IADIS International Journal on Computer Science and Information Systems
Vol. 3, No. 2, pp. 66-77
ISSN: 1646-3692

MULTIMODAL INTERFACE DESIGN: AN EMPERICAL INVESTIGATION ON EFFICIENCY

Mohammad M. Alsuraihi
Department of Computing, School of Informatics, University of Bradford

mmnalsur@bradford.ac.uk

Dimitrios I. Rigas
Department of Computing, School of Informatics, University of Bradford

d.rigas@bradford.ac.uk

ABSTRACT

This paper introduces an empirical multi-group study for investigating the experienced use of visual and multimodal interaction metaphors for designing interfaces. The study aimed at comparing the efficiency of multimodal interaction using voice-instruction and speech along with limited use of the mouse and the keyboard against the efficiency of using visual-only interaction metaphors for using interface-design environments. Efficiency of these interaction metaphors was compared in regard to task accomplishment time and frequency of error-occurrence. In order to carry out this comparative investigation, three experimental interface design toolkits (TVOID, OFVOID, and MMID) were built from scratch. TVOID and OFVOID interacted with the user visually only using typical and time-saving interaction metaphors. The third environment MMID added another modality through vocal and aural interaction. Then, these environments were tested independently by three groups of experienced users. Each group consisted of 40 users. The results showed that the use of vocal commands and speech with limited use of the mouse for completing tasks was more efficient in terms of shortening task accomplishment time and reducing the number of errors than the use of the typical and time-saving visual-only interaction metaphors: graphical menus, toolbar, toolbox, properties-table, hot-keys, scrollable-tags, instant-menus, textual-help, and textual messages.

KEYWORDS

interface-design, usability, efficiency, experienced-user performance, multimodal interaction, voice-instruction, speech.

1. INTRODUCTION

User interfaces of the existing interface-design environments are annoyingly crowded with graphical widgets that interact with its target users (designers) through the visual channel only. The auditory system has been neglected in the development of user-interfaces (Rigas *et al*, 2001a). The heavily focus on conveying information through the visual channel when designing interfaces causes the user to experience information overload by which important information may be missed (Oakley *et al*, 2002). In order to solve complexity problems with the current visual user interfaces, Rigas *et al* (2001b) suggest that interfaces could be designed in away that visual metaphors communicate the information that 'needs' to be conveyed to the user and the auditory metaphors communicate the other part of information which is used to perform tasks. The usability problems: miss-selection and interface intrusion into the task that emerge from using graphical widgets, like menus, buttons, and textual messages, could be solved by employing auditory feedback (Beaudouin-Lafon and Conversy, 1996; Brewster and Clarke, 1997). The Emacspeak (Raman, 1997) was a pioneering project as it integrated spoken feedback with application contents. It was found that the use of speech and other auditory metaphors, like earcons and auditory icons, helped the users to make fewer mistakes when accomplishing their tasks, and in ‘some cases’ reduced the time taken to complete them (Rigas *et al*, 2003).

Previous studies suggested the use of sound (speech, earcons and auditory icons) for enhancing the efficiency and solving the usability problems with the graphical metaphors used for designing interfaces. In our opinion, sound only enhances the direction of interaction between the interface and the user, but not the opposite. Interaction is a dual-direction communication between the user and the interface.

In this study we introduce voice-instruction as another modality for enhancing efficiency of the other direction (i.e. between the user and the interface). The study aimed at investigating the efficiency of experienced use of voice-instruction and speech along with the limited use of the mouse and the keyboard against the efficiency of using the visual-only interaction metaphors: graphical menus, toolbar, toolbox, properties-table, hot-keys, scrollable-tags (i.e. tags of commands that are shown successively as the mouse scrolls up or down), instant (popup) menus, textual-help, and textual messages. The results showed that the employment of voice-instruction have dramatically reduced the potential need for using the mouse and the keyboard in most of interaction scenarios, namely, activating menu-command functionality, drawing controls (objects), setting properties, and searching for help. This led to significantly shortening task accomplishment time and reducing the number of errors.

2. EXPERIMENTAL TOOLKITS

In order to carry out the efficiency study presented in this paper, three experimental interface-design toolkits were built using Microsoft visual C#.

