MSc Thesis

“Evaluation of a novel interface for browsing lists”

by Xavier Klingenfus

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, 2007

Abstract

The study aimed at evaluating a novel navigation system for browsing digital lists of items. The particularity of the interface is that it relies of the small world phenomenon. Items in the lists are considered as elements that own two regular connections (previous and next item) but also a potential random link to any other item in the list, which should allow speeding up the scrolling. Two experiments showed that the system is better than a regular one by outputting better time performances (on long lists) but equally that it does not yet compete with existing quick systems in terms of task completion times and usability.

Acknowledgement

Several people gave me a precious help throughout the entire project and thus I would very much like to thank:

- Paul Cairns, my supervisor, who inspired the study and provided me with inestimable guidance and insight,

- Kim and Adam, the remarks of whom were probably a small step to them but a giant leap for me,

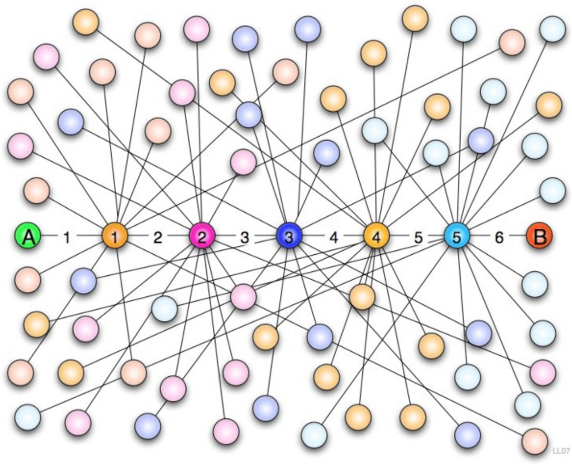
- and all the participants who generously agreed to take part in my experiments.

Table of content

1. Introduction

A small world

In 1967, Stanley Milgram enlightened the social psychology community with his *Small World* experiment (1). This theory also called *Six degrees of separation* states that "if a person is one 'step' away from each person he or she knows and two 'steps' away from each person who is known by one of the people he or she knows, then everyone is no more than six 'steps' away from each person on Earth" (4). The underlying principle behind relies on the fact that within a network of people, everyone is linked to each other by a certain number of intermediate connections. The theory does not only pertain to real persons and can be conceptualised: within a network of interconnected nodes or elements in a small world configuration, every item is linked to each other by six intermediaries. In practice and with respects to spatial representations, every element has many close connections but also some unexpected long-range connections. To summarize the concept, the idea is that **nothing is too far away** within a network based on the small world phenomenon.



A is separated from B by 6 steps
http://en.wikipedia.org/wiki/Image:Six_degrees_of_separation.png

Controlling the world

Ever since, this paradigm has influenced many community networks in various domains, such as mathematics (3) or hypermedia (5). But the important point is not to know that any element can be reached within six intermediate steps: it is rather to be aware that a short path exists between any two elements. As an illustration, a study proved that every piece of information on the Internet is available through 19 clicks, i.e. every page is 19 clicks away from each other (2). But in order to turn this phenomenon into a feasible way to effectively find information, an important factor is missing. Being aware of the sole existence of a short path linking any two elements is not sufficient: the **network needs to be structured** to be efficiently used. It is of paramount importance to know about the positions of the nodes. The solution to solve this issue would be to sort or classify each node. The best way to achieve that is to arrange them by alphabetical order according to their respective identification ('names' in the case of people as the nodes). If the elements can be spatially represented (for instance, John being 'after' Johannes but 'before' Jonathan in a list of names), then the path between two nodes can be more easily identified since the starting node, the ending node and the intermediate nodes will be known. Let us assume that, in a list of people arranged by their first name by alphabetical order, there is only one name for each letter of the alphabet and each person only knows the two names encapsulating him/her plus one random name in the list. If Adrian wants to be introduced to Patrick, whom he does not know yet, we can acknowledge that his random connection should better be with Roger rather than with Daniela because there are supposedly fewer intermediate contacts between Roger and Patrick than between Daniela and Patrick. It will then take Adrian less time to meet Patrick.

Conformist subjects

To be able to reason so, the network needs another characteristic: the system has to be **isomorphous**. It is only by supposing that every node behaves the same way and have the same properties that grounded decisions can be made. It is only by supposing that Roger has two friends – Quentin and Suzanne –, one of whom being himself friend with Patrick (…- Patrick – Quentin – Roger – Suzanne – …).

The idea

These three metrics, namely the awareness of the existence of a short path in a network, its alphabetical structure and the common properties across the set of nodes it contains allowed Paul Cairns to devise an algorithm underpinned by these fundamental characteristics. The domain of application of his algorithm concerns the navigation across lists of items. Although the specifics of the system will be reviewed later, a glimpse can be given: the principle of this algorithm is that it provides the lists it is implemented in with shortcuts which allegedly enable a faster navigation by allowing random long-range connections between items.

The structure

This report presents then the evaluation of this novel navigation type. Consequently, a couple of research questions arises:

- Does this system bring a significant added-value in terms of browsing lists more effectively?

- Is this system more powerful and more usable than current or existing systems?

- Would people be keen to adopt such a system, how easy to assimilate would it be?

- What are the limitations of such a novel navigation system?

- Where could this system be possibly implemented? Where should it not be?

To fully answer these questions, two main experiments have been carried out. The first one intended to prove the added-value of the system. The novel interface, named *Small Networks* (SN), was compared to a traditional list without any 'quick access' technique. The participants were set up in the context of use of a mobile phone contact list. The second experiment sought to challenge the SN system with two current systems providing a shortcut technique to access the elements contained in the list. The rationale was to check whether the novel system could compete with regards to performances with the existing systems. The interfaces simulated here a MP3 player playlist. For both experiments, the participants were asked to perform tasks and completion times were recorded, as well as their experience using the different interfaces, through a System Usability Questionnaire. The last sections of the thesis will be mainly devoted to a discussion about the utility of the system and an analysis on its restrictions and the potential areas that would need a deeper investigation.

References

1. Milgram S., (1967) "The small-world problem", *Psychology Today, volume 1*

2. Reka A., Jeong H. and Barabasi A-L., (1999) "The diameter of the world wide web", *Nature 401, 130-131*

3. *"Collaboration Distance"*, American Mathematical Society, available at: http://www.ams.org/mathscinet/collaborationDistance.html
Accessed the 4th of August 2007

4. *"Six degrees of separation"*, Wikipedia definition, available at: http://en.wikipedia.org/wiki/Six_degrees_of_separation
Accessed the 4th of August 2007

5. *"Six degrees of Wikipedia"*, available at: http://tools.wikimedia.de/sixdeg/
Accessed the 4th of August 2007

2. Literature review

In this chapter, I will present a brief summary of the past or current researches on, first, attempts to improve navigation interfaces and then on small world related work. The last section will examine further the compatibility between IT and this phenomenon.

1. Speedy navigation

Browsing lists is an activity that belongs to the category of the lengthy tasks, such as scrolling, navigating through interfaces or searching. Although they can be performed quite quickly, it is rarely as fast as it should be for users. Sometimes, it can even become laborious. For instance, finding one particular picture within a photo album: if no specific display is turned on (for example, the thumbnail view), the task will take a very long time depending on the size of the album. With systems growing up exponentially in complexity and in content, navigation will play in the future, if not already, an important role in improving usability and performances in task completion processes. This sheds light on the need of having interfaces providing smooth and fast navigation since accuracy and speed are the two factors regarded by the users. These are the sectors an enhanced usability should impact on.

Speeding up navigation has been the research topic of numerous studies, unveiling in the same occasion several techniques judiciously employed. Work has been done on adaptive shortlist [6] where the underlying principle consists of monitoring the user operations to provide afterward a customised interface. In this paper, the authors devised a "smart agent" embedded in the browser that records the visited web pages and creates a bookmark system where the most viewed pages appear first. The more a page is browsed, the more it is likely to be browsed again, the more easily and rapidly accessible it should then become. Other studies related to probability have been carried out by Thimbleby. He showed that on a mobile phone, the number of keypresses needed to attain a particular feature could be reduced by the adoption of a Huffman tree which would sort out the functionalities by likelihood of occurrence [16]. Still on mobile phones, he also demonstrated that the use of a hash code as a shortcut for accessing functions was more performing that the traditional approach (i.e. typing **7** **3** **2** on the keyboard to access the **S** **E** **A**rch function)[17]. Similarly, Cockburn and Gin concentrated on reducing the amount of time spent on navigating through menus [3] by enlarging the menu active areas and by dropping the delay in the display when entering leaving these areas. Cockburn and al. also focussed on improving user performances when scrolling through digital documents [4]. They devised an ingenious overview display with thumbnails that proved to be faster than traditional scrolling. In addition, Cockburn and Smith [5] examined as well the added value of code elision in a development framework and attested that programmers would need less time to complete tasks using this scheme versus the usual display.

The examples previously cited do not only show alternative designs but also points out that there exist faster solutions for a large variety of applications. Therefore, this can hint at quicker ways to browsing lists as well.

2. The small world networks

A network following the “Small World” principle is a structure where each of its composing nodes “are not neighbours of one another, but most nodes can be reached from every other by a small number of hops or steps” [19]. In other words, entities spatially distant from each other can be linked by mutual connections.

Originally studied in social sciences [10], this theory found applications in a large variety of domains. Even though it has been popularized and used for more recreational activities (collaboration between actors [21] and mathematicians [20]), it has also many concrete and serious purposes. In natural sciences, the theory allows to understand, explain and predict phenomena as it has been discovered that a certain number of natural structures were small worlds or had similar properties. As an example, it has been observed that networks of proteins with connections obey to the small world paradigm [2]. A study also showed that in genetics, transcriptional regulatory networks are small worlds [12] as well as the biological metabolism of certain organisms can be described in the same way [7]. Similar patterns can also be identified in food webs [11]. In the industry, some man-made structures follow equally the small world network models. For instance, the Indian railway network turned out to have small world properties [15]. Generally, road maps, electric power grids, neural networks or phone call graphs are all systems that can show signs of small world networks characteristics.

But now the question is to know how this can be relevant to our problem. Let us consider the example of the spread of an infectious disease [9]. If the configuration is a small world network where nodes are people, it is quite simple to imagine that the propagation of the disease would be more rapid than under regular conditions. More broadly, “models of dynamical systems with small-world coupling display enhanced signal-propagation speed, computational power and synchronizability” [18]. This refers to the notion of speed of information propagation. In other words, within a small world network, information is transmitted faster than in regular lattices.

3. Small world and IT

This idea of having ubiquitous small world networks in a large array of disciplines can apply as well to computer sciences and the information technologies sector. A couple of researches have proven that these structures already pervade the Internet itself [1] [14] and online communities [8]. Similarly, Puniyani and al. coupled the small world theory and scale free graphs to study quick navigation through search strategies [13]. In each of the example cited in this chapter, the small world theory relates to the notion of bringing closer distant entities, the notion of shrinking distance between apparently far-away objects. The outcomes of that is the enhanced speed in communication within those networks, which can be translated into a considerable gain of time. Therefore if we can understand, structure and control the small world effect, we can envisage reversing the process and purposely create systems with small world characteristics (by contrast with "discovering" small world networks in existing systems or natural phenomena), thus with improved rapidity and easier information transfer. Developing an algorithm that would enable to propagate data the same way as a virus propagates by infecting people would open new horizons and bring up a novel means of exchanging information.

As said in the opening of the chapter, browsing and scrolling through lists are lengthy activities, all the more when lists get larger. Then, why not trying to implement a small world based system permitting a fast navigation for lists? This is what the present report deals with.

References

1. Adamic L., (1999) "The small world web", *Proceedings of ECDL, volume 1696, 443-452*

2. Bork P., Jensen L., von Mering C., Ramani A., Lee I. and Marcotte E., (2004) "Protein interaction networks from yeast to human", *Current Opinion in Structural Biology, volume 14, 292-299*

3. Cockburn A. and Gin A., (2006) "Faster cascading menu selections with enlarged activation areas", *Proceedings of Graphics Interface 2006, 65-71*

4. Cockburn A., Gutwin C. and Alexander J. (2006) "Faster document navigation with space-filling Thumbnails", *Proceedings of the SIGCHI conference on Human Factors in computing systems, 1-10*

5. Cockburn A. and Smith M., (2003) "Hidden messages: evaluating the efficiency of code elision in program navigation", *Interacting with Computers, volume 3, 387-407*

6. Debevc M., Meyer B. and Svecko R., (1997) "An adaptive short list for documents on the World Wide Web", *IUI 97, 207-211*

7. Fell DA. and Wagner, A., (2000) "The small world of metabolism", *Nature Biotechnology, volume 18, 1121-1122*

8. Iamnitchi A. and al., (2004) "Small-world file-sharing communities", *INFOCOM 2004, volume 2, 952-963*

9. Kuperman M. and Abramson, G., (2001) "Small world effect in an epidemiological model", *Physical review letters, Volume 86, 2909-2912*

10. Milgram S., (1967) "The small-world problem", *Psychology Today, volume 1*

11. Montoya J. and Sole, R., (2002) "Small world patterns in food webs", *Journal of theoretical biology, volume 214, 405-412*

12. Potatov A., (2005) "Topology of mammalian transcription networks", *Genome Informatics, volume 16, 270-278*

13. Puniyani A., Lukose R. and Huberman B., (2001) "Intentional walks on scale free small worlds", *arXiv:cond-mat/0107212v1*

14. Reka A., Jeong H. and Barabasi A-L., (1999) "The diameter of the world wide web", *Nature 401, 130-131*

15. Sen P., Dasgupta S., Chatterjee A., Sreeram PA., Mukherjee G. and Manna S., (2003) "Small-world properties of the Indian railway network", *Physical Review E, volume 67, 036106.1-036106.5*

16. Thimbleby H., (2000) "Analysis and simulation of user interfaces", *Proc. BCS Human Computer Interaction 2000, volume 14, 221-237*

17. Thimbleby, H., (2001) "The computer science of everyday things", *Australian computer science communication, volume 23, 3-12*

18. Watts D. and Strogatz S., (1998) "Collective dynamics of 'small-world' networks", *Letters to nature, volume 393, 440-441*

19. *"Small world network"*, Wikipedia definition, available at:
http://en.wikipedia.org/wiki/Small-world_network
Accessed the 27th of August 2007

20. *"The Erdos number project"*, Erdos Number project webpage (Oakland University), available at:
http://www.oakland.edu/enp/
Accessed the 27th of August

21. *"The six degree of Kevin Bacon"*, Roger Hiemstra's webpage (University of Syracuse), available at:
http://www-distance.syr.edu/bacon.html
Accessed the 27th of August 2007

3. The *Small Networks* interface

This section of the thesis will briefly flesh out what the “Small Networks” (SN) interface tested in this study consists of and how it concretely works when implementing a list of elements.

