EINDHOVEN UNIVERSITY OF TECHNOLOGY

Screen orientation: a matter of how you look at it

Citation for published version (APA):
Beumer, J. J., Klimbie, P. B., Poll, L. H. D., & Waterham, R. P. (1993). *Screen orientation: a matter of how you look at it: graphical screen design for visually impaired users*. (IPO rapport; Vol. 951). Instituut voor Perceptie Onderzoek (IPO).

Document status and date:
Published: 25/11/1993

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 20. May. 2025

Institute for Perception Research
P.O. Box 513 - 5600 MB Eindhoven

25.11.1993

Rapport no. 951

Screen orientation:
A matter of how you look at it.
Graphical Screen Design for
Visually Impaired Users

J.J. Beumer, P.B. Klimbie,
L.H.D. Poll and R.P. Waterham

Voorwoord

Cognitiewetenschap
In 1989 ben ik gestart met de studie Cognitiewetenschap. Verschillende vragen binnen de cognitiewetenschap spraken mij aan: Hoe kunnen menselijke denkprocessen gebruikt worden om computer intelligenter te maken?
Hoe kunnen computers beter toegankelijk gemaakt worden voor mensen?

Het probleem
Al snel liep ik echter tegen een groot probleem aan bij het volgen van de studie. Door mijn visuele handicap was ik genoodzaakt om met bepaalde hulpmiddelen het computerscherm uit te lezen. Hiervoor zijn speciale hulpmiddelen in de handel, zoals een spraakmodule die op commando van de gebruiker het scherm voorleest, of een brailleleesregel die de tekst van het scherm weergeeft op een metalen plaat in brailleletters. Dit zijn fantastische hulpmiddelen, maar al snel merkte ik een grote tekortkoming van deze aanpassingen: als de computer de letters op het scherm tekent (grafische mode), dan begrijpen deze hulpmiddelen de letters op het scherm niet meer.
Dit komt omdat de spraakmodules en brailleleesregels gebruik maken van het feit, dat iedere letter in de computer een nummer heeft (ASCCI codes). De brailleleesregel leest dan deze ASCII codes en weet welke letters hij naar de brailleleesregel moet sturen. De grafische mode maakt echter geen gebruik van deze ASCCI codes, maar gebruikt verzamelingen puntjes (pixels). Zo kun je een a op heel veel verschillende manieren schrijven (cursief, vet, klein of groot etcetera).

De oplossing
De oplossing van het probleem zou dan als volgt moeten zijn: de pixels moeten herkend worden als letters en vertaald worden naar ASCCI codes (Optical Character Recognition ofwel OCR genoemd). Daarna kunnen deze letters (ASCCI codes) op de oude manier naar de brailleleesregel gestuurd worden.
Maar de vertaling van de pixels naar ASCCI codes is niet de oplossing voor alle problemen die rijzen wanneer en visueel gehandicapte met een grafisch scherm wordt geconfronteerd. Op een grafisch scherm kan namelijk veel meer informatie worden gezet zonder dat de informatie onoverzichtelijk wordt, omdat je verschillende lettertypes en kaders kunt gebruiken om informatie van elkaar te onderscheiden. Dit is voor een visueel gehandicapte een probleem, omdat hij geen overzicht

heeft over de gebruikte kaders en lettertypes en daarom moete heeft om verschillende soorten informatie van elkaar te onderscheiden. .
Er waren twee redenen waarom ik een bijdrage wilde leveren om deze problemen op te lossen. Enerzijds omdat ik steeds vaker met het probleem geconfronteerd werd dat ik bepaalde programma's of delen van programma's niet kon lezen. Anderzijds omdat de studie Cognitiewetenschap zich bezig houdt met de toegankelijkheid van computers voor mensen.