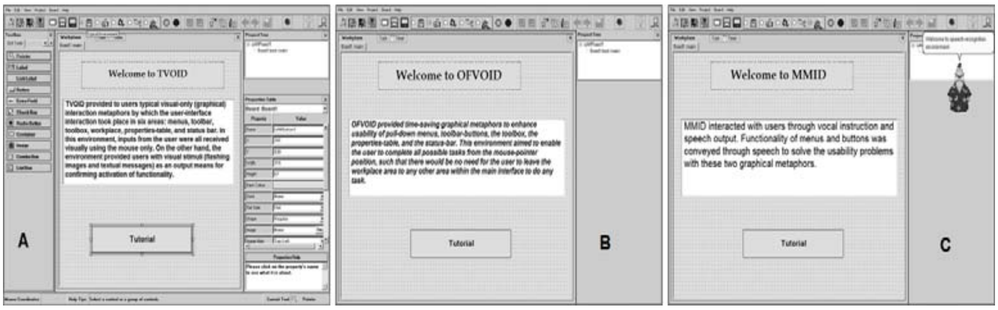

Figure 1. Screenshots of the experimental toolkits: (A) TVOID, (B) OFVOID, and (C) MMID

2.1 TVOID

TVOID or Typical Visual-Only Interface Design toolkit imitates the style of interaction implemented in most of the existing interface-design environments like Microsoft Visual C# and Java NetBeans IDE. It interacts with the user visually-only with no involvement of other senses like the auditory system. This interaction takes place in six areas within its main interface: menus, toolbar, toolbox, workplace, properties-table, and status-bar. Figure 1.A shows a screenshot of TVOID.

2.2 OFVOID

OFVOID or On-the-Fly Visual-Only Interface Design toolkit allows the user to do all tasks from the position of the mouse-pointer. There is no need for the mouse-cursor to leave the workplace area to do any job. The environment is facilitated with a number of time-saving features: hot-keys, scrollable-tags, instant-menus, and interactive mouse-cursor. Hot-keys in this environment are used to activate menu-command functionality like for instance using the keys Ctrl+C for copying. Scrollable tags are used to select tools while drawing by scrolling the mouse up/down over the form being designed until the required tool is reached. They are also used for setting properties of drawn controls (objects), which can be done by scrolling the mouse up/down over the selected control until reaching the required property. Instant or popup menus are comprehensively used for activating menu-command functionality, drawing, and setting properties. These interaction metaphors worked as substitutes for the menus, toolbar, toolbox, and the properties-table used in TVOID. The interactive mouse cursor shows currently active tool and mouse coordinates. It works as a substitute for the status-bar. Figure 1.B shows a display of this environment.

2.3 MMID

MMID or Multi-Modal Interface Design toolkit provides a combination of visual, vocal and aural interaction. It is a speech-recognition and text-to-speech based environment that allows limited use of the mouse and the keyboard. There is no need for the user, in this environment, to use any of the graphical interaction metaphors implemented in TVOID and OFVOID. It rather enables the user to interact with it from the position of the mouse-cursor using vocal instructions. The system command receptor in MMID is represented by a friendly character that listens to commands and interacts with the user through speech and facial expressions. The vocal commands are designed to be in the form of simple one to three English-words. Figure 1.C shows a screenshot of MMID.

3. EMPIRICAL MULTI-GROUP STUDY

In order to fulfill the aim of the study, the experimental design toolkits were tested empirically by three independent groups of users. The reason of performing these experiments independently was to avoid the learning-effect or the experience that can be gained by the users when testing one toolkit and moving to the other. Each group consisted of 40 users. All the groups were asked to accomplish the same tasks (9 tasks). The tasks were formulated to be increasing in complexity. Each task was composed of two to five functions. Each group attended three sessions of training before commencing the experiments, for learning how to use the toolkit assigned to it to test. Each training session took two hours. This intensive training was necessary to make the subjects experienced in using their toolkit as this was an essential objective to fulfill the aim of this study. Efficiency was measured by timing accomplishment of each function and frequency of error-occurrence under the three environments.