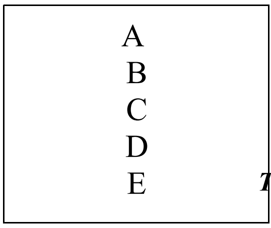
1. In theory

Relying on the small world phenomenon, the SN interface is a system aiming at allowing a quick navigation within lists of elements as it was purposely built under a small world network configuration. It provides an effective and novel way to browse the lists with increased performances when attempting to reach any element. The particularity of this navigation type is that two elements belonging to a network of interconnected entities and apparently spatially far from each other can be linked by short connections. Concretely, the theory applied to browsing lists states that there should be a way to access any element from any position in a list in a shorter time than a pure item-by-item (one by one) scrolling would take, that is a means that would permit a faster navigation through fewer intermediate steps. The resulting decreased task completion time would stem from the fact that the path that was gone across to reach an item contained fewer steps. Therefore a lessened distance would be covered: in other word, a shortcut would be used. In this case, a step basically means a click on a button. Let us say that we have a list containing the alphabet: twenty-six letters, arranged in the order we all know. In a classic setup, travelling from A to Z would require 25 moves. In a small world configuration, going from A to Z should take X moves where $X<25$, this only with a special scrolling type where unexpected connections between letters could save some moves.

2. In practice

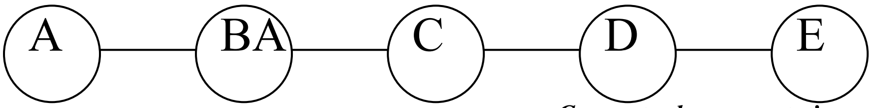
To completely understand the principle underpinning the system, we have to define two things: what a connection is and how it allows to travelling faster.

To clarify what is meant by the term *connection*, let us come back to our list containing the alphabet. Here is how it could generally appear on a screen:



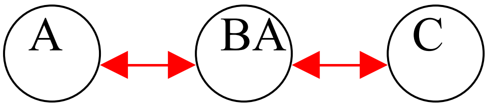
Traditional screen display

But conceptually speaking, the representation can resemble the following:



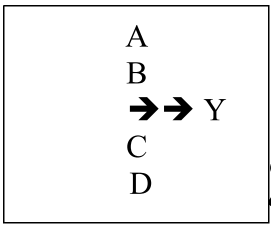
Conceptual representation

A connection is what links two entities, two elements in a list. It is a virtual path that allows a two-ways communication between the two elements it connects. In a normal configuration, each entity owns two connections (unless the list does not allow cycling in which case the first and last entities only have one connection).



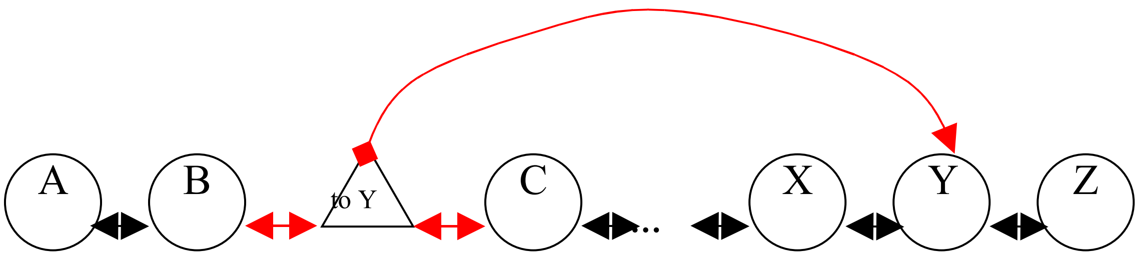
Two two-ways connections for the entity B

As mentioned before, the strength of the SN interface dwells in that it saves moves by providing random long-range connections. This means that some random nodes in the list will connect to some different random nodes, elsewhere in the list. This is realised by the creation of a special entity which is not of the same kind as the regular ones. On a display, this new entry will then have three connections: two two-ways links with the previous and next item in the list and one one-way long-range link with another random item. However, conceptually, the link will be considered as the third connection from the previous regular element. The following example shows an example of long range connection from B to Y:



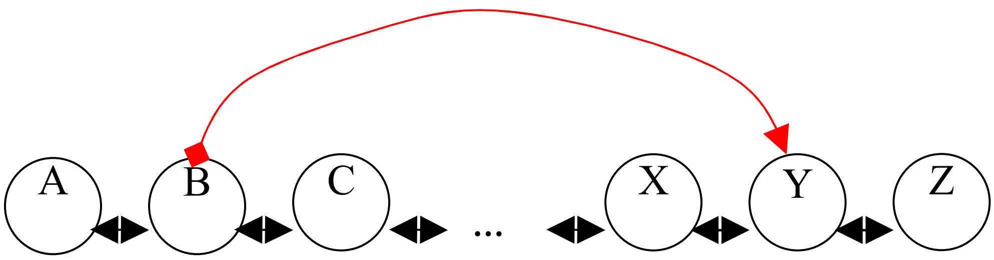
Graphical representation (on the display): B has two two-ways connections (A and C) and one link (Y).

The user will then build a mental model of the list, resembling the following:



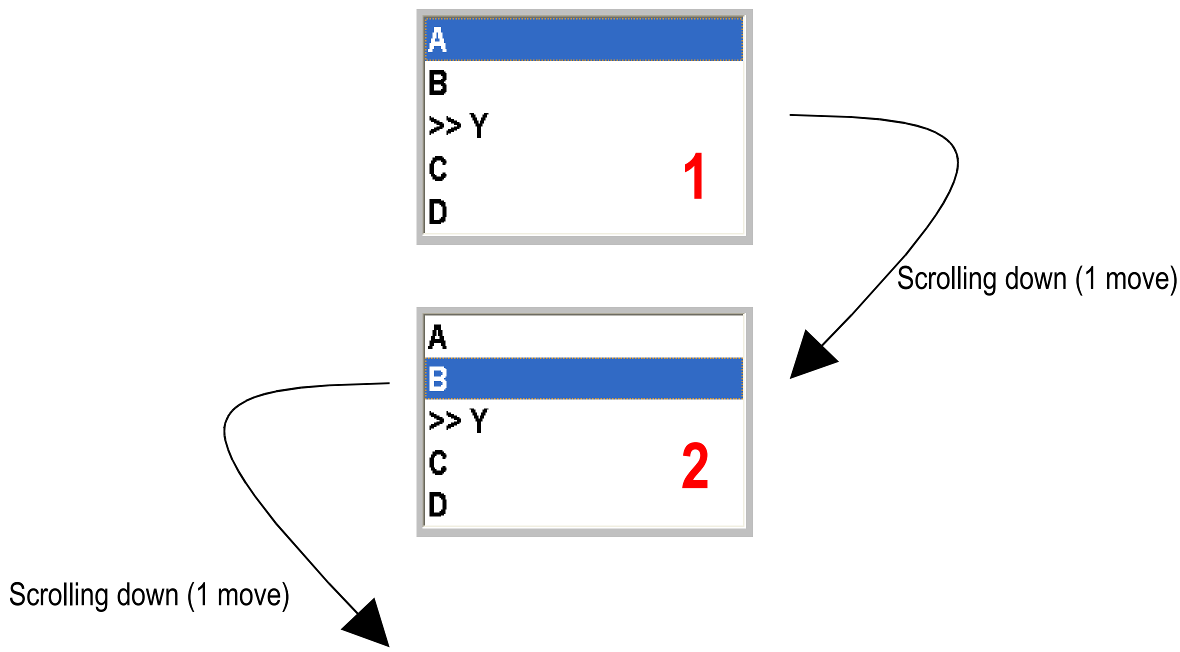
Physical representation (mental model)

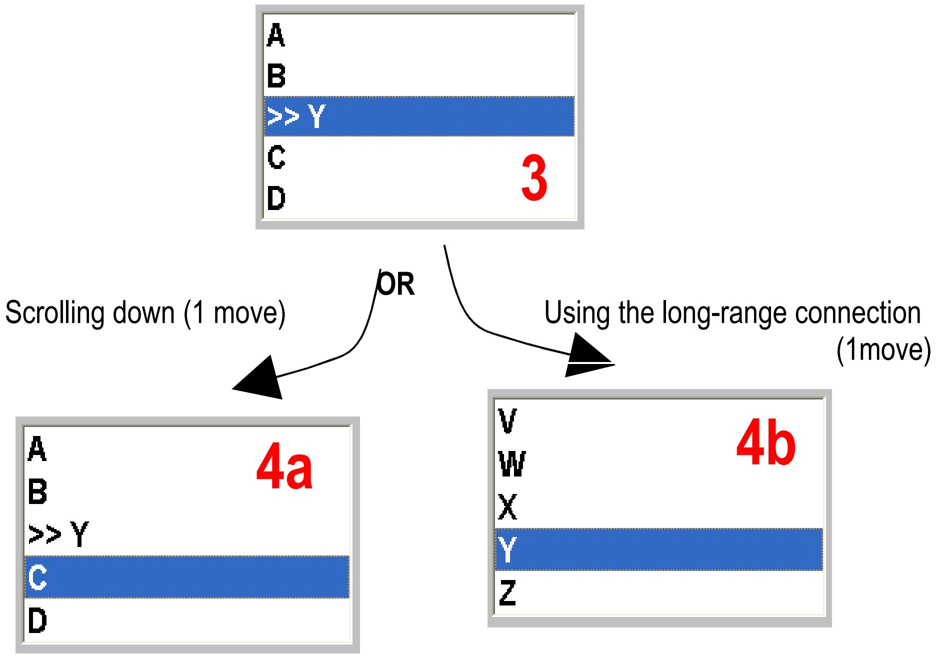
But as said before, the conceptual diagram does not account the ‘link’ as an entity:



Conceptual representation

Regarding the navigation, the course of the operations is simple to assimilate. If the special entity (the ‘link’) has the focus, three moves are possible: a back move will land on B, a move forward will set the focus on C and the ‘link’ move will take the selection to Y. What follows is a sequence of **4** possible moves illustrating the principle detailed above, displayed on a small screen.





3. Specifics

There is certain number of occurrences of the word "random" in the previous paragraphs. Indeed, the twofold randomness of the algorithm is an integral part of the theory (see section 4). On one side, the positions of the links have to be randomly set up. Conceptually, each regular element of the list is eligible to be assigned a link, but the system only generates shortcuts for a few of them. On the other hand, the destinations of these links are also the result of a randomization process, the sole restriction being that a link cannot link to itself. The rules governing the control of the randomizations will be fleshed out further in the report.

With regards to the notion of communication, we need to point out a couple of aspects. As on common and normal lists, a two-way path is provided between items. That means that the list can be browsed in any direction (upwards and downwards or leftwards and rightwards). Concerning the links, the communication between the connecting entry and the connected entry only operates in one way (link TO destination). At this stage of the evaluation, it has been arbitrary decided not to provide a 'cancel' or 'previous' option. Moreover, a regular entry offering the possibility to jump back to its original link would be conceptually wrong since it would create a another different kind of entity and would add complicatedness to the understanding of the system, rendering it less consistent.

The last important point to bear in mind deals with the controls. In a minimal configuration, the algorithm would ask for merely three hard keys: two for opposite direction moves and one for the use of a link.

4. The small world network algorithm

In order to develop the algorithm supporting this new system, we have to refer back to Watts and Strogatz work [1]. They devised the "rewiring" procedure that allows a regular network to be converted into a small world network one. In other words, it creates the set of random connections. Adapted to the list case, the procedure consists of several steps:

1: we start by considering the first element of the list.

2: with probability p, the element is assigned a shortcut

3: if the element has a shortcut, then we connect it to its destination element, chosen at random among all the elements in the list, with forbidden "self-linking" for sole condition,

4: we repeat the operations from step 1 for each element all the way to the end of the list.

The important point is to control p. If p=0, then the networks is regular as no connection have been set up. If p is too high, the level of randomness increases and the networks gets disordered. In both experiments, p has been adjusted to **0.1**.

The next chapter will extend the technical review of the system.