Het project
Ik was niet de enige die het probleem van de grafische schermen voor visueel gehandicapten onderkende. Er bleek een E.g. project te zijn die het probleem van de grafische schermen voor visueel gehandicapte mensen moest oplossen.
Na enig speurwerk bleek op het Instituut voor Perceptie Onderzoek in Eindhoven een afdeling 'Communicatiehulpmiddelen Gehandicapten' te bestaan. Binnen deze afdeling werd deelgenomen aan bovengenoemd E.G. project, die VISA (Video Information and Signal Analysis) werd genoemd.
De partners in het VISA project zijn de universiteit van Nottingham, een engels bedrijf 'Sensory Vision aid', Frank audio data (een duits bedrijf die hulpmiddelen levert voor blinden en slechtzienden) en het IPO. De universiteit van Nottingham verzorgt de OCR, Frank Audio Data verzorgt de hardware (de spraakmodule en de aansturing van de spraakmodule), 'Sensory Vision Aid' zal het systeem later evalueren, en het IPO heeft verschillende taken.
De eerste taak van het IPO is gebruikersacties te voorspellen met als doel om de OCR sneller te doen verlopen. Dit zal ik verder niet uitleggen. De tweede taak is het herontwerpen van windowschermen zodat gebruikers beter met de grafische schermen kunnen werken, en ten derde het coördineren van de evaluatie van het systeem.

Mijn opdracht
Ik heb mij bezig gehouden met de tweede taak van het IPO. Om er achter te komen hoe windowschermen er uit moeten zien voor visueel gehandicapten, heb ik eerst een literatuur onderzoek gedaan en daarna heb ik drie gebruikersexperimenten afgenomen.

De tijdsbesteding

Ik heb 30 weken in totaal over mijn stage gedaan en 8 weken in totaal over het schrijven van mijn artikel. Ik heb 4 dagen in de week gewerkt, en de onderstaande aantallen weken aan de diverse fasen besteed:

/5/ Literatuuronderzoek. 17-8-92 t/m 18-9-992
/2/ voorbereiden eerste experiment. 21-9-92 t/m 2-10-92
/2/ afnemen eerste experiment. 5-10-92 t/m 16-10-92
/2/ analyse eerste experiment 19-10-92 t/m 30-10-92
/2/ voorbereiden tweede experiment 2-11-92 t/m 13-11-92
/1/ afnemen tweede experiment.16-11-92 t/m 20-11-92
/2/ analyse tweede experiment.23-11-02 t/m 4-12-02
/2/ voorbereiden experiment 7-12-92 t/m 18-12-92
/2/ afnemen derde experiment. 21-12-92 t/m 8-1-93
/4/ analyse derde experiment. 11-1-93 t/m 25 1 93
/6/ verdere uitwerking van de experimenten en het bundelen van de resultaten. 25-1-93 t/m 15 -5 93.
/8/ schijven van het artikel. 19-5-93 t/m 7-7-93

Indeling van het artikel

Het artikel bestaat uit een introductie ('INTRODUCTION'), beschrijving van het eerste experiment ('EXPERIMENT 1'), beschrijving van het tweede experiment ('EXPERIMENT 2') beschrijving van het derde experiment ('EXPERIMENT 3') en een discussie ('GENERAL DISCUSSION').

In de introductie schets ik het probleem (grafische schermen zijn nauwelijks toegankelijk voor visueel gehandicapten), de oplossingen die hiervoor reeds zijn getroffen om het probleem op te lossen (software bridges) en de aanpak van ons project. Ook schets ik de aanpak die ik heb gekozen om een deel van het probleem - het overzichtelijk maken van window schermen voor visueel gehandicapten - op wil gaan lossen.

In het eerste experiment beschrijf ik - aan de hand van enkele metingen die gemaakt zijn in dit experiment en aan de hand van een analyse die het gedrag beschrijft van de proefpersonen hoe ziende mensen taken uitvoeren in een Windows 3.1. omgeving. Ook geef ik aan hoe de resultaten van dit experiment kunnen worden gebruikt in het tweede experiment.

In het tweede experiment beschrijf ik - weer aan de hand van enkele metingen en een GOMS analyse - hoe visueel gehandicapten zoeken in een Window omgeving waarvan de schermen door mij ontworpen zijn. Wel gebruiken zij hier geen computer, maar stellen vragen aan iemand die de schermen voorleest.

In het derde experiment wordt de zoekstrategie van visueel gehandicapten in kaart gebracht wanneer ze in een Window omgeving taken uitvoeren (wederom gebruik ik hierbij enkele metingen en een GOMS analyse). Zij maken gebruik van de

hardware van Frank Audio Data (een spraakmodule) en twee verschillende schermindelingen.
In de 'general discussion' geef ik een samenvatting van de experimenten en breng de experimenten met elkaar in verband Ook geef ik aan welke aspecten verder onderzoek verlangen.