Also, one group of users, consisting of 80 users, was recruited to test efficiency of textual and multimodal help interaction metaphors dependently. The users in this group were asked to explore how to do a specific task by using the textual and then the multimodal help tools. The learning effect had no influence on the data to be collected, because using the textual-help tool was completely different from using the multimodal-help one in terms of searching for keywords and conveying help content. Efficiency of the two help interaction metaphors was measured by timing access to the required help-page.

4. RESULTS AND DISCUSSION

4.1 Task Accomplishment Time

As mentioned earlier, the On-the-Fly environment (OFVOID) enabled the user to do all tasks from the workplace area (mouse-cursor position) using time-saving interaction metaphors. Nevertheless, its efficiency in terms of task accomplishment time was found not significantly different from the other visual-only environment TVOID (t = -0.37, P = 0.72). Figure 2 shows that 78% of the tasks were completed faster using the typical environment TVOID. There were two reasons behind this result. First, the use of scrollable-tags in OFVOID motivated the users to miss the required tool/property at least once each time a tool/property was selected. Secondly, the use of instant-menus in OFVOID implied branching from one menu to another until reaching the required tool/property. This was similar to using the graphical menus in TVOID. However, TVOID offered this feature to only activate menu-commands functionality (e.g. File, Edit, View …etc), while OFVOID enabled the user to use this feature to perform more actions like selecting tools and setting properties. This implied more branching, and hence longer tool/property access time. Figure 2 also shows that task accomplishment using the multimodal environment MMID was faster than it using the two visual-only environments. This difference was found significant (F = 4.39, P = 0.02). The multimodal environment enabled the users to command the system by voice and listen to feedback and functionality confirmation, by which the use of the mouse and the keyboard was limited.

The following paragraphs put more light on these results by comparing efficiency of the different sets of interaction metaphors offered in the three environments according to functionality.

4.1.1 Menu-Command Interaction Metaphors

Most of the menu-commands in TVOID were activated using the menus instead of the toolbar. Table 1 demonstrates that most of the users in this environment preferred to use the menus more than the toolbar. Also, it shows that most of the users who tested OFVOID preferred to

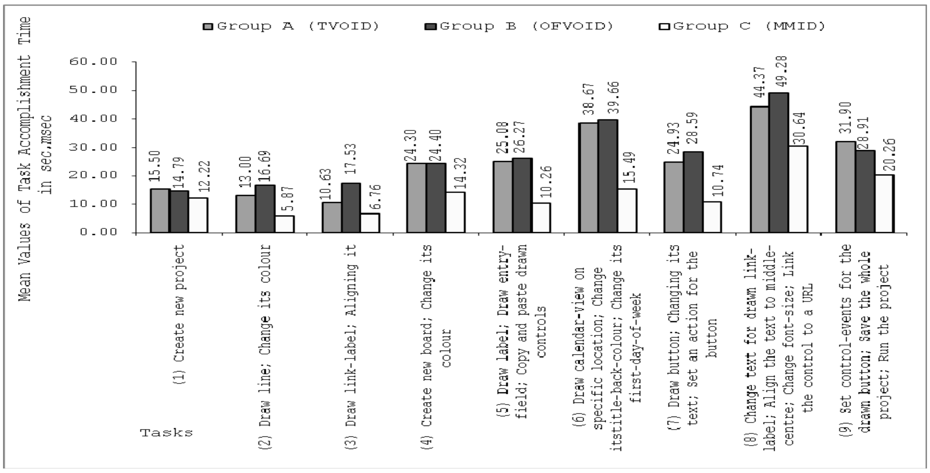

Figure 2. Mean values of time taken to accomplish nine tasks by 120 experienced users using TVOID (Group A), OFVOID (Group B) and MMID (Group C)

Table 1. Frequency of users who used the menus and the toolbar in TVOID, and instant-menus and hot-keys in OFVOID to activate menu-command functionality