Reference

1. Watts D. and Strogatz S., (1998) "Collective dynamics of 'small-world' networks", *Letters to nature, volume 393, 440-441*

4. Development of the simulations

For the two experiments around which this thesis was composed, the participants had to perform activities on two PC applications which were run in the context of use of a mobile phone contact list and a mp3 player. This chapter is devoted to a brief technical review of those applications that I developed on my own using Microsoft Visual Basic 6.0.

1. The motivation

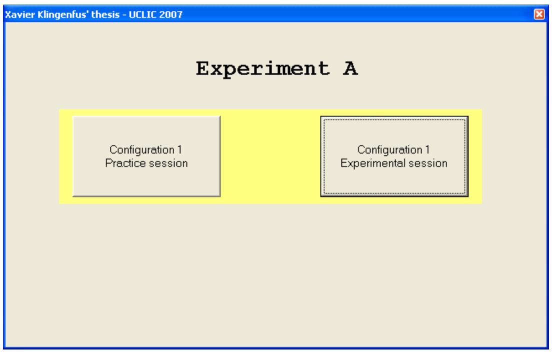
Testing a new navigation system asks for a lot resources, especially if it has to be implemented on handheld devices. As my budget was very tight for this project, I could not envisage creating physical prototypes of a mp3 player and a mobile phone embedding the system. Therefore I had to create my own environment and decided to lead the experiments on a computer. By carefully considering the controls and the layout of the interface, I tried to reproduce the experience of using those devices by designing two computer applications. Moreover the underlying algorithms enabled me to easily track the performances and store them on a “ready to use” log file.

I opted for Microsoft Visual Basic 6 as the development framework for two main reasons. First, programming was one of the principal majors of my undergraduate course, wherein I was taught the Visual Basic language among many others. Therefore I believe I am fairly skilled at scripting with it. The second reason is the ease of use of the language. Not only does it facilitate the design of user interfaces, but it also embeds a very clear and straightforward code structure, which makes the development pleasant and effortless.

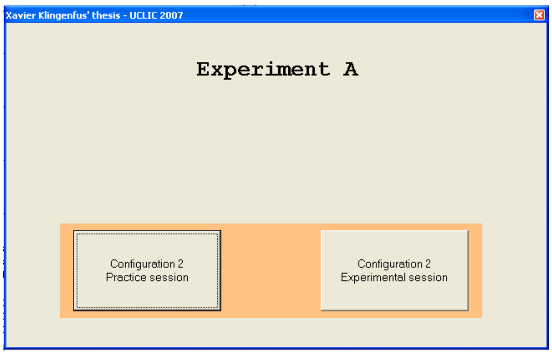
The two interfaces explained hereafter stage the use of a mobile phone contact list and a mp3 player. These devices, which are by now completely and widely adopted, integrate lists in their mode of operation. Therefore, rather than simulating a senseless list of letters, I preferred to set up a real environment to favour the understanding of the system and to prevent boredom among the participants.

2. The mobile phone contact list

In terms of the user interfaces, the applications stuck to a strict minimum not to influence nor distract the participants. Therefore simple screens were created. In the first experiment, the participants were asked to perform tasks on a mobile phone contact list. As the experiment was a between-subject procedure regarding the navigation system, the application had to comprise two different parts.



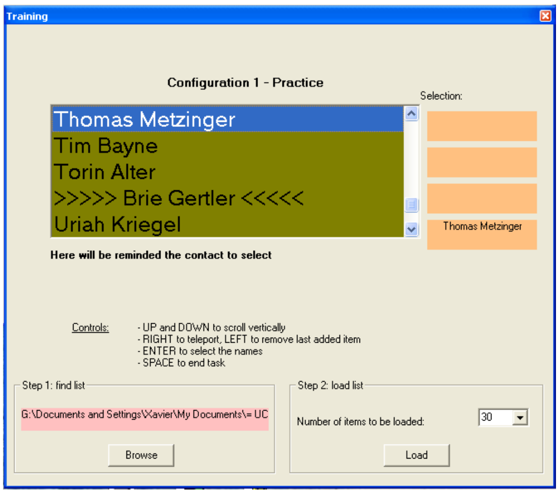
Start screen for the SN configuration



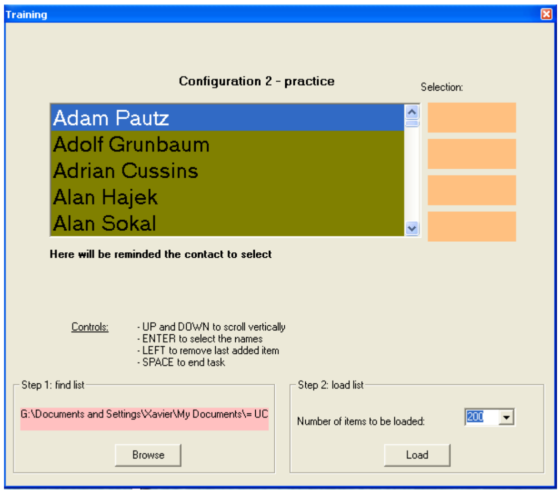
Start screen for classic configuration

The start screens displayed two buttons leading to the practice sessions and the experimental sessions. Hereafter are presented the two practice interfaces. They only differed in that for the SN configuration, the list of contacts included links (surrounded by “>> <<” to make them stand out) and allowed the use of a “jump” button. However, the course was the same on both systems: first the list had to be located (bottom-left frame), then it had to be loaded up while specifying the size (bottom-right frame). The participants could then manipulate the interface by scrolling the list, selecting names and taking shortcuts. In addition, the screens were only displaying 5 lines of entry as it is the usual layout for phone screens, according to an informal survey that I led among my acquaintances (20 respondents). With respects to the controls:

- UP and DOWN arrows to scroll vertically
- RIGHT to use the links (SN configuration)
- ENTER to stack up selected names in the oranges boxes
- LEFT to remove them
- SPACE to end the task (only working on the experimental procedure)

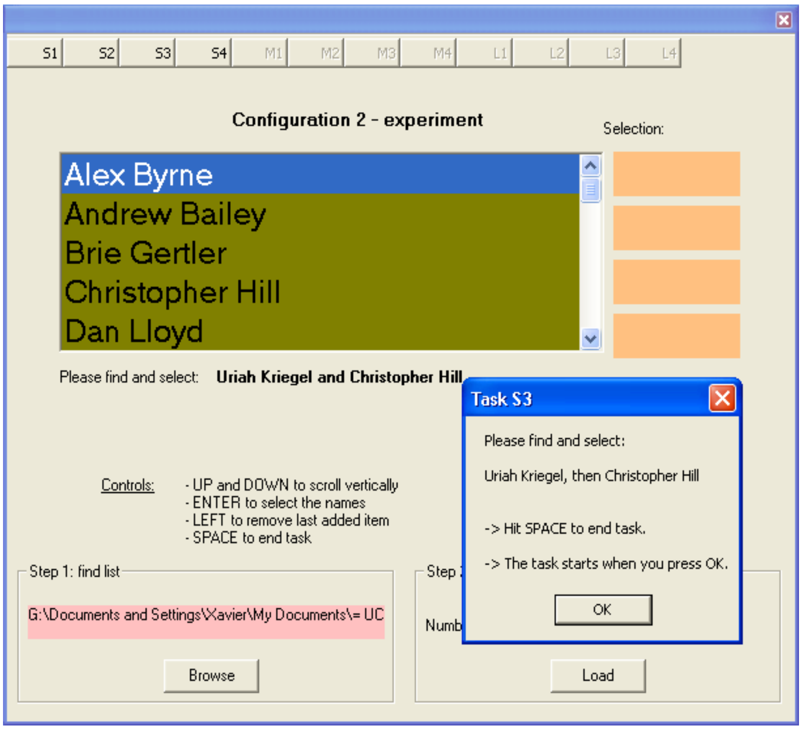


Practice screen for the Small Networks config.



Practice screen for the classic configuration

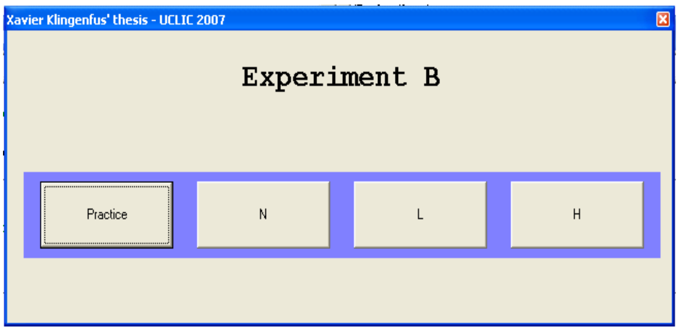
Once the practice session was over, closing the window led back to the first screen where access to the experimental sessions was given. The sole dissimilarity with the practice screens regarded the set of buttons on the top, which corresponded to the different tasks to perform. Once one of them was pressed, a message would pop up, summarizing the task to achieve. The counter started when OK was clicked and stopped when the participant hit SPACE. The task completion times were being tracked in the background and results were saved in an external file after all tasks were achieved, when closing the window.

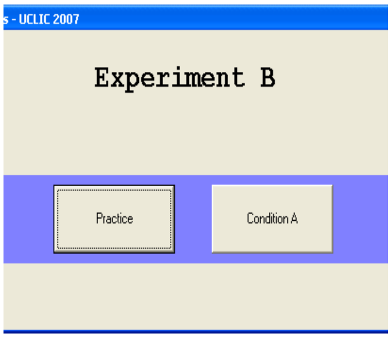


Experimental session for the classic configuration

3. The mp3 player

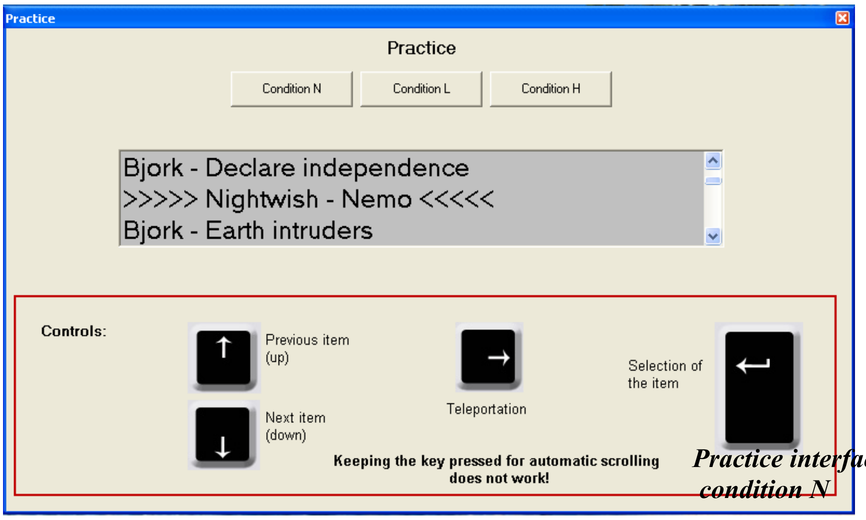
The second experiment was staging the use of a mp3 player, the kind of which that could also act as a memory stick (i.e. limited functions and controls, small display). This being a within-subject experiment implied the design of a unique interface for all participants. However the fourth and last conditions having been figured out quite late in the process was implemented on a different interface for a gain of convenience and time. This did not present a problem since both applications were run simultaneously and it did not interfere with the task randomization. The same principle as for the first experiment was applied here in that the first screen turned out to be a portal, giving access to a practice session and four experimental sessions. The first screen hereafter presents the portal for condition N, L, H and the practice session whereas the second one only relates to condition A.





Main application (3 conditions) *Second application (4th condition)*

The practice session intended to get the participant familiarised with the navigation types and the controls, as for the first experiment.



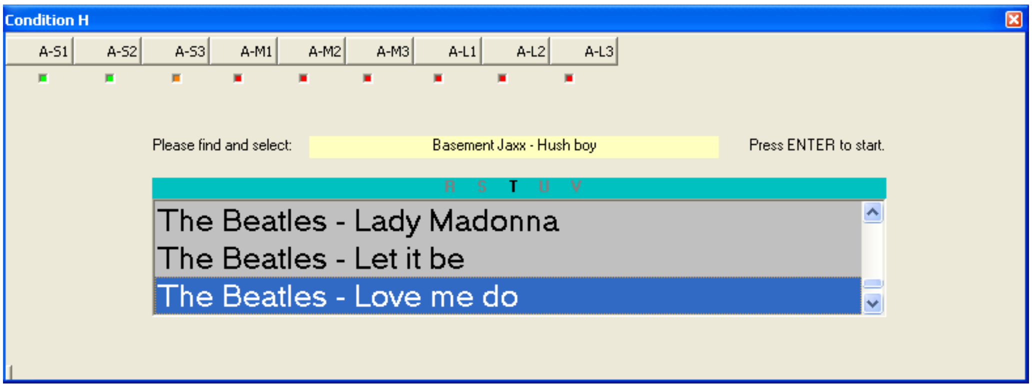
Practice interface for condition N

Regarding the latter, the two SN interface had the same configuration as for the first experiment, except for the ENTER key, the function of which replaced SPACE (which was deactivated). This was in an attempt to reduce the set of controls to be memorized and to smoothen the navigation. Two other competing shortcut systems were also added, thus two more algorithms to develop.

The first one, called ‘Hitlist’, was a basic “Page up / page down” system, resembling highly to the implementation on a traditional PC. In sum, the scrolling keys remained unchanged whereas the Page up and Page down moves were achieved through hits on LEFT arrow and RIGHT arrow.

