics, Visualisation and Usability Centre, College of Computing, Gerogia Institute of Technology, Atlanta, GA, USA. Retrieved August 29, 2007, from http://www.gvu.gatech.edu/user_surveys/survey-1998-10/

Klein, G. (2000). *Sources of power: How people make decisions*. Cambridge, MA: MIT Press

Langer, E. J. (1989). *Mindfulness: Choice and control in everyday life*. London: HarperCollins.

Logan, G. D. & Taylor, S. E. & Etherton, J. L. (1996). Attention in the acquisition and expression of automaticity. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22, 620-638.

Logan, G. D. (1990). Repetition priming and automaticity: Common underlying mechanisms? *Cognitive Psychology*, 22, 1-35.

Meshkati, N. (1991). Human factors in large-scale technological systems' accidents: Three Mile Island, Bhopal, Chernobyl. SAGE, *Organization and Environment*, 5 (2), 133-154.

Norman, D. A. & Shallice, T. (1986). Attention to action: Willed and automatic control of behaviour. In R. J. Davidson, G. E. Schwartz & D. Shapiro (Eds.), *The design of everyday things*. New York: Doubleday.

Norman, D.A. (1983). Design rules based on analyses of human error. *Communications of the ACM*, 26(4), 254 – 258.

Payne, S. J. (1991). Display-based action at the user interface. *International Journal of Man-Machine Studies*. 35, 275-289.

Resnick, M. L. (2001). *Task based evaluation in error analysis and accident prevention*. Paper presented at the Human Factors and Ergonomics Society 45th annual meeting, retrieved July 2007.

Robertson, I. H. & Manly, T. & Andrade, K. & Baddeley, B. T. & Yiend, J. (1997). "Oops!" Performance correlates of everyday attentional failures in traumatic brain injured and normal subjects. *Neuropsychologia*, 35, 747-758.

Shiffrin, R. M. & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review*, 84, 127-190.

Toft, B., & Mascie-Taylor, H. (2005). Involuntary automaticity: a work-system induced risk to safe health care. *Health Services Management Research*, 18, 211-216.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin,* 1(6), 80-83.

Appendix I

Participant demographics

First experiment

#	Year of birth	Sex	Expertise (1–5)
1	1982	F	5
2	1953	F	3
3	1975	F	4
4	1982	F	4
5	1977	M	4
6	1957	F	4
7	1978	M	3
8	1978	F	4
9	1952	F	3
10	1971	M	4
11	1983	F	5
12	1955	F	4
13	1980	F	3
14	1982	F	3
15	1983	M	3
16	1982	F	5
17	1985	M	5
18	1978	F	5
19	1980	F	5
20	1984	F	4

Second experiment

#	Year of birth	Sex	Expertise (1–5)
1	1969	F	5
2	1967	M	4
3	1972	M	4
4	1974	M	5
5	1958	F	3
6	1980	F	4
7	1980	M	3
8	1982	F	3
9	1978	M	4
10	1982	M	3
11	1981	F	3
12	1973	M	4
13	1983	F	3
14	1962	F	3
15	1964	F	4
16	1974	F	3
17	1969	M	5
18	1978	M	4
19	1976	M	4
20	1954	F	3

Appendix II

Raw experiment results example

This is an example of how raw results were recorder from the experiment software. In the first line, the first number is a random number assigned to the participant, the second number represents their sex (0=female, 1=male), the third number is their year of birth and the fourth number is their subjective expertise level, on a scale of 1-5 (1 being a novice, 5 being an expert).

In subsequent lines: the first column is a counter of the current mathematical problem being displayed; the second column shows whether the participant's answer to the problem was correct or incorrect; the third column is the number of msec they took to respond to the DB since its appearance on the screen; the fourth columns shows their response to the DB question; the fifth column shows the counter of seconds that the user sees during the experiment; finally, the last column shows the total number of correct mathematical answers, as shown to the participant. Obviously, if they answer "yes" to the "reset" dialogue box, this number will be reset to zero.

In this example, the participant answered "no" to the first, second and fourth "reset" DB, but missed the third. The "reset" DBs can be easily found, as they correspond to the first row of the repeating mathematical problem numbers (15, 25, 30 and 45). These numbers appear to repeat because two dialogue boxes were actually displayed one after the other, as explained in the relevant chapters: an ExpDB with an UnexpMSG and an UnexpDB.

3141171	1	1974	3		
1	#TRUE#	609	YES	00:00:03	1
2	#TRUE#	406	YES	00:00:05	2
3	#TRUE#	437	YES	00:00:07	3
4	#TRUE#	391	YES	00:00:09	4
5	#TRUE#	422	YES	00:00:11	5
6	#TRUE#	438	YES	00:00:14	6
7	#TRUE#	437	YES	00:00:16	7
8	#TRUE#	391	YES	00:00:18	8
9	#TRUE#	438	YES	00:00:20	9
10	#TRUE#	515	YES	00:00:22	10
11	#TRUE#	422	YES	00:00:25	11
12	#TRUE#	625	YES	00:00:27	12
13	#TRUE#	458	YES	00:00:29	13
14	#TRUE#	450	YES	00:00:31	14
15	#TRUE#	2000	NO	00:00:36	15
15	#TRUE#	906	YES	00:00:37	15
16	#TRUE#	462	YES	00:00:39	16
17	#FALSE#	453	YES	00:00:44	16

18	#TRUE#	438	YES	00:00:47	17
19	#TRUE#	422	YES	00:00:50	18
20	#TRUE#	421	YES	00:00:51	19
21	#FALSE#	390	YES	00:00:55	19
22	#TRUE#	406	YES	00:00:57	20
23	#TRUE#	375	YES	00:01:00	21
24	#TRUE#	516	YES	00:01:02	22
25	#TRUE#	1422	NO	00:01:07	23
25	#TRUE#	984	YES	00:01:08	23
26	#TRUE#	422	YES	00:01:10	24
27	#TRUE#	390	YES	00:01:12	25
28	#TRUE#	422	YES	00:01:14	26
29	#TRUE#	485	YES	00:01:16	27
30	#FALSE#	451	YES	00:01:19	27
30	#FALSE#	2610	YES	00:01:22	0
31	#TRUE#	390	YES	00:01:24	1
32	#TRUE#	422	YES	00:01:27	2
33	#TRUE#	406	YES	00:01:29	3
34	#FALSE#	422	YES	00:01:32	3
35	#TRUE#	391	YES	00:01:34	4
36	#TRUE#	453	YES	00:01:36	5
37	#TRUE#	437	YES	00:01:39	6
38	#TRUE#	406	YES	00:01:41	7
39	#TRUE#	460	YES	00:01:43	8
40	#TRUE#	484	YES	00:01:46	9
41	#TRUE#	453	YES	00:01:48	10
42	#TRUE#	406	YES	00:01:51	11
43	#TRUE#	422	YES	00:01:54	12
44	#TRUE#	469	YES	00:01:57	13
45	#TRUE#	1219	NO	00:02:00	14
45	#TRUE#	687	YES	00:02:01	14
46	#TRUE#	500	YES	00:02:03	15
47	#TRUE#	440	YES	00:02:05	16
48	#TRUE#	455	YES	00:02:07	17
49	#TRUE#	437	YES	00:02:09	18
50	#TRUE#	578	YES	00:02:11	19