Met dank aan...
Graag zou ik Tony Jameson willen bedanken voor de vele uren die hij besteed heeft om mijn stage te begeleiden en mijn scriptie te herzien. Ook bedank ik Ronald Waterham en Leo Poll, die altijd ter plaatse commentaar wilden geven over mijn stage en scriptie. Een sleutelpositie had voor mij ook Balthasar Klimbie, die alle software en hardware die ik voor de experimenten nodig had heeft ontworpen.
Verder zijn er vele IPO medewerkers en medestudenten geweest, die mij fantastisch hebben geholpen met het maken van plaatjes, het zoeken van literatuur en het voorlezen daarvan. Ook allemaal verschrikkelijk bedankt.

1. Introduction.

Graphical User Interfaces and Visually Impaired Users

The rapid growth of computer use is a positive development for visually impaired people (Edwards 1992). The reason is that visually impaired people (called VIPs) usually have a job related to a fixed work place, because they would have difficulties with jobs that are related to different locations (like the work of a postman). Adaptations with speech synthesizers (A. Arat et al. 1992) (Blenchorn 1992) and braille displays allow them to work with the same software used by sighted people, if the software consists of standard application programs (Weber 1992) (Edwards 1990).

But in the last few years many applications have been made that make use of a graphical user interface (GUI). It has been discovered that these programs are more easy and efficient to work with. This creates a huge problem for VIPs, because their computer adaptations (braille display and/or a sound synthesizer) can only work in a non-graphical mode (Edwards 1992). When an application is introduced in the working environment of visually handicapped people, they cannot work with it. In the worst case, this means that they cannot do their jobs and have to be discharged.

Two Approaches

Much work has been done to overcome this problem. for instance a program that makes the graphical modes of the word processor Word accessible (Weber 1992). We can distiquish two possible solutions: software bridges and hardware adaptations.

A software bridge is a program that runs on the same computer as the original software with the graphical interface. On a certain level in the computer the graphical information is represented as ASCCI codes. On this level, the software bridges recognize these values and use them for the output.

Over the last few years, a couple of these software bridges have entered the computer market (Edwards 1992). There is a software bridge for the Apple Mackintosh: the soundtrack (Edwards 1992). It is an experimental auditory word processor for the Mackintosh, which uses speech and non-speech sounds to make most conventional mouse interactions accessible to blind users. Apart from this, there are software bridges for DOS computers, An example is Slimware, a software bridge for MS Windows.

The second solution is a hardware adaptation. This option does not interrupt a system which contains the original software with a graphical user interface. On an abstract level, it can be seen as a robot: the robot looks at the screen, recognizes the information presented on the screen, and types this information with mechanical iron hands in the computer (see the picture on the front page). In reality this means that the video signal is used to recognize letters (optical character recognition) by means of an external computer.

The advantages of using a hardware adaptation

The hardware solution has many advantages. The first is that the software which should be written to recognize the items on screen, can use the fact that most graphical user interfaces are basically the same. The GUI environment mostly consists of windows, containing icons, a closebutton, pull down menus and pop up menus, etcetera (these are called window objects and are described in detail further on)). Therefore the software to recognize these window objects should be written only once, while the OCR (optical character recognition) can complete the recognition of the whole graphical screen (for instance the OCR should recognize the name of items in a pull down menu). Now all graphical interfaces can be analyzed by using the software called above, only new window objects in new programs have to be defined.

The second advantage is that no interaction effects occur between the OCR software and the original software. Every programmer will know that, when several programs are running simultaneously, they could interfere with each other.

The third advantage is that less conventions are needed to make the original program applicable to the existing bridge. These conventions are important because otherwise another adaptation of the bridge software would be required for every new program or new version of the program (Blenkhorn 1992). An example of such a convention is for instance, , when running a new version of the original program together with the bridge software, the following convention should be taken account of: "Both in the old version of the original program and in the new version, the same processor should be used." However, this convention is not needed when using the hardware adaptation.

The VISA Project

In the TIDE (Technological Initiative for Disabled and Elderly) project called VISA (Video Information and Signal Analysis), which was started in January 1992, we opted for the hardware solution. The aim of the project is to make an entrance for VIPs to a Window environment. In this way VIPs will again be able to use programs which were no longer accessible to them.