The second, ‘Alpha’, consisted of an alphabetical horizontal scrolling, inspired by the one on the Sony Walkman AW3000. The vertical scrolling is still operated the same way, whereas LEFT arrow and RIGHT arrow enable a horizontal scrolling through the alphabet. Screenshots highlighting the dissimilarities within the four different navigation systems are provided in the chapter 6.

The experimental interfaces slightly differed from the previous experiment. The buttons composing the taskbar (here literally the bar of the tasks) were automatically adjusting the proper list path and size. Below each button, a little light indicated the status of the corresponding task: green for done, orange for being processed and red for undone.



Experimental interface for the condition ALPHA. Task A-S3 is being processed whereas tasks A-S1 and A-S2 have already been performed.

4. Design justifications

User interface

As a student more specialised in programming than in graphical design, I focussed my efforts on framing elements in a clear and comprehensible way on the interface rather than in an artistic fashion. I attempted to support memory as much as it was possible: since participants were not familiar with the list contents, names and song entries to be found should not have been memorized and were therefore appearing in an obvious form. Moreover, fields that required a particular attention stood out by a distinctive colour, in order to highlight their content. However, the colour palette has not really been studied. The only consideration taken into account was the selection colour, which had to be blue as on most of the current application and systems. Performing the tasks by focussing diligently on the lists was the only thing that mattered for participants.

Controls

Here is a table summarizing the controls for each experiment:

1st experiment: the mobile phone contact list

KEY	**FUNCTION**
UP/DOWN arrows	Vertical scrolling
ENTER (Return)	Stacking elements in the selection boxes
LEFT arrow	Removing elements
RIGHT arrow	Jump (SN configuration only)
SPACEBAR	End of task (timer of)

2nd experiment: the mp3 player

KEY	FUNCTION
UP/DOWN arrows	Vertical scrolling
ENTER	End of task
RIGHT arrow	Jump (SN), horizontal scrolling (A), Page down (H)
LEFT arrow	Horizontal scrolling (H), Page Up (H)

The rationale for assigning the 'Jump' function to the right arrow stems from the fact I considered this feature as a navigation control: when positioned on a link, the participants could either go up, down or take a jump, this third move being part of the navigation (contrary to ENTER or SPACEBAR, which have task-related functions). Therefore, I believe that by setting up the controls this way, a fast navigation would be encouraged and facilitated. Moreover, cycling through the lists (jumping from the first to the last item) has not been authorised since most of the current devices to not allow this practice.

Limitations

Fast scrolling

The principal drawback in these applications was the lack of fast scrolling system. That means that the participants had to hit the keys as many times as needed to scroll (vertically or horizontally), i.e. they could not keep the buttons pressed to browse the list. This is mainly due to the complicatedness of regulating the pace. It cannot be set up as a property; therefore an algorithm would have had to be sought. As there are many parameters to take into considerations (for instance, the "progressive pace") and as time is a valuable resource that I could not afford to waste, I decided to block the fast scrolling (the default speed being too high for adequate scrolling). This applied for all conditions. Moreover, an informal survey among acquaintances of mine (20 respondents) revealed that 90% of the respondents never keep the arrow keys pressed when scrolling.

5. The algorithms

Some of the algorithms implemented here raise interest. Although I tried to transfer Paul's original Java code into Visual Basic, it was not an easy task. Here are showed the most hassling parts of the code I have to develop.

Class 'myItem'

To properly manipulate the different elements of the list, I resorted to a Collection (similar to an array) of instances from the class 'myItem' that I devised by myself. Doing so avoided me to set up a 2-dimensions array, which is in the same occasion not as flexible as a collection. Typically, a 'myItem' object has 4 attributes and 8 procedures, each of which allowing to get or set a value for each of the attributes. This item has a name, an id, a potential shortcut (name) and the id of the potential shortcut.

In brief, an element of the collection corresponds to an item in the list. Hereafter is a screenshot of this class.

```
(General)                                   (Declarations)

Private id As Integer                  'id of the item
Private name As String                 'name of item (ex: andrew bailey)
Private shortcutid As Integer          'id of the shortcut
Private shortcutname As String         'name of the shortcut


Public Property Get getId()
    getId = id
End Property

Public Property Let setId(x As Integer)
    id = x
End Property

Public Property Get getName()
    getName = name
End Property

Public Property Let setName(x As String)
    name = x
End Property
Public Property Get getSId()
    getSId = shortcutid
End Property

Public Property Let setSId(x As Integer)
    shortcutid = x
End Property

Public Property Get getSName()
    getSName = shortcutname
End Property

Public Property Let setSName(x As String)
    shortcutname = x
End Property
```

Randomization

The following screenshots are self-explanatory (comments in the code) and describe how the double randomization is operated. The procedure emphasized here is in charge of creating the shortcuts within the collection of elements, before the list is put at display (“rewiring procedure”).

```
Private Sub Command3_Click()        '#### WIRING
Call clearlist
'##################################################
'# - listname is a collection of myItem objects   #
'# - a myItem object contain: an ID               #
'#                 a NAME                         #
'#                 a SHORTCUT_ID                  #
'#                 a SHORTCUT_NAME                #
'##################################################
Dim i, y As Integer
'#############################
Dim seed As Long            '#
Dim rewire_frac As Double   '#
                            '##########################################################
seed = 1            ' only used when same randomization for each occurrence is wanted #
rewire_frac = 0.1           '##########################################################
'#############################
```

```
Dim rnd1 As Double  'used to randomly select an element (a position) which a shortcut will be assigned to
Dim rnd2 As Integer 'used to randomly select the destination of the shortcut
'#### wiring ####

Randomize       'important for random sequences, without it the same random numbers appear.
    '### random shortcut assignment
    For i = 1 To listName.Count

        rnd1 = Rnd

        If rnd1 < rewire_frac Then
         rnd2 = Int(Rnd * listlength) + 1                            '### between 1 and collection.count
         listName.Item(i).setSName = listName.Item(rnd2).getName     '### name taken from list1
         listName.Item(i).setSId = rnd2                              '### id of name in the COLLECTION

        End If
        ' if rnd1 is below rewire_frace, then the algorithm assigns a shortcut to the item i in the list.
        ' rnd2 indicates the random destination element.
    Next
```

Important note: the list was rewired every time its size changed, for both experiments. Therefore the set of links differed across the lists and was unique for a given list and a given size.

5. Experiment A – Proving the added-value

After having exposed the small world network theory and its technical application, time has come to eventually focus on the fundamental purpose of the thesis: evaluating the system. This chapter describes and give the results of the first experiment carried out in this study.

1. Description

Aim

The goal of this first experiment is to test the added value of the small world based shortcut system. In order to do so, and as no work has been done so far on such a topic, the first step of the testing intends to detect whether or not the system supposed to provide shortcuts does effectively allow a quicker access to elements in a list. Consequently, this experiment aims at comparing performances regarding particular tasks on two different interfaces: a classic one with no particular quick access to the items (scrolling only) and the SN interface, which will implement the shortcut technique derived from the "small world networks" paradigm. This first experiment was conducted in the context of use of a mobile phone, i.e. a device which many are familiar with and use it regularly. Moreover, a mobile phone involves a list interaction for most its primary feature (making calls, sending texts).

As formulated by its designer Paul Cairns, this new system should turn out to be particularly effective when browsing large lists. The null hypothesis is therefore that this shortcut system has no impact on the times needed to access items in lists and it does not bring any added value in terms of usability and task completion time regarding a no-shortcut interface.

Objectives

The objectives leading the experiment are double:

- first, with respects to the technology itself, the principal measurement sought is the task completion time, i.e. how fast a user goes when performing particular tasks
- then, the study also attempts to find out about the user experience peculiar to both interfaces and gather feedback in terms of ease of use, learning processes and user satisfaction

Course of the operations

The experiment was a four step procedure.

- The participants were first asked to fill a consent form which explained them the overall goal of the study, what their commitments were and what rights they had.
- The experiment started off with a demographic questionnaire which intended to figure out about participants' basic demographic information and mobile phone usage (especially regarding their familiarity with the tasks to perform).
- Once this document filled out, they began to manipulate the interfaces. In both conditions, the experimental procedures were preceded by a short practice session.
- To conclude the experiment, the participants were asked to complete a slightly modified version of the System Usability Scale questionnaire. Irrelevant questions were removed whereas an extra section peculiar to the experiment was added.

All these documents are reported in the appendix.

Design

In order to test whether the SN shortcut system reduces the task completion times, two interfaces were therefore created: the classic one (**C**) only allowed the users to scroll vertically whereas the **SN** configuration embedded the shortcut system. But what does also matter in this context is the number of items contained in the list (i.e. the size of the list) in that it supposedly impacts the use a of a quick access technique. An informal survey of 20 respondents revealed that people (mostly students) have 110 contacts in their phone. The least number of entries turned out to be 30 whereas the largest contact list contained up to 200 names. In consequence, three list sizes have been considered: short (**S**, 30 items), medium (**M**, 110 items) and large (**L**, 200 items). To summarize, participants were asked to perform tasks on two differences interfaces with three different sizes. This gave rise to six experimental conditions:

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="3">List length</td></tr>
<tr><td>Short</td><td>Medium</td><td>Large</td></tr>
<tr><td rowspan="2">Navigation</td><td>Small networks</td><td>SN-S</td><td>SN-M</td><td>SN-L</td></tr>
<tr><td>Classic</td><td>C-S</td><td>C-M</td><td>C-L</td></tr>
</table>

As for the dependent variables, the task completion time and the usability feedback were already mentioned earlier. To a minor consideration, the experiment also logged the number of shortcuts used to achieve a particular tasks under the SN configuration and the selection of the names for each task as well (a task basically consisted of selection names). The two latter variables were recorded as tools for observing on how participants were navigating through the systems rather than considered as decisive factors of primary concern.

To summarize, there were two independent variables,

- the type of interface (two levels: "small networks" and "classic")
- the list length (three levels: "short", "medium-sized" and "large")

and four dependent variables,

- the task completion time, as the primary measurement
- the number of shortcut used (for SN configuration) for complementary information, as well as
- the selection of names (to check accuracy) and
- user feedback through the SUS questionnaire.

Tasks

The study was conducted as a mixed measures experiment: between subject for the interfaces but within subject for the different list sizes. Each participant was then asked to perform a series of 12 tasks, 4 per list length, on a sole interface. For each of the 6 conditions described above, the tasks were similar across interfaces and in a controlled order. The tasks were named according to the list length they referred to:

- 4 tasks for the short lists: S1, S2, S3, S4 (identical across the two interfaces)
- 4 tasks for the medium-sized lists: M1, M2, M3, M4
- 4 tasks for the large lists: L1, L2, L3, L4

The tasks mainly consisted of browsing the lists and selecting the names defined in the task instructions. The full tasks description is reported in the appendix.

In order to control the sequences of the tasks and avoid learning effects, the following Latin square was devised for both interfaces:

Participant	**1st task**	**2nd task**	**3rd task**	**4th task**
i	S1	S2	S3	S4
ii	S3	S1	S4	S2
iii	S2	S3	S1	S4
iv	L1	L2	L3	L4
v	L3	L1	L4	L2
vi	L2	L3	L1	L4
vii	M1	M2	M3	M4
viii	M3	M1	M4	M2

Participant	**5th task**	**6th task**	**7th task**	**8th task**
i	M1	M2	M3	M4
ii	M4	M2	M3	M1
iii	M3	M1	M4	M2
iv	M1	M2	M3	M4
v	M4	M2	M3	M1
vi	M3	M1	M4	M2
vii	L1	L2	L3	L4

viii	L2	L3	L1	L4

Participant	**9th task**	**10th task**	**11th task**	**12th task**
i	L1	L2	L3	L4
ii	L2	L3	L1	L4
iii	L3	L1	L4	L2
iv	S1	S2	S3	S4
v	S2	S3	S1	S4
vi	S3	S1	S4	S2
vii	S1	S2	S3	S4
viii	S3	S1	S4	S2

Material

For this study, the application simulating the contact list interface had to be developed. This has been done using Microsoft Visual Basic 6 (refer to previous chapter). The usage of the keyboard was somehow restricted to the participants. Only the following keys could be accessed:

- UP and DOWN arrow to scroll vertically
- ENTER (RETURN) to stack names in the selection area
- LEFT arrow to remove the latest selected name
- SPACE to terminate the task
- RIGHT arrow (SN configuration only) to shortcut.

The list equally had some specifics:

- It was arranged by alphabetical order, by first name.
- The fast scrolling was deactivated.
- The last element could not be reached through a backward move from the first element and vice versa.

Procedure

All the participants were individually tested in the usability lab 2 in Remax House, UCLIC department. They were first explained that they would take part in an experiment intending to compare two different interfaces allowing browsing lists and that their performances on one of this system would be recorded. They were also introduced with the context and told that the system simulated different mobile phone contact lists. They would then be asked to perform tasks on different list lengths. Likewise, precisions were given on the series of tasks, namely that the experiment would comprise 12 tasks, the goal of which would consist of browsing the lists and selecting name(s), reproducing then real life situations such as making calls or sending SMS. They were also informed that the sought variable was the task completion time but the emphasis was put on the fact that, although they should attempt to be as quick and accurate as possible, they had to endeavour to behave as natural as possible.