Functions	**Number of Users Who Used**			
	TVOID		**OFVOID**	
	Menus	**Toolbar**	**Instant Menus**	**Hotkeys**
Activating the command: *New Project*;	38	2	2	38
Activating the command: *New Board*;	38	2	39	1
Activating the command: *Select All*;	40	0	34	6
Activating the command: *Copy*;	39	1	34	6
Activating the command: *Paste*;	39	1	34	6
Activating the command: *Save Project*;	36	4	29	11
Activating the command: *Run Project*;	13	27	29	11
Average	**35**	**5**	**29**	**11**

activate the menu-commands using instant-menus instead of hot-keys. Using these two interaction metaphors (menus and instant-menus) implied full dependency on the mouse and more load on user's brain when looking for the right menu-item, which sometimes implied branching from one menu-list to another. However, the results in Figure 3 show that the use of instant-menus in OFVOID was more efficient than the use of menus in TVOID in terms of menu-item access time. The feature of emerging instant-menus from the mouse-cursor position in OFVOID (i.e. pressing the mouse right button to show a menu) saved the time needed for moving the mouse to the right menu in TVOID. On the other hand, Figure 3 shows that the users of MMID completed their tasks faster than the users of the two visual-only environments TVOID and OFVOID. The design of MMID limited the need to use the mouse or the keyboard as it replaced all menu-commands and keyboard entries with speech.

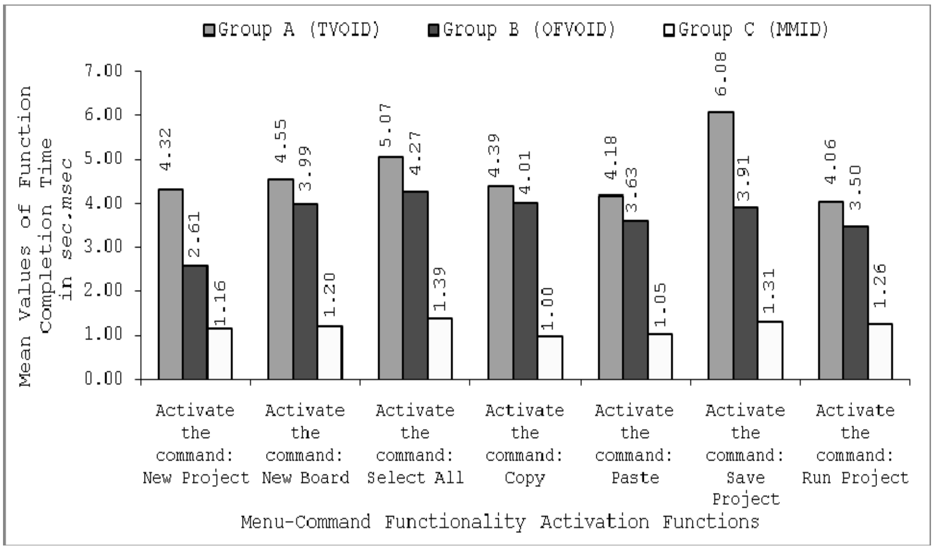

Figure 3. Mean values of time taken by 120 users to activate functionality of seven menu-commands using TVOID (Group A), OFVOID (Group B), and MMID (Group C)

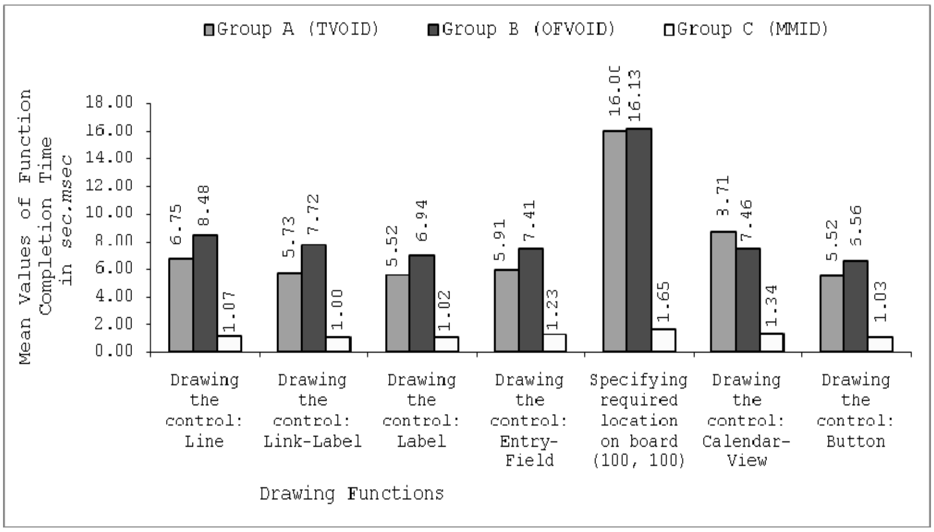

Figure 4. Mean values of time taken by 120 users to complete seven drawing functions using TVOID (Group A), OFVOID (Group B), and MMID (Group C)