The hardware we use is a fast 486 personal computer with

MS DOS, a mouse and a TASO screen of Frank Audio Data (see below) and we will call this the VISA computer, or he VISA-comp system.

The TASO Screen

The TASO screen (see figure 1) is, combined with a speech module, a screen reader which can transfer non-graphical text into speech. All information on the screen can be read in the following way:

The TASO screen contains two sliders. They are called TASOs: Tactile Acoustic Screen Orientation. The TASOs are located on a modified keyboard. One part of the keyboard is located in front of the normal keyboard and contains the horizontal slider (explained below), the other part is situated on the left or right side of the ordinary keyboard. It contains the vertical slider (explained below), two buttons - one to change the speech volume and another to change the rate of speech - and a keyboard that is similar to a telephone keypad. This keyboard can be used to change commands. For instance, the user can specify if he wants the current line to be automatically repeated or if capital and small letters are to be contrasted through the speech signal (Edwards 1989). The speech synthesizer is a little box that can be placed anywhere.

The sliders can be used to give the position on screen that has to be read. They are square and can be pushed and moved. One slider can be moved vertically (the vertical slider) and gives the line to be read (called 'line selection'). By slightly moving the vertical slider down, the next line can be read. The other slider can be moved horizontally, and gives the position in the current line (called 'column selection'). If the slider is moved one position to the right, the next letter can be read.

The information can be read by pushing the sliders. If the slider is pushed, the system starts reading directly, and it immediately stops reading as soon as the push is released. When pushing the vertical slider, the line which is positioned is read either partially or as a whole, according to the commands that are used to control the speech, but each word will be pronounced in its entirety. When the horizontal slider is pushed, the system spells the words of the current line.

Various sounds are used to distinguish the different moves of the sliders. When moving the vertical slider, a high beep is given if a line is empty c.q. contains no information. A very low sound beep (a tone four octaves lower than the empty line tone) will be given when the cursor is presented on the line currently positioned. Five different tones (tones within the range of one octave; the highest tone is two octaves lower than the empty line tone and the lowest tone is one octave higher than the cursor position tone) will be heard in sequence when moving the vertical slider five positions (up or down). After five lines, this sequence of five tones will be repeated in the same order. These five different tones will only be heard in sequence if some information on these lines is presented. When an empty line appears, a high beep is given. When a new line with information is reached again, the sequence of the five tones is repeated from the beginning of the sequence of tones. According to the designers, this would enable users to move over several lines at a time without counting, because they would know which sounds they should hear.

The horizontal slider also makes use of these musical sounds. It uses an octave for giving information about an item presented on a position - if the position contains an item. The highest tone of this octave is one octave lower than the empty line beep. The lowest tone of this octave is two octaves higher than the cursor position beep. The tones of the octave used for the horizontal slider make a distinction between normal letters, capital letters, numbers and special characters, each with their own tone within this octave. The horizontal slider uses the same high and low beeps as the vertical slider to indicate whether a position contains information (high) and the cursor is situated on it (low).