After having filled out the consent form and the questionnaire wherein they were free to ask questions, the participants carried out a short practice session in order to familiarize themselves with their respective interface and the corresponding controls. Throughout this training session, explanations related to the keys, the characteristics of the lists and the types of tasks were given on the fly and questions were encouraged at any time. A brief note was also exposed on the non-obligation to select the names in the order they were displayed by the instructions. In the SN configuration, a particular attention was drawn on the shortcut system and its mode of operation (teleportation). It was also indicated that the use of the link was not mandatory at all and that it was completely up to the participant to decide whether s/he wants to use a link or not. To check that participants had assimilated the controls and the navigation, the practice sessions ended up by a short informal "test-task" administered as follows: "Could you show me how you would select X from this position in the list?"

Subsequently, the experimental session could start. Once the proper screen set up and the correct list loaded, the tasks occurred successively, one at a time, with a short break between each of them. Every four tasks, the list was reloaded with the right amount of items. The course for a typical task resembled the following process.

- The experimenter (myself) pressed the button corresponding to the task to be performed.
- This action displayed a popup window unveiling the instructions related to the task and an OK button.
- The task and the counter began after OK was pressed. Although the counter started off, the elapsed time was not consultable when the task was on; only the task completion time was, by the end of the task.
- To support memory and minimise recall activities, the name(s) to be found and selected were standing under the list in an obvious form (bold characters).
- Once the task had started, the participant could browse the list and accomplish his mission.
- When a name was selected, it would appear on the right of the list to confirm the selection.
- Once the task over (SPACE got pressed), a label indicating the final time would pop up. Records were automatically tracked into a log file. From that point, the next task was ready to begin.

Once the twelve tasks were achieved, the participants were asked to fill out a customized version of the SUS questionnaire. Then, a short debriefing talk was engaged, first to flesh out the purpose of the study but also to elicit participant's opinions and suggestions.

Pilot study

A short pilot study was conducted prior to the experiment with one participant (who was not allowed to take part in the experimental procedure) to principally find out about the controls and the best way to arrange them. .

Ethical considerations

The procedure governing the experiment has been approved by the Psychology Department's Ethics Committee through Dr. Essi Viding. Moreover, by signing the Informed Consent Form, the participants are aware of:

- the possibility to withdraw at any time and to ask for a deletion of their records
- the anonymity of the data stored
- the use of their data under the Data Protection Act 1998.

Ethical approval number: MSHCI/2007/008

2. Results

9 females and 7 males were recruited for this experiment. Mostly aged between 19 and 29 (94%), they were all experienced mobile phone users, having all possessed one for more than 2 years. In addition a large range of the participants uses their device on a daily basis which makes them familiar with the navigation system. However the survey showed that 56% of them access contact by using the “first letter” keyboard shortcut whereas 38% browse short lists (received calls, calls made…). In sum, the use of shortcuts and browsing short lists are commonplace.

User performances

The following table summarizes the timings obtained for each of the tasks per interfaces:

	Classic (8 pa.)		**Small Networks (8 pa.)**		
	Average time in sec.	Likelihood to select by alpha order	Average time in sec.	Likelihood to select by alpha order	MODE n. of shortcuts used
Short 1	**6.5**	N/a	7.31	N/a	1
S 2	**5.3**	N/a	6.4	N/a	0
S 3	**8.58**	87%	9.45	100%	1
S 4	**19.2** **6.3**	75%	24.3	37%	1
Medium 1	21	N/a	**16.4**	N/a	1
M 2	**11.6**	N/a	15.5 6.28	N/a	1
M 3	28.1 9.77	87%	**22.7**	100%	1
M 4	**30.5**	87%	47.4	37%	0
Large 1	32.8	N/a	**17.6**	N/a	2
L 2	**15.6**	N/a	18.3 10	N/a	1

L – 3	41 13.4	75%	**23.5**	100%	1
L – 4	**53.9** **14.2**	75%	55.1	75%	2

In bold: winning times (C vs SN for each task)
In blue: standard deviation in seconds (provided when the mean is not representative)
N/a: Not applicable

A glimpse of this table drags the attention on a couple of points:

- For short lists, performances are better without the shortcut system (the task completion times being lower)
- For medium-sized lists and large lists, the highest performances are evenly shared between the classic and the SN interface.
- For large lists however, the performances that are higher on the classic interface win with a short margin whereas performances greater on the SN interface have a more substantial margin.

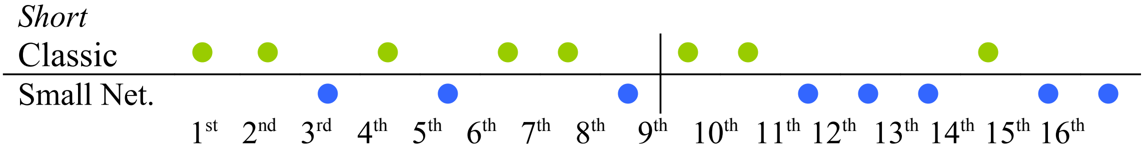
Let us now look closer at the set of data gathered. The following Mann-Whitney test examines the ranks of each participant according to their performances on each list.

Interface		**N**	**Mean rank**	**Sum of ranks**
Small	1	8	10.38	83.00
	2	8	6.63	53.00
	total	16		
Medium	1	8	8.75	70.00
	2	8	8.25	66.00
	total	16		
Large	1	8	6.13	49.00
	2	8	10.88	87.00
	total	16		

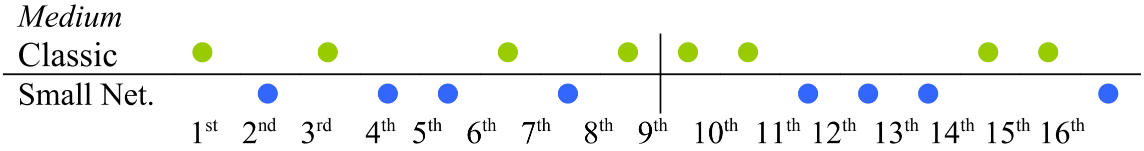
1: SN interface, 2: Classic interface

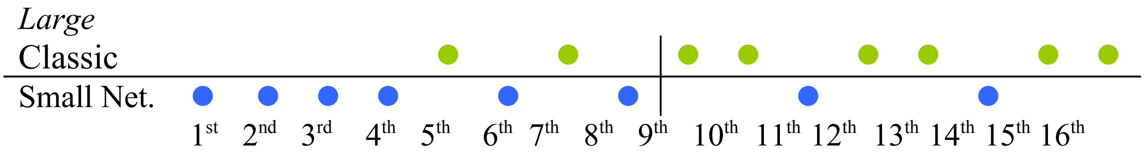
Apart from the medium-sized list case, where there is no significant difference, there is an important interval between the average ranks of both interfaces on the small and the large lists. This can be illustrated by the following chart which can be seen as a graphical version of the above table:

Short



Medium





The colour weightings are quite expressive under this form. For the short list, the greatest spread of colour towards the highest ranks being green clearly indicates that the best performances have been achieved on the classic interface. Conversely, for the large list, the blue being the colour that occupies the highest ranks shows that the participants were the most performing on the SN interface. In other words, the SN system turns out to be much more effective in terms of task completion time on large lists.

Moreover, let us consider the following Test Statistics table:

	Small	**Medium**	**Large**
Mann-Whitney U	17.000	30.000	23.000
Wilcoxon U	53.000	66.000	49.000
Z	-1.575	-.210	-1.995
Asym. Sig. (2-tailed)	.115	.834	**.046**
Exact Sig. [2*(1-tailed Sig.)]	.130	.878	.050

The probability of having such a pattern (with respects to the ranking) being lower than 0.05 means that this event is relatively unlikely to happen. As this was predicted, the results are therefore very significant and prove that the outcomes are not due to chance.

Usability feedback

Classic configuration
The participants seemed to be really annoyed by the lack of shortcut system (they were all familiar with at least one on their device) and wouldn't like to use this interface which they judge not efficient enough. However the system is fairly easy to handle and understand and the learning process is quite straightforward.

Regarding the speed with which the tasks were performed, the results appear to be conform to the expectations in that the longer the list gets, the more unsatisfactory the pace becomes. In other words, not having any shortcut system does not globally bother the participants on short lists but significantly impacts their performances as soon as the lists grow up.

SN configuration
This system did not turn out to have a great success since half of the users would not like to use it regularly. This could be explained by the fact that they are already familiar with a system showing a greater efficacy. Nevertheless, despite a certain lack of efficiency, the participants expressed a feeling of confidence using the interface,

not finding it particularly complex. In the same way, assimilating the system did not emerge to be a major step as the process is fairly intuitive once adopted.

The major drawback reported by the participants was the link (shortcut) randomization that occurred for every change of list size. This observation is at some point surprising since the whole content of the lists changed as well, impacting therefore the usefulness and the relevance of the shortcuts. This can probably be seen as an attempt to express a desire of stability, i.e. a randomization of the shortcuts should not occur every time the mobile phone is switched on in order to provide a better control and mental model of the system.

In terms of task completion time, the added value of this system is highlighted when comparing performances on large lists: the satisfaction is much greater on this interface than it was for the classic one. It also emerged that for short lists, participants using this interface were not as satisfied as those using the classic system, still with respects to speed.

General observations

Natural differences
Anonymous and discrete observation of the way the participants behaved revealed that some aspects peculiar to each of them affected the study. The first consideration deals with how they use the controls: some of the participants utilise only one hand whereas others prefer to use both. When they operate with both hands, one is generally devoted to the navigation whereas the other one is used either to hit RETURN or SPACE. These dissimilarities actually alter the overall speed of the tasks. Moreover, some participants type faster than others: either slowly and confidently, nervously and eager to be the quickest possible, or any nuances in between these. Discrepancies can also be noted in the way people think and treat information. On one hand the completion times can be influenced by the tendency to heavily rely on the reminders or to memorize the sequence of names displayed during the tasks instruction. On the other hand, some participants will adopt strategies (selection of names by alphabetical order) whereas other will not. In the case of the SN configuration, strategies ask for more important cognitive effort which widens even more the range of natural differences in terms of mental abilities.

Use of shortcuts
As such an algorithm had never been implemented for a similar purpose, it was interesting to detect how the participants would react and especially act when confronted to this system. A couple of noteworthy observations was then pointed out. The first one relates to the tendency to use the shortcuts. It was detected that the closer to the goal participants get, the less they tend to consider using shortcuts. This can be explained by the fact that they judge the decision making time higher than the time it would supposedly take to reach the desired item: using shortcuts becomes then not worth. In this case, the decision making process consists of determining whether, first, the teleportation will bring the selection closer to the name to select (reducing the distance to desired item, done by alphabetical comparison) and then whether the expected gain of time makes it worth using it versus the needed time to scroll to the desired element. In the same way, the observations showed that most of the

participants were seeking for long distance teleportations rather than doing small jumps. This appears to be surprising as well, since in an ideal world, one could think than any gain of time, even very little, would be valuable. But this is a double edge action: one can gain time through a short leap but one can also miss a long range connection that would positioned after the short range connection, therefore not displayed at the moment… The observations also singled out one major limitation of the experiment: the teleportation origin and the target were not recorded, providing thus no statistics about the use of numerous shortcuts. From what could be observed, an exhaustive use of shortcuts for one particular task could mean a better performance (smart use) as well as it could lead to higher completion time due to the confusion it provoked.

3. Conclusion

This first experiment was devised to test a new shortcut system for browsing lists. As no work had been done on this beforehand, it was necessary to conduct a study aiming at gathering first performances on specific tasks but also user feedback. This has been achieved in the context of use of a mobile phone; this device is one of the most common systems providing the users with a list interaction and has been widely adopted by millions. The true and fundamental goal of this experiment was in fact to show the added value of this system. By definition, an interface implementing a quick access method to any elements in a list should show significant results in terms of task completion times. Therefore, an experiment intending to measure performances on a no-shortcut system and the SN system was carried out in order to highlight the added value of this new interface.

The outcomes of this first study were multiple. Firstly, the idea of having shortcut techniques present in lists was reinforced. The participants, who were quite familiar with existing methods on mobile phones, clearly manifested signs of annoyance when using the classic interface. Second, the comparison of the performances showed an interesting point: the larger the list gets, the more efficient the SN system becomes, which invalidates the original hypothesis. By that, it is meant that this new shortcut system is particularly useful long lists. Also, the interface surprisingly turned out to be easily assimilated.

However, participants voiced their reluctance about using the system. The reason is simple and has briefly been introduced in the previous paragraph: people are already familiar with existing shortcut systems, and in the case of a mobile phone contact list, those systems are more performing. It would not be necessary to compare the SN interface with the traditional input of the first letter of a name to reach the proper entry in the list: the latter system would easily win the contest. And this is how people operate on their phone. As a matter of fact, the mobile phones own a secret weapon: they embed a keyboard. In sum, the first lesson that can be drawn from this experiment is that this SN system cannot make a difference on a device incorporating a keyboard, because “keyboard shortcuts” would prevail over the rest. This is particularly true when it comes to selecting string items (chains of characters).

Therefore, let us consider the problem under another angle. It is now a fact, that this new concept is pretty convenient and useful for large lists of items but that it cannot challenge keyboard-based shortcut systems. It could consequently be interesting to challenge this system with other shortcut techniques not involving the use of a keyboard. This must not sound like a desperate attempt to find a utility to this system! As an example, mp3 players are a wide domain of application for this kind of technology. Apart from highly branded devices such as the Ipod, there are tones of mp3 players with smaller capacities that can generally also be used as storage systems (for instance, memory sticks). Currently those appliances have basic scrolling systems. It can then be imagined that if the SN would appear to be competitive for this market, it could be a breakthrough in the navigation for this industry… But to do so, we need to confront it to its potential competitors. This is the aim of the second experiment.