4.1.2 Drawing Interaction Metaphors

Drawing objects visually-only using TVOID implied performing three actions: (1) selecting the required tool from the toolbox, (2) specifying the drawing location by reading mouse-coordinates from the status-bar, and finally (3) drawing the control using the mouse. Similarly,

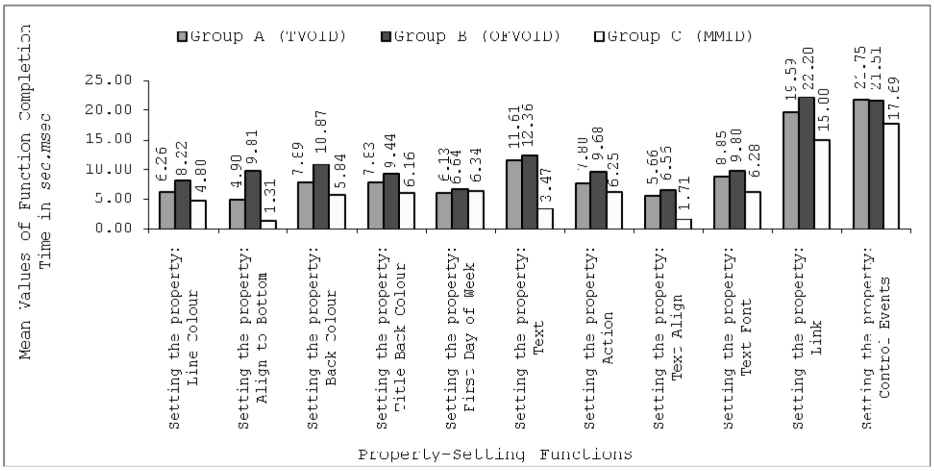

Figure 5. Mean values of time taken by 120 users to complete eleven property-setting functions using TVOID (Group A), OFVOID (Group B), and MMID (Group C)

drawing an object using OFVOID required doing three steps: (1) using instant-menus or scrollable-tool-tags to select the required tool, (2) specifying the drawing location by keeping looking at the mouse-cursor, and (3) drawing the object using the mouse. On the other hand, the users of MMID did not have to select tools or look at the status bar or the mouse-cursor to locate the drawing coordinates. Instead, they could directly command the system to draw at the required location by speaking location coordinates followed by the name of the required tool. These design features caused the dramatic difference shown in Figure 4.

4.1.3 Property-Setting Interaction Metaphors

The users of TVOID used the typical properties-table to set properties of drawn objects. The design of properties-table exposed all the listed properties to users and caused them the least mental effort when searching for a particular one. On the other hand, finding a property using scrollable-tag lists in OFVOID caused the users to put more mental effort. It was noticed that the users who used this feature to reach properties spent more time than their counterparts who used the properties-table. This result is shown in Figure 5. In addition, it was noticed that most of task-time during using scrollable-tags was spent in mistakenly miss-selecting (passing) the required properties; and then in returning to them by either scrolling up (short way) or keeping scrolling down (long way through the whole list) until reaching the required property. Sometimes this phenomenon happened repeatedly more than once. Scrollable-tags were found more error-prone than instant menus. The reason of this returned to the design of scrollable-tags, which were sorted alphabetically, and showed only one property (or command) each time the mouse was scrolled. This design motivated the user to think fast of where the potential position of the required property would be and concurrently scroll up or down based on the alphabetic order of the currently shown property’s name. This led to passing the required property at least for once. Observation of task-accomplishment during using scrollable-tags showed that 83% of the users made this error. On the other hand, Figure 5 demonstrates that setting properties by voice was more efficient than using the properties-table or scrollable-tags. Vocal properties saved the time spent by users to identify the locations of wanted properties within the properties-table and the time spent in keeping track of their alphabetical order during using scrollable-tags lists.