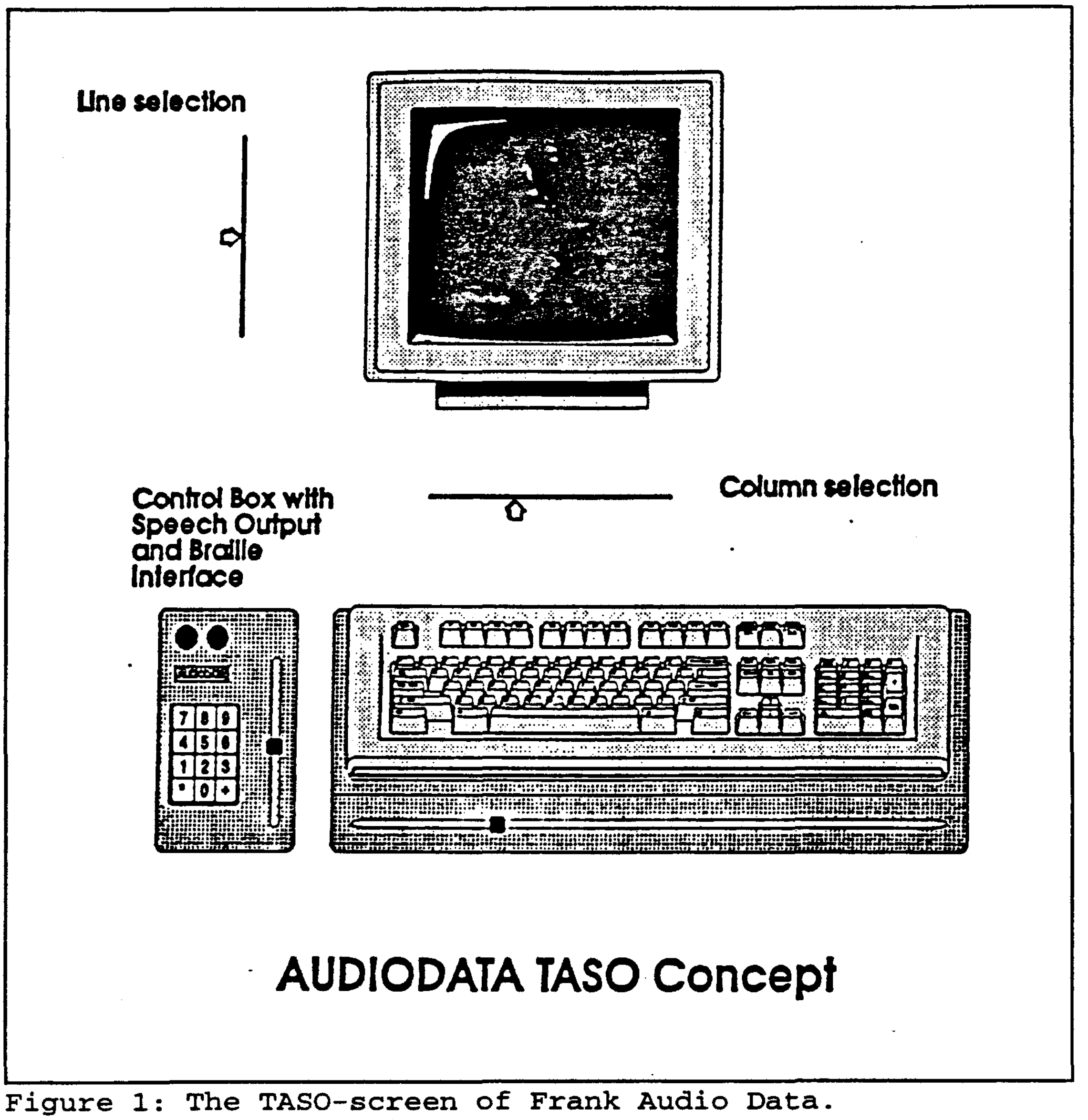


Figure 1: The TASO-screen of Frank Audio Data.

The VIPs' Way of Looking at the World

Now that the OCR software has been built and the screen reader of Frank Audio Data has been explained, it is necessary to consider how information should be presented on screen to VIPs. This is not obvious, because we believe that VIPs `look at' the screen in another way than sighted people do.

As visually disabled people perceive the world only by means of hearing, smelling and with haptical perception, they 'look at' the world differently. In contrast, sighted people gather most information by seeing and the other senses are of minor importance (Knaapen 1986). As a result, visually impaired people have another conception of the second and third dimensions. For example, when a blind person describes a bicycle, he says: 'It is a wheel with a handlebar and a bar connects this wheel to another wheel, and on this bar there is a saddle with yet another bar with a pedal and another pedal beside it.' Evidently he has a partial image of the bicycle, so that the wheel will disappear as soon as he stops touching it (Knaapen 1986). In contrast, the description of a well sighted child will only mention the most important parts of the bicycle (not the different bars which connect the different parts of the bicycle) and the child will refer to them as a whole. It will say that the bicycle has two wheels, and not, like the blind child, 'one wheel and another wheel'.

Another argument for giving a different screen design is that hearing gives people a different kind of information than seeing does. Hearing is in time and over space while seeing is over time and in space (Gaver 1986). This means that visual objects can only be seen when one looks at them, but sounds can be heard all around. Another aspect is that one can look at an object for some time, while sounds can only be heard at the moment they are produced.

The experiments

The present research involved three experiments intended to shed light on the following question: "What general principles of graphical screen design are best suited to the way in which VIPs perceive the world?" We are trying to shed light on this very broad question by investigating certain specific tasks within a specific system ('Windows' version 3.1.).