6. Experiment B – Comparison of systems

After having showed evidence of the added-value brought by this novel system, we need to see if it can compete with the existing solutions. Here is presented all the work done related to the second experiment.

1. Description

Aim

To be really concise, the latter experiment taught us one thing: the SN system seems to be fairly effective in increasing performances on large lists, which demonstrates signs of its added value. It can therefore be considered as a *shortcut system* worthy of this name. But it is not the only system favouring fast navigation: we are all familiar with the inputs of the first letter to find a name in our phone contact list or with the *Page Up* and *Page Down* keys on our computer keyboard, and so on… hence the desire to challenge this novel design with existing systems with the aim of comparing performances.

But the outcomes of the first experiment questionnaires raised an important matter as well: the system would not be suitable on devices incorporating a proper keyboard: indeed, there is no need of an experiment to prove that traditional keyboard-based shortcut methods would be more efficient and more easily learnt. From another point of view, this means that it could be more worth exploring its application on devices with restricted controls. This is why this second experiment framed the use of a mp3, especially targeting the models that also act as memory stick. This kind of appliances generally only counts a multi-directional key and a validation button, the whole with a limited screen.

Illustration of "mp3 player – memory stick"

To a minor level of importance, feedback from the users who took part in the first experiment and observations revealed that having an entire sequence of characters (as the shortcut) can be quite a load to process. To check whether this workload can be diminished, one of the SN interface challengers will also be one of its alternatives, with the names being replaced by a sole letter with regards to the links.

The experiment will then make participants perform tasks on four different interfaces:

- SN with names as the shortcuts in the list
- SN with letters
- “Hitlist”, which consists of a Page Up/Page Down system
- “Alpha”, which enables an alphabetical scrolling.

The null hypotheses are double:

- The performances on the SN systems do not outrun the other shortcut systems
- Both in terms of performances and usability, the SN version with letters should not prevail on the one with the names

Objectives

There are two main objectives:

- Measuring the task completion times on the four interfaces to establish a ranking in terms of performances
- Gathering feedback with respects to usability, in order to rank as well the interfaces according to the participants’ preferences.

Course of the operations

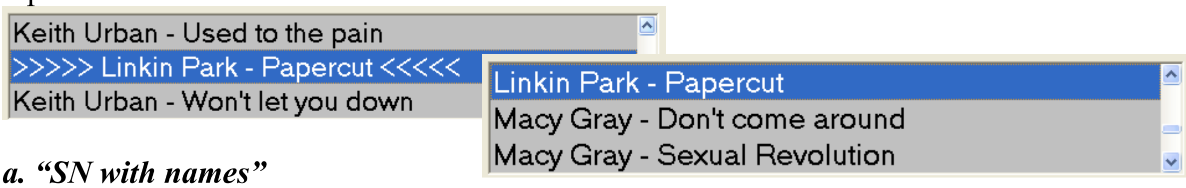
The experiment was conducted on a similar basis as the first one.

- The participants were first asked to fill a consent form.
- A demographic questionnaire was then distributed, the goal of which was to figure out about participants’ basic demographic information and their experience with mp3 players.
- Once this document filled out, the experimental procedure could start, after a practice session.
- An altered version of the System Usability Questionnaire was handed round to close the session with the participants.

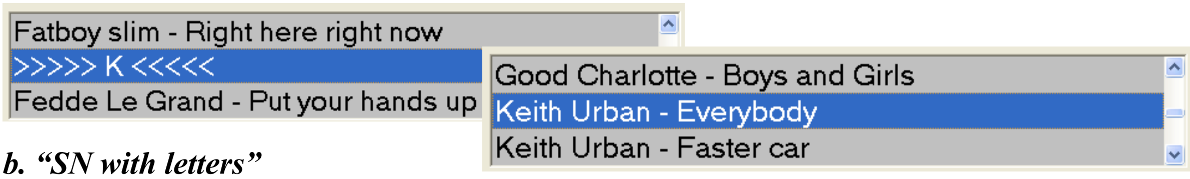
All these documents are reported as appendices.

Design

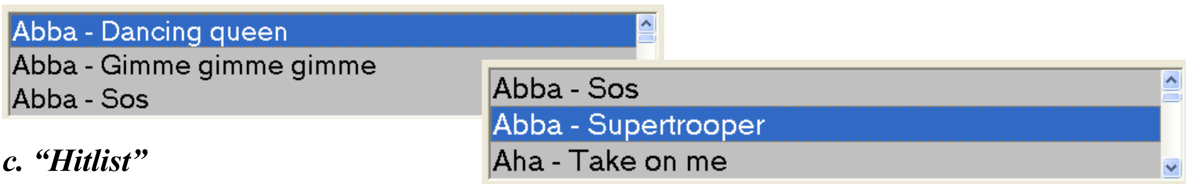
In the aim of comparing the previously tested **SN** system with a variation of itself and commonly adopted shortcut systems, four interfaces were created. Similar to the first experiment, the controlled test, called **N**, was implementing the SN shortcut system having names as the links. The second interface emerged from a variation of the original version of the latter: names were replaced by letters, a click on which leading the participants to the first entry beginning with the specified letter – this interface has been named **L**. The third interface was labelled “Hit list” – **H** – and consisted of a fast navigation system. The particularity of this system is that the main controls generally enable the users to browse not only from elements to elements (traditional scrolling) but also from pages to pages within the list (three by three as there are three lines of display). The fourth condition – **A** – permitted the participant to scroll alphabetically through the list. The screen only displayed three lines of text, as on many of current mp3 of that kind. Here are some screenshots of the four interfaces:



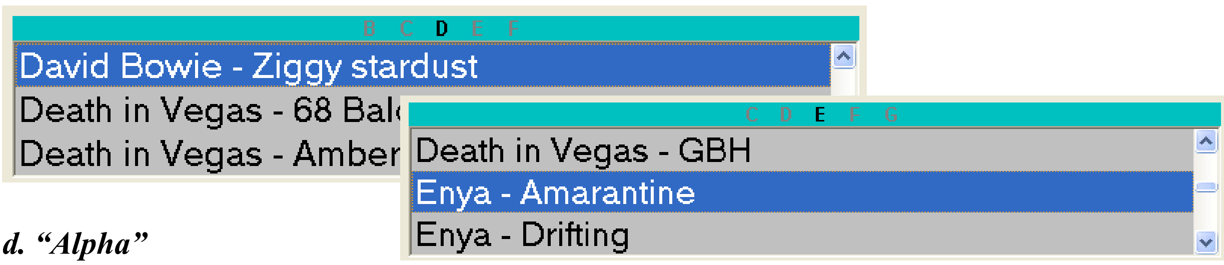
a. "SN with names"
By pressing *right arrow* on a link, the user gets to the corresponding entry in the list.



b. "SN with letters"
By pressing *right arrow* on a link, the user lands on the first entry starting with the selected letter.



c. "Hitlist"
By pressing *right arrow* on a link, the user lands on the first item of the next page downward. *Left arrow* does the opposite move.



d. "Alpha"
By pressing *right arrow*, the next latter in the alphabet takes the focus and the first name starting with it is selected. *Left arrow* does the opposite move.

Another facet analogous to the first experiment is the size of the playlists. Although it has been proven that the SN based algorithm was efficient on large lists, it is still necessary to run the test on different list length for the simple reason that nowadays, mp3 players are differentiated on their storage capacities. This means that memory sticks acting also acting as mp3 players are generally classified in three categories: 256Mo, 512Mo, 1Go. Moreover, as the optimal list size has not yet been determined, the length of the lists still impacts the performances and is still under consideration. As we are interested in large lists, we can therefore set the number of items in the lists as the maximum number of songs of a standard format (4Mo), as follows:

Length of playlist	Number of songs (items)
Short (256Mo)	60
Medium (512Mo)	120
Large (1Go)	240

This gave rise to 12 experimental conditions:

		Playlist length		
		Short	Medium	Large
Type of interface	"SN" with names	*NS*	*NM*	*NL*
	"SN" with letters	*LS*	*LM*	*LL*
	Hit list	*HS*	*HM*	*HL*
	Alpha	*AS*	*AM*	*AL*

In terms of sought information, the principal rationale that drove the previous experiment still motivates the current one. Regarding both SN configuration, the task completion times stood as the most important piece of data to be collected as the main purpose of the study being to compare performances. However, unlike the first test where it was lacking, a track of the teleportations was kept in order to help identify whether the participants use the system in a smart way. In the Hitlist and Alpha conditions, the task completion times obviously also recorded, as well as the key pressed, to offer statistics about the usage of the shortcut keys.

In consequence, the experiment comprised 2 independent variables,

- the type of interface (four levels: SN with names, SN with letters, Hitlist, Alpha)
- the playlist length (three levels: "short", "medium" and "large")

and four dependent variables,

- the primary being the task completion times
- the use of shortcuts – take off point and landing point – (for SN configurations)
- the sequence of keys pressed (for the Hitlist and Alpha configurations)
- feedback through the SUS questionnaire.

General open observations were also made during the experimental sessions in order to spot any phenomenon that had not been thought of beforehand.

Tasks

In order to avoid the personal differences mentioned in the previous chapter, the study carried out was a within subject experiment: each participant performed the same tasks. The procedure included a series of 32 tasks: 9 per interfaces and 3 within each interface for each list size. The tasks consisted of selecting a particular song in the list, implying substantial scrolling time to enable a proper time comparison afterward.

Scrolling directions were purposely set up to mix upward and downward moves. The sequence of tasks as well as the sequence of interfaces has been randomised to avoid learning effect.

Material

Similarly to the first experiment, the application was developed using Microsoft Visual Basic 6. The participants were manipulating the system individually in the usability lab 2 of the Remax House, UCLIC department.

The use of the controls was obviously highly restricted. Only the following could be used:

- *Up/Down arrow* to scroll up and down
- *Enter* to select the requested entry in the list

- In the case of the SN configuration, *right arrow* would allow the jumps when the selected item was a link
- For the Hitlist, *left arrow* and *right arrow* enabled a Page Up (- 3 items) and Page down (+ 3 items) navigation
- In the Alpha condition, *left arrow* and *right arrow* permitted the horizontal scrolling between letters.

Procedure

After having consented to take part in the experiment, the participants were administered a demographic questionnaire which intended to find out about basic demographic information and their mp3 player usage. Once this filled out, they were briefly explained the course of the operations, i.e. that the study aimed at testing different list navigation systems and that they would have to perform a series of 36 tasks on 4 different interfaces with 3 list sizes. The emphasis was put on the fact that they should try to be as fast and accurate as possible, as well as behave in a natural way.

After that, the two applications were simultaneously launched and the participants could start the practice session. They could then familiarize themselves with the four interfaces before beginning the experimental stage. During this training programme, they were told a couple of specific points. First, the controls peculiar to each interface were fleshed out: I manipulated the interfaces in the first place before I asked them to do so and select a random song. The particularities of the screen were also pointed out, namely that cycling was not implemented as well as fast scrolling. Was also mentioned the fact that the tasks were timed. When the participants had completed the training session, they went on to the experimental procedure. They were given a very short explanation about the experimental environment. Once again, the emphasis was put on the fact that the use of the shortcuts for the SN interface was not obligatory. Then, for every single task, I pressed the corresponding button to display the song to find. Once the participant had seen the song, he hit ENTER and began to browse the list. ENTER was used both for starting and stopping the counter. After the 36 tasks had been performed, the participants were handed a modified version of the System Usability Scale, to get some feedback on the four interfaces in terms of usability and preferences.

Ethical considerations

The procedure governing the experiment has been approved by the Psychology Department's Ethics Committee through Dr. Essi Viding. Moreover, by signing the Informed Consent Form, the participants are aware of:

- the possibility to withdraw at any time and to ask for a deletion of their records
- themselves being observed
- the anonymity of the data stored
- the use of their data under the Data Protection Act 1998.

Ethical approval number: MSHCI/2007/015

2. Results

The 15 participants selected for this experiment were all aged between 19 and 35. 73% (11) of them owned or had ever possessed an mp3 player for more than a year, therefore were quite familiar with the list browsing environment. However, only 27% (4) had ever had a small screen device (ex: memory sticks), the navigation of which primarily consisted of a one by one scrolling.