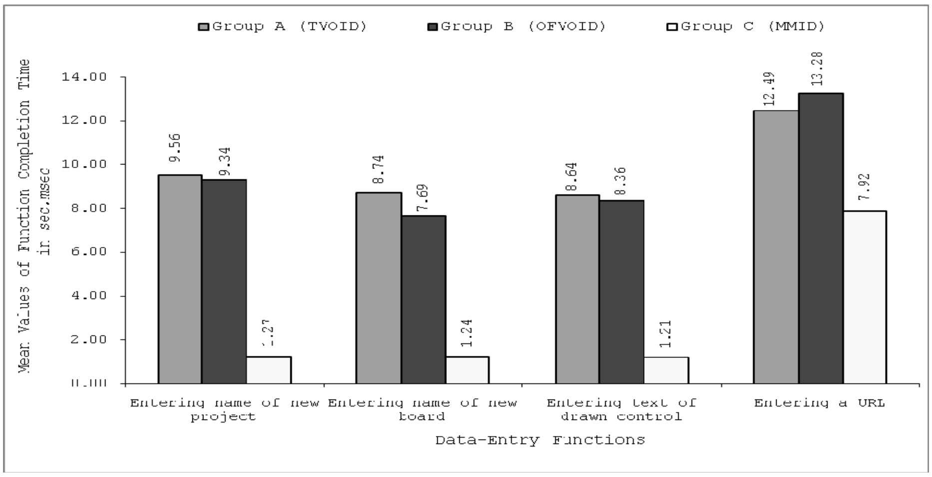

Figure 6. Mean values of time taken by 120 users to complete four data-entry functions using TVOID (Group A), OFVOID (Group B), and MMID (Group C)

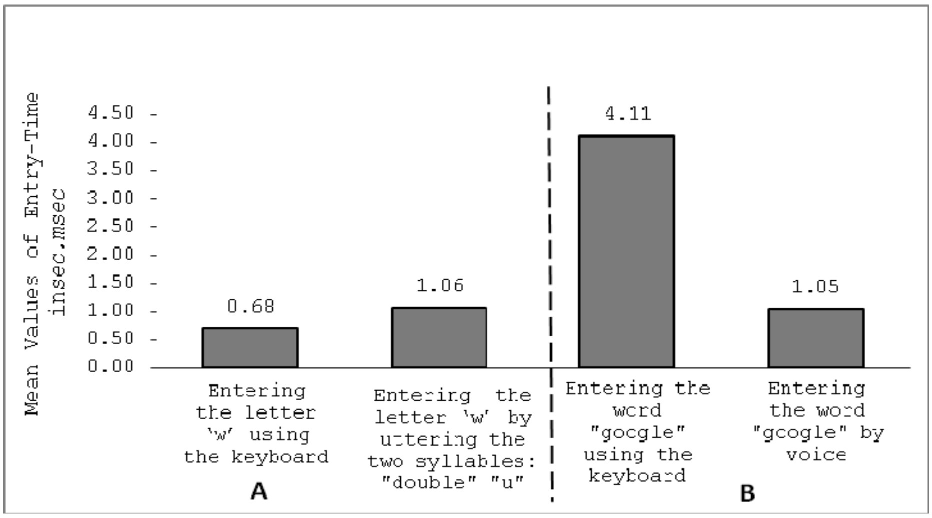

Figure 7. Mean values of time taken by 80 users to enter the letter 'w' and the word "google" using the keyboard and by speech