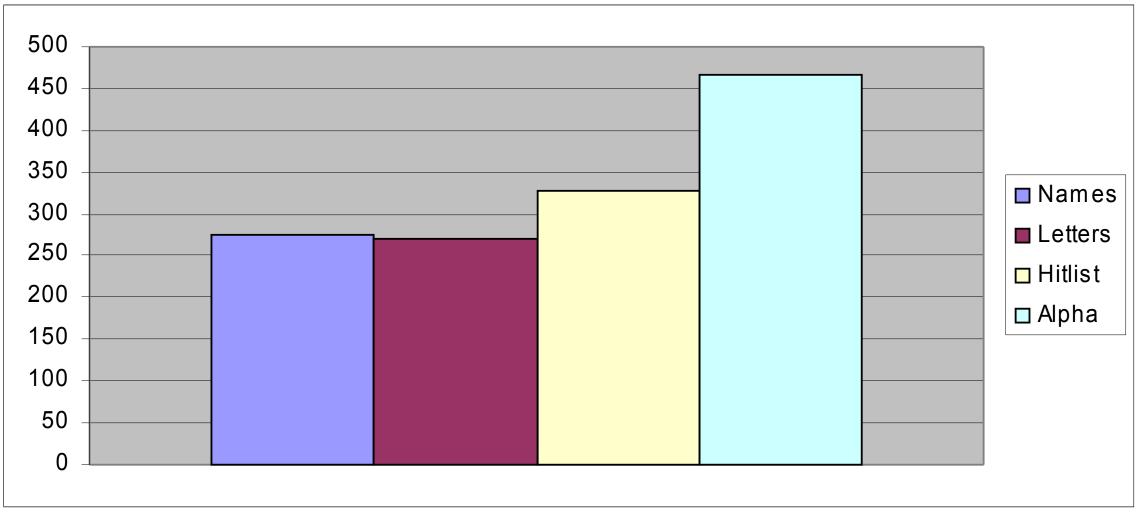
User performances

To provide significant results, rankings were established: for each task for each participant, a classification based on the completion time on each interface was created and points were attributed. For instance, the participant #1 on task S1 had the following records:

Names	**Letters**	**Hitlist**	**Alpha**
10 sec.	17	41	9

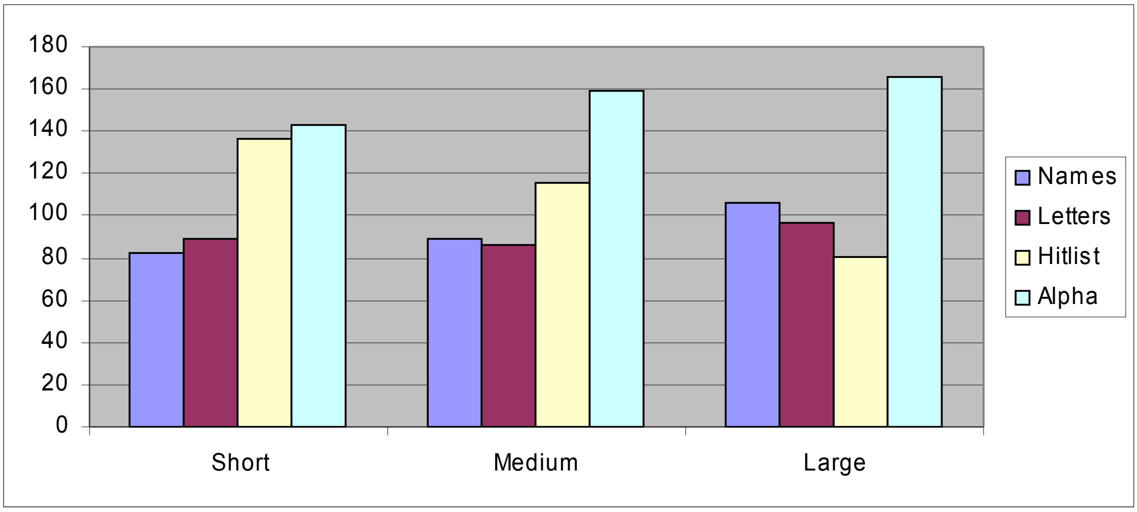
Therefore the following scheme was applied:
1st: Alpha 4 points
2nd: Names 3 pts
3rd: Letters 2 pts
4th: Hitlist 1 pts

The graph below shows the overall interface ranking based on the points attributed from the entire set of tasks among the group of the 15 participants. The scores depend here on the task completion times. Consequently, the higher the score is, the faster the navigation of the system is.



General ranking of the interfaces based on the time performances

At a first glance, this allows to say that the Alpha interface dominates by large the other navigation systems in terms of performances. Hitlist occupies the second place, whereas Letters and Names are almost ex-aequo, lying respectively at the 4th and 3rd position. But a classification by list size gives more insight.



Ranking of the interfaces by list size based on the time performances

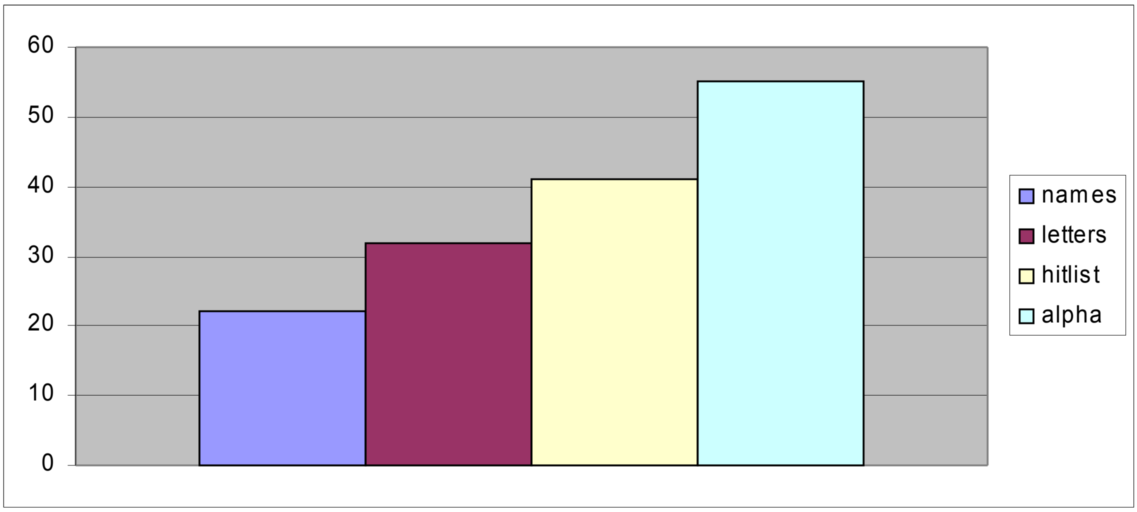
This clearly highlights 3 important points:

- Whatever the interface, Alpha always takes the lead. It is then by far the best system in terms of speed.
- The force of Hitlist decreases when the amount of items in a list increases. In other words, the bigger the list gets, the less effective the system becomes. This will be explained further in the report.
- The two SN systems appear to be of interest as soon as the list grows. This matches the conclusion of the first experiment. But they still cannot compete with the dominant interface.

Usability feedback

Satisfaction

The following graph presents the scores based on question 10 of the SUS questionnaire which asked participants to rank the interface by order of preference. The best score represent the most preferred system (the same marking scheme as previously has been applied).



General ranking of the interfaces by order of preference

The ranking is quite meaningful. The SN systems are not as successful as the two others.

Randomness

People generally do not like uncertainty. This is what they primarily echoed in the SUS questionnaire handed after the experiment. The randomness appeared for most of them quite destabilising and this point also stood out in the first experiment. The unexpected and confusing character of the two SN based algorithms did not favour its integration and appreciation. Similarly, a bad distribution of the links throughout the list can frustrate the users, who expect a performing system but end up by browsing a simple list not allowing a fast navigation. Consequently, they also expressed the need to structure the shortcut system. A couple of interesting ideas emerged from their experience: the shortcuts could occupy positions following a certain pattern (for instance, one every 10 entries) or they could link to songs from the artists that have the most entries in the list. These are potential directions for future design, showing in the same occasion that the participants had a good grasp of the system and its limitations. However, the twofold randomness has one positive trait according to one of the participants: it can help rediscover the playlist and highlight forgotten songs precisely by providing links to random entries.

Controls

The use of the controls also turned out to be discussed and can be subject to improvements. The main controversial key was the LEFT arrow. Among the participants who manifested interest to this key (11), 20% of them would have wanted an option for cancelling the latest shortcut gone through. The remaining users would have preferred to use LEFT as a back move through the alphabet. Clearly, if a link is supposed to take the selection further down the list, they would teleport by pressing RIGHT arrow, but if it takes them upwards, then LEFT arrow should be used.

Navigation
In terms of navigation, the SN algorithms did not provide sufficient satisfaction either. The Names interface, and the Letters interface to a minor extent as well, asks for too much cognitive effort and situational awareness when browsing the lists. Similarly to the first experiment, the use of the shortcuts demands reflection: the participants have to mentally represent the alphabet and decided whether a given letter will bring them closer to another given letter. Moreover, those algorithms maintain a situational awareness in that the users have to pay a careful attention to the list while browsing, to be sure not to miss any potentially efficient shortcut. Comparatively, the participants could use the Hitlist system while not even looking at the list for a few seconds… These observations apply even more to the Names interface since the string to analyse contains more information.

Alpha and Hitlist
Regarding the two non SN-based algorithms, a few notable things were mentioned. Alpha received all the praises: it was widely adopted by 90% of the participants and proved to be very intuitive, easy to use and performing. On the other hand, Hitlist, the navigation system of which was straightforward and easy to assimilate, turned out to be quite limited. The feedback gathered especially singled out one drawback of the system. The longer the list gets, the more "boring" and time-consuming it becomes to scroll to reach the desired song. This system which is in fact a slightly enhanced version of the traditional scrolling (3 by 3 instead of 1 by 1) brings up the same annoyance as the latter when the list size increases. This also matches the outcomes of the graphical analysis.

General observations

Cognitive factors
Even though cognitive aspects had more chances to be revealed through the SUS questionnaire, I observed a certain number of interesting aspects. In terms of level of attention, as hinted at by the questionnaire, the two SN systems required more awareness in that each element of the list has to be considered in case it is a link. On the two other interfaces, the users expect what will take place and can relieve their situational awareness. With respects to cultural differences, an important point has been made, which restricts the implementation of this algorithm for certain markets. When Asian participants were testing the SN systems, they would repeat loudly the alphabet to retrieve the sequence of letters corresponding to the links they were considering. This shows here that an additional effort has to be made; people unfamiliar with the Latin alphabet would struggle more to find out about the usefulness of a shortcut. In the same way, such a system would not work on writing system not based on an alphabet (for instance, the Chinese logographic system). Another thing that I had the chance to spot was that annoyance and boredom can surprisingly speed up task completion times. But I suppose this is only valid in an experimental condition where formal assignments are given to subjects.

Strategies

Several strategies were observed during this experiment. As they were asked to be as fast and accurate as possible, the participants put into practice different tactics. The first important remark concerns the use of strategies itself: strategies are not fixed, they are evolving. Often, the users began to browse the lists following a certain pattern and after a while, allegedly to improve their performances, changed their tactic.

Studying these tactics was also of great interest. One particular strategy observed consisted of using only links that would take the selection on an entry of the artist to be sought. Basically certain participants would only consider the shortcuts if the latter were connecting to songs from the same artist as the song to be selected. This usually happened when the users are not familiar with the content of the list, at the beginning of the experiment. After a few minutes, this could mutate into another similar strategy: a lot of participants become, after a while, only interested in connections to one or two letters away from the entry to select. This means that they were not considering short-range nor long-range connections, only very useful ones. The opposite tactic could appear as well insofar as the participants could get tempted to use any shortcut taking them closer to the goal, "less far to the song to reach". Obviously the "one-letter forth" moves were not drawing a particular attention, but a couple short-range connections to achieve the goal was often the result of this strategy. However all these behaviours can be called into question and one can wonder to which extent the experiment suffers from the Hawthorne effect. Did the participants use shortcuts because they felt supposed to do so when being presented this novel interface?

Limitations and restrictions

Performances were somehow slightly hindered by the limitations and restrictions of the application. The first concerns the playlist itself, as a confusion emanated from the band whose name started with "The". A third of the participants would expect to find "The Beatles" under B rather that T and got therefore destabilized. The alphabetical arrangement I applied is however similar to most of those existing on current systems. The second negative facet was the disposition of the controls. Half of the participants tended to use ENTER as the teleportation key for shortcut for the few first tasks, although I had put a particular emphasis on this aspect beforehand. This clearly indicates that a special key devoted to the shortcuts would not be necessary. But it also tells that half of them consider that using a link would bring them to "a new list" rather than further up of down in the list, as ENTER cannot be assimilated as a navigation key. By assigning the RIGHT arrow to the shortcut moves, I was hoping speeding up the process by grouping the navigation keys. Apart from that and of relatively minor importance, the distribution of the shortcuts across the lists could in some cases discredit the SN algorithms inasmuch as pointless links (linking to the next song, for instance) or too few links would kill the interest raised by the novelty effect of the interface.

3. Conclusion

After having demonstrated in the first experiment that the SN system was literally better than nothing in terms of navigation and this being particularly true on large lists, I tried in this second study to challenge this system with existing fast navigation environments for browsing lists. The factors examined were double: proving that the SN interfaces overrun competing systems and that using one letter instead of an entire chain of characters as the links would lead to better performances. To achieve so, the study staged the use of an mp3 player, the kind of which that does not incorporate advanced control features and low screen capacities.

Regarding the time performances, the study managed to confirm the first hypothesis, which stated that SN algorithms would allow a "less fast" navigation than the other systems. Therefore the results go against the predictions and it can be declared that the SN interfaces may not allow a navigation system as fast and effective as expected. However this statement can be attenuated by a pinpointing analysis: even though Alpha appears to be much more powerful on all interfaces, the SN based systems still turned out to be more effective than Hitlist on medium-sized and large lists. Therefore the first hypothesis is only partially confirmed. With respects to the usability feedback, the Alpha system still proved to be by far the most usable and easy to learn, which makes it the best overall navigation system. The SN systems are relegated to the last positions in the usability ranking, which stands in accordance with the first experiment results. This is mainly due to the higher cognitive effort demanded by those interfaces and the destabilising randomizations.

With regards to the second hypothesis, the study revealed that, first, the participants tend to prefer the *Letter* version of the SN system. The SUS questionnaire unveiled very similar results on the different points for the two systems, but the final ranking by order of preference clearly favoured the *Letters* version over the *Names* version. However, the same observation cannot be made about the performances for both. The analysis showed very similar results on the three list sizes, and even if the *Names* version seems to have a small advantage, the margin is not significant enough to generalise this finding. In sum, as there is almost no difference in terms of performance but a slight gap in terms of preference, the *Letters* version would probably be slightly more popular.

The bottom line of this second experiment is that the SN algorithm may not be as powerful as expected. When challenged to potential competitors, it showed some limitations, namely a reduced usability and higher task completion times. Even though its competing interfaces (Alpha, Hitlist) were taken from devices with better capabilities than simple "Mp3 players – memory stick", one can question the domain of applicability of this algorithm. To be cleverly used, it seems that it should not be implemented everywhere, because of its drawbacks. This is what the next chapter will discuss: why does it not work effectively and in which domain could it be useful.

7. Discussion

In this chapter of the thesis, I will briefly recall the findings and outcomes of the two experiments. Following that, the discussion will principally focus on the reasons that can explain the reluctance to the SN interfaces and will end up by proposing potential improvements that could favor its adoption.

1. Summary and outcomes of the experiments

The first experiment, which had the participants manipulate a simulation of a mobile phone contact list, aimed at proving the added-value of the SN navigation system. The between-subject experiment therefore compared performances and satisfaction rates on the SN interface and a regular list. It turned out that the larger (in amount of entries) a list gets, the better the performances are, in terms of task completion times, on the SN interface. However, a usability evaluation showed that although the new structure was easily discoverable, the participants would not opt for such a system, due to the confusion caused by its "randomizing" facet.

This led me to conduct a second experiment. After having proven that the SN interface could be qualified as a shortcut system, since it allowed to perform tasks faster than on regular lists, the goal of the experiment was to bring in competing shortcut systems. It was also a good occasion to test the format of the links, namely a sole letter or a whole string of characters. The results demonstrated that, first, with respect to the performances, the SN interfaces can hardly compete with the two other systems (even though the effectiveness increases when the lists grows up), and second, on the usability side, the participants would still prefer the competing systems, judging the SN interfaces too confusing and too compelling in terms of cognitive effort. Besides, the letter version appeared to be the most appreciated.

Consequently, these findings brought up a question subject to discussion:

- Why isn't the SN system adopted?

To a minor consideration, the following points will be discussed as well:

- Why did the participants prefer the letter version?
- Why does the SN interfaces show greater efficiency on large list?

2. Reluctance to using the SN system

To a certain point, the users seem to dislike this novel interface. Even though the performances turned out to be interesting, there is a clear reluctance to using this system.

Destabilizing randomizations

One of the main factors that the participants voiced in their feedback concerned the double randomization of the shortcuts, i.e. the locations and the destinations of each link. The fact is that people like certainty, they like controlling their environment. By randomizing its structure, the interface loses to some extent its stability and thus contributes to displeasing the user. In sum, the display of the lists becomes at some point "user-independent", which is quite destabilizing. Unease and confusion are the feelings that emanate from this. Moreover, a bad randomization (for instance, a link connecting to a very near entry) can emphasize this disorientation. The case where the randomizations would occur only once during the first utilisation of the device (as suggested by a couple of participants) would not seem to bring much improvement, as mobile phone contact lists and mp3 player playlists are environments that change very frequently; therefore to be consistent, the set of links should be updated regularly.

High cognitive load

Another worry that participants expressed more or less directly dealt with the mental effort they had to make to use the system. The first decision making process that differentiates this method from other systems obviously connects to the links. The user has to mentally visualize whether a given link will take him closer to the goal by comparing alphanumeric characters (most often only letters). In addition, he is not even sure to do the best choice given that he may miss a more effective link further down in the list. And the negative effect caused by this overload could be amplified depending on the user moods (tiredness, nervousness…) or origins (not familiar with the user alphabet). Apart from that, the system demands as well a sort of situational awareness. The randomization of the location of the links operates so as the user never knows when (or where) a shortcut will appear. Under a different angle, the user always expect the next item brought on the display to be a useful shortcut and therefore has to pay a lot of attention while scrolling through. In comparison with Hitlist where the user can almost be distracted or look elsewhere while scrolling, the SN interface compels the user to maintain his level of situational attention to a high level. Moreover, such a system only suits languages with a proper alphabet. Let us consider the example of the Chinese sinography which comprises more that 45 000 pictograms. There is no arrangement order. The SN shortcut systems would then get dull since one of its fundamental characteristic (see introduction) would broke down (namely its alphabetical structure). To summarize, overloading the user mental processes has a cost: it is time-consuming and asks for unnecessary effort.

Familiarity with existing systems

In the case of the mobile phone experiment, the SN interface proved to be better (on large lists) than a system with no shortcut at all. However, the usability feedback revealed that even though the system enabled fast navigation, people would not like to use it. I suspect that this judgement is underpinned by an implicit comparison with what they actually do when manipulating their own mobile phone. Most of us are now familiar with the keyboard shortcuts: when browsing my 170 entries contact list, I will press **5** if I want to select John. This action will take the selection to J and will save me a large amount of keystrokes. It will basically take 2 seconds at maximum. Such a system overruns by large the SN system. In consequence, asking people whether they would like to get rid of an ultra fast and straightforward technique at the expense of a cognitively demanding and confusingly randomized system is most likely to result in a negative answer.

In the use case of a mp3 player, the observation is the same. But the context is here slightly different as the kind of mp3 player the experiment investigated does not currently embed any shortcut system. In this situation, Alpha appears to be the equivalent of the keyboard system on mobile phones, namely a very quick and straightforward interface. As more than 70% of the participants never had to deal with such a kind of music player, they imagined that Alpha, which is less demanding in terms of mental effort and quicker in terms of performances, would be the best fit. Nevertheless, the usability feedback also ranked Hitlist as second. The explanation would be the reluctance of the SN system (for the reasons mentioned earlier) rather than a particular taste for Hitlist. But one thing is sure: the time performances are not the only factor influencing the appreciation of an interface (only 6 out of 10 people who ranked Hitlist among their two first choices of preference also had Hitlist among their two best interfaces in terms of performances). In other words, 4 participants preferred to use Hitlist whereas it did not permit them to achieve their goals as fast as the other interfaces.

Entertaining navigation

To some extent, the SN interface appears to resemble a sort of game: by that I mean that it can be an entertaining way to browse a list. Here, entertaining does not mean amusing, but rather signifies “unusual” or “tricky”. It belongs to a sort of recreational activity. This might be as well what people do not want: when browsing a contact list, there is usually no time for fun, whatever the purpose of the phone call is. Selecting a name is something the users want to be clear and brief; they probably do not want to be bothered nor distracted by some impish navigation systems. This applies less importantly to the case of the mp3 player, where the whole device is dedicated to recreational or relaxing activities. As hinted at by a participant, the randomizing algorithm can help rediscover tracks that have not been played for a while.

3. Other findings

Letters prevail over strings

The outcomes of the second experiment highlighted the fact that even though the time performances are fairly even for both conditions, the letter version of the SN interface has a higher satisfaction rate than the name version. This can be explained by several reasons. First it seems almost obvious that one letter is easier to mentally process than an entire string: when confronted to a chain of characters, people tend to consider the entry as a whole, not as a sequence of letters. It therefore calls for a more important effort to spatially (in the list) locate the destination entry of the link. In the same way, the letters bring more structure and certainty as they lift the selection to the first item of the corresponding letter whereas the names can land the selection anywhere.

However one can compare in depth the usefulness and effectiveness of both systems. As said before, the letter version is more structured and clearer. But let us now consider a playlist of 300 entries, 250 of which start with P. In this case, a link to P would appear to be much less effective than a link towards any random entry starting with P, which would take the selection much closer to the sought song (if starting with P in this case) versus taking the selection to the first song beginning with P. In sum, the two systems do not fulfil the exact same function.

The larger the better

The major finding of the first experiment, confirmed by the second one, relates to the increasingly better performances as lists grow up. The phenomenon is simple: due to the randomization, the destination of the links can be anywhere on the list. This means that the larger a list gets, the larger is the number of regular moves that can be saved by a useful jump through a link. Indeed, the analogy with the small world phenomenon is here particularly evident: having random long range connections within a network of 10 elements would seem to be quite ineffective whereas it would turn out to be much more productive in a network of 200 elements. Or what makes a difference when comparing the SN interface with regular lists or Hitlist is the time saved. And time is saved by saving moves. In conclusion, the more moves that can be taken over by a jump, the greater the time difference gets (in comparison with the other systems) and the better the performances are.

4. Potential improvements

Less randomization

As clearly stated before, the randomizing trait of the SN interface was the most controversial factor for the adoption of the system. Indeed, the randomization is twofold: the position of the links in the list and the destination of each link are randomly picked up. This feeling of disorientation principally emanates from the lack of structure triggered by this double randomization. But there could be a way to thwart this effect: removing one of the two randomizing agent. As the main force of the algorithm dwells in the long range connections, the randomization of the destinations should be kept unchanged. However, the positioning of the links is a factor that could be very performing if well controlled. Therefore structuring the location of the links could be a solution and could bring some advantages. By arbitrary setting up the positions (for instance, one link every 5 elements), people would know when to expect a link and would probably have more power on the list. In addition, this could also sort out the situation where a bad randomization would occur – parts of the list cluttered by too many links or out of links –; the proposed structure would ensure a steady link presence across the list. But this solution is somehow breaking down the small world paradigm, thus it needs further investigation.

Allowing back moves

In the course of the two experiments, using a link was a one-way move: neither a “cancel” nor a “back move” function was implemented. Nonetheless, some participants expressed their concerns about not having the possibility to go backwards. De facto, making this possible would allow the cancellation of accidental or misjudged moves. Consequently it would be interesting to see the added-value of a back move function embedded in the navigation, i.e. to see whether permitting the going back from the destination of the link to its taking off location would add ease and control. Another aspect related to the control was suggested in the usability feedback: both the RIGHT and LEFT arrow should be used to take links. The LEFT key could be employed for links leading backwards in the list (in the alphabet thus) and the RIGHT key for the forward moves. These two suggested features deserve to be examined carefully, again in order to detect whether they could facilitate the navigation.

8. Conclusion

In this last section of the thesis, I will endeavour in the first place to answer the research questions originally defined, before briefly moving on to the weaknesses of the study and to the future work that can be achieved to enrich the knowledge about this system. Eventually I will relate my findings to the value the work brings to HCI.

Outcomes

This thesis aimed at evaluating a novel interface for browsing lists. Throughout the study, I sought to answer the questions that such an enterprise raises. Here are my findings.

Does this system bring a significant added-value in terms of browsing lists more effectively?
Indeed, this new interface, based on the small world paradigm, does bring an added value. The experiments showed that the system can be qualified as a shortcut method as the time performances it occasions proved to be better than on a regular list. But this has to be toned down, as the effectiveness of this interface can only be revealed on lists where the number of items gets substantial enough.

Is this system more powerful and more usable than current or existing systems?
As this time, it is unfortunately impossible to say that it can compete with the existing system, at least with the systems tested in the experiments. In terms of both usability and performances, the results are not as successful as expected.

Would people be keen to adopt such a system, how easy to assimilate would it be?
Even though the feedback gathered from the participants demonstrated that the system was easy to handle and understand, people are not keen to using it. Due mainly to its double randomization which is a confusing factor, the interface appears to make people reluctant navigating through it.

What are the limitations of such a novel navigation system?
As hinted at in the previous section, the two randomizing aspect prevent the users to have a full control of their interface, which is quite destabilizing and stands out as the main drawback. Moreover, it is cognitively compelling.

Where could this system be possibly implemented? Where should it not be?
The study did not examine this aspect with a fine tooth comb and as the product is only in his early days, it is hard to give a definite answer. However, I would reckon

this system to be associated with low capacities (screen, controls) devices, connected at some point with the entertainment world (for instance mp3 players).

Limitations

Even though the study attempted to be as objective and accurate as possible, the scope of it has to be narrowed down, due to some limiting factors. First, the PC applications have to be called into question. Although I tried to "copy" interfaces on a computer, the simulations I devised can not replace the effectiveness of real models, which could not be manipulated for obvious reasons of resources. The second factor is the number of the competing systems tested. In the final ranking of the interfaces, the SN algorithm occupied the positions 3 and 4 out of 4, which gave a relatively negative outline of the systems. But one can think of other shortcut techniques in the aim of establishing a more complete classification, where the SN interfaces would not be laying at the last positions. Eventually, one can probe the effect of the content on the navigation. Participants were facing lists of elements unknown to them, which may have perturbed them. They may have behaved differently if they had been browsing their personal device.

Future work

A couple of areas would deserve to be investigated farther. The main domain that could be reviewed would be the randomization, wherein one parameter could be removed, as discussed in the previous chapter. By doing so, one could find out whether the randomizing aspect of the system is the principal reason leading to the reluctance the participants showed. One could also question the influence of the display on the navigation, in terms of screen size or lines displayed to see if it plays a role of the navigation speed.

Value to HCI

This navigation system designed following the small world networks paradigm is undoubtedly a novel way for browsing list. This thesis presented a first approach to it in that it evaluated its performances and the way it would be used. Although a lot remains to be achieved to render this system competitive and to make it adopted widely, this is a first step forward in the attempt of speeding up the navigation across lists. As rapidity has become an unavoidable criterion in nowadays world, this might bring a small contribution to the general expectations and is then worth being explored further in depth. I hope that this will trigger interest among researchers and that it will lead to further studies intending to broaden the body of knowledge of this navigation interface and consequently of HCI.

9. Appendices