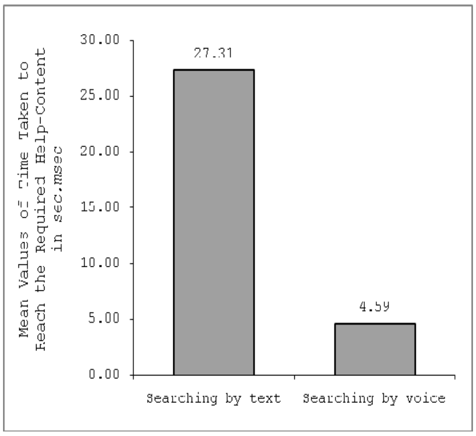

Figure 8. Mean values of time taken by 80 users to search for a keyword until reaching its help-content-page using the textual and multimodal help tools

4.1.4 Data Entry Interaction Metaphors

Figure 6 presents the mean values of time taken by the users to enter the required data during the experiments through the keyboard and speech. The results show that the users who entered data by voice using MMID performed much better than their counterparts who used the keyboard in the two visual environments TVOID and OFVOID. MMID saved efficiency from being dependent upon user skills when using the keyboard (i.e. accuracy and pace of typing). On the other hand, to enter abbreviation letters, like "www" by voice, these letters should be spoken one by one (i.e. “double u” “double u” “double u”). The time averages shown in Figure 7.A demonstrate that typing the letter 'w’, was faster than speaking it as speaking the letter implied uttering two syllables: "double” and “u". On the other hand, the results in Figure 7.B shows that entering a whole word by speech was faster than entering it using the keyboard. For example, to enter the word “google”, the user can speak it as a whole (i.e. "google"), not letter by letter (i.e. ‘g’ ‘o’ ‘o’ ‘g’ ‘l’ ‘e’) as when using the keyboard.

4.1.4 Help Interaction Metaphors

The help tools (textual and multimodal) were tested dependently by eighty users, such that each user was asked to search for a particular help-page using a particular keyword. The users were required to find out how to configure an event called *OnShow*, using the textual tool at first and then the multimodal one. Searching for the keyword "OnShow" textually implied using one of the conventional ways (i.e. the tab *Contents* or *Index*). Using the tab *Contents* implied doing redundant and irrelevant mouse movements and clicks during looking for the keyword.

On the other hand, using the tab *Index* implied using the keyboard and the mouse to find the required help-content. These two graphical metaphors caused the user to take much more time than should be during looking for the required help-page. Speaking the keyword directly using the multimodal tool was a faster way to reach the required page. This result is illustrated in Figure 8.

Table 2. Frequency of user errors during using TVOID, OFVOID and MMID

Errors	Number of Error Occurrences N/A stands for Not Applied		
	TVOID (Group A)	**OFVOID (Group B)**	**MMID (Group C)**
Mouse slipped and unwanted menu-item was hit.	10	1	N/A
Mouse slipped and unwanted button was pressed.	2	N/A	N/A
Mouse released by mistake during drawing and too small control was drawn.	4	5	N/A
Confusing a property’s value with another of another property.	2	N/A	N/A
Typing errors	26	12	6
Missing or passing required tool/property during using scrollable-tags.	N/A	60	N/A
System could not recognize spoken command. None action happened.	N/A	N/A	19
System misunderstood spoken command and unwanted action happened.	N/A	N/A	4
Total	**44**	**78**	**29**

4.2 Errors

Table 2 shows error-occurrences during interaction with the visual and multimodal metaphors implemented in TVOID, OFVOID, and MMID. Mostly, three errors could happen when using

MMID: typing errors, unrecognizing of vocal commands, and misrecognising of vocal commands.

The typing errors, shown in Table 2, were made because of using the keyboard. Most data-entry in MMID was done vocally. Therefore, the frequency of typing-error occurrence was reduced significantly. The problem with voice-instruction is sensitivity toward sound. This problem was caused the occurrence of the last two errors listed in Table 2. Speaking a command incorrectly or unconfidently led to the frequent occurrence of unrecognizing or misrecognition of spoken commands. If a spoken command was not recognized by the command-receptor in MMID, nothing would happen. This error occurred nineteen times during using this environment. A more serious problem happens if the vocal command was incorrectly interpreted because it will be very likely for an unwanted action to take place. This problem occurs when the command-receptor gets confused if the spoken command sounds different from the way it should be spoken. This error happened rarely in MMID (only four times), mainly, because MMID was designed to map voices to commands only if a high degree of confidence was reached.

Also, Table 2 shows that at least four types of errors could take place with the visual-only (graphical) interaction. The usability problem of mouse-slipping that is likely to occur when using menus and buttons did not happen at all during using MMID. The menus and buttons were replaced in this environment with vocal commands.

Also, the problem of finger-slipping on the mouse while drawing, which leads to drawing too small object was not applicable in MMID. This environment helped the users to avoid using the mouse by enabling them to draw by voice.

The problem of confusing a property's value with another happens when two properties have the same initial values. For example, in properties-table, the properties Name and Text of a button would have the value "Button1" initially when the button is created. This problem was not applicable in OFVOID and MMID because setting properties in this environment was done differently. In context of the study, TVOID, as an imitator of most of the existing interface-design environments, gave the same initial values for a control every time that control was drawn. This design tricked two users in Group A (TVOID users). The users were asked to set the property: "Text". These two users did not notice the property "Text" because it was far down the list; they set the wrong value (value of the "Name" property) because it was the first property that they saw showing the control's label in its value-field. The main problem in OFVOID that caused the occurrence of sixty errors was the use of scrollable tags to select tools and set properties as explained in section 4.1.3.

5. CONCLUSION

This paper introduced an empirical multi-group study for investigating the experienced use of visual and multimodal interaction metaphors for designing interfaces. The study aimed at comparing the efficiency of multimodal interaction using voice-instruction and speech along with the least possible use of the mouse and the keyboard, against the efficiency of using visual-only interaction metaphors for using interface-design environments. Efficiency was measured in terms of task accomplishment time and frequency of error-occurrence.

The results demonstrated that using speech for input and output when designing interfaces can significantly increase user performance in two ways. First, it shortens the access time to commands and messages more than the conventional visual-only (graphical) metaphors can do. Using graphical metaphors imply going in short and long paths (using the mouse and/or the keyboard) to reach a command or deal with a message. Secondly, using speech can efficiently save tasks from being interrupted by the inevitable nature of visual alerts, which require full visual attention. Conveying messages to the user using speech keeps the user aware of actions taking place outside the eye-focus area while performing designing tasks.

Looking at efficiency from the errors angle showed that using visual-only means to design interfaces can create a fertile environment for usability problems. The results argued that using graphical metaphors frequently until the user becomes familiar with their use would increase the probability for error occurrence. A high level of familiarity causes users to interact with graphical metaphors with haste and less care, which in turn causes errors to take place. On the other hand, the frequent practice of vocal instruction helps users to lessen the impact of speech recognition problems, as it makes them learn the way each vocal command should sound prior to uttering it.

This paper argues that using vocal and aural interaction metaphors along with the least possible use of the mouse and the keyboard can significantly enhance efficiency of interface-design in terms of task completion time and frequency of errors. Future research will investigate the effect of speech on the other parameters of usability: *effectiveness* and *satisfaction*.

REFERENCES

Beaudouin-Lafon, M., & Conversy,S. 1996, Auditory illusions for audio feedback, ACM CHI'96, Vancouver, Canada.

Brewster, S., & Clarke,C. V. 1997, The design and evaluation of a sonically-enhanced tool palette, ICAD'97, Xerox PARC, USA.

Oakley, I., Adams, A., Brewster, S., & Gray, P. D. 2002, Guidelines for the design of haptic widgets, BCS HCI 2002, London, UK.

Raman,T. V. 1997, Emacspeak-an audio desktop, IEEE Proceedings of Compcon '97.

Rigas, D., Hopwood,D., & Yu, H. 2003, The Role of Multimedia in Interfaces for On-Line Learning, 9th Panhellenic Conference on Informatics (PCI'2003), Thessaloniki, Greece.

Rigas, D., Memery, D., & Yu, H. 2001, Experiments in using structured musical sound, synthesised speech and environmental stimuli to communicate information: is there a case for integration and synergy?, Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing 2001.

Rigas, D., Yu, H., Klearhou, K., & Mistry, S. 2001, Designing Information Systems with Audio-Visual Synergy: Empirical Results of Browsing E-Mail Data. Panhellenic Conference on Human-Computer Interaction: Advances on Human-Computer Interaction, Patras, Greece.