First goal of the experiments

The first goal of the experiments is to find out how people who work in a Window environment get to the application they want to reach. To achieve this, we will investigate what information on screen attracts attention, what information on screen is used and how frequently every part of information is used.

Subjects performed four queries which were designed to find applications (application search tasks), like 'Go to Word

Perfect and when you are sure you have found it, return to the program manager.' All queries started in the program manager.

Second goal of the experiments

The second goal of the experiments is to find out how people search for information to manipulate something in the window environment, for example, when they search for the possibility to save the settings on exit. In order to achieve this goal, it is important to know which information is used the most and which the least (i.e. the importance of the information). Four queries were designed to reach this goal.

The second goal is broader than the first, because it considers the Window environment as a whole and not just at the execution of a simple query as the first goal did.

Constraints on designing queries

In our opinion it would be wise to select items and applications on several layers of the selection tree and to observe how people find the information. We wanted to see how subjects search for information, either when they make little or much use of their working memory. Therefore we used several layers in the selection tree, as the risk of overloading the working memory increases when more layers in the selection tree are passed (Paap et al. 1988).

We have to make sure that items and applications are presented to the subject in a random order, to control for the influence of the previous search on subjects while they are searching for a new item or application.

We also added some queries which have unexpected screen changes (e.g. when the game 'back gammon' is left, the system asks in a dialog box 'Do you really want to quit?' The subjects had to answer this question in order to leave the game.) We wanted to see how the subjects react to this unexpected screen change. We expected that sighted people react in a different way to unexpected screen changes than VIPs.

Finally we have to take into account seriously that both norma sighted people and VIPs may react differently to information presented in a different way (e.g., the difference between a pull-down menu and a dialog box).

In summary, we will take into account the considerations mentioned above by presenting the greatest possible variety of menu items and applications on different layers of the selection tree and by using all different kinds of windows.

Overview of Experiments

In the last few paragraphs of this introduction, we will give a short overview of the experiments. We will briefly describe how the tests are performed and what is tested; what the aims of each experiment are, how the experiments are performed. We will also describe how the experiments are related and how the results will be used in the VISA project.

Experiment 1

The first experiment aims at discovering how the information should be presented on screen to the VIPs by observing sighted people who have no experience with GUI (Graphical User Interface) environments. We used novice users instead of experienced GUI users for the reason that we have to use VIPS in our second and third experiments, and they have no experiences with a GUI environment either. As we want to compare the search strategies of sighted people and visually impaired people (see below), all should be novice GUI users.

After the experiment we will describe the information search strategies of the subjects with a GOMS analysis. The GOMS analysis makes use of the hypothesis that the user's cognitive structure consist of four components: a set of Goals, a set of Operators, a set of Methods, and a set of Selection rules. The goals define a state of affairs to be achieved and determine a set of possible methods by which it may be accomplished. The set of operators are used to define these goals. These operators are for example 'activate the close button'. The set of methods is not a separate step in the GOMS analyses, but a sequence of actions, to perform (part of) a task, for example find an application'. There are several possibilities to perform this action (methods). The selection rules are made to choose amongst competing methods that achieve a certain goal. An example of this GOMS analysis is given in appendix 4.

With this GOMS analysis we can describe more easily which information is more and which is less important in a certain situation. We will call these descriptions principles. These descriptions are extended by using literature, which give some information about screen design (Paap et al. 1988) (Marshall et al. 1987).

Experiment 2

In the second experiment we observe how VIPs work in a Window environment when information is offered on screen in a way designed to be especially helpful to VIPs and in accordance with the principles formulated after the first experiment. We also want to see if the VIPs can understand and work with the screen representation made after these principles.

Experiment 3

In the third experiment we will look if VIPs can work with a special screen design more easily than with a normal graphical screen design for a Window environment, by testing two different screen designs of a Window environment. One screen design gives the same spatial information as an ordinary graphical Window screen used by sighted people. The other gives information in accordance with some principles that have been made in the first and second experiment and it takes the interface of the TASO screen, the speech synthesizer and the perception of the world by the visually disabled into account.

Way of use in the VISA project

The results of these experiments show which information search strategies are used by VIPs who are working with the TASO screen in a 'normal' Window environment and in a Window environment which has a screen design specially designed for VIPs. Apart from this, it will give the most important issues which should be taken account of in the development of a screen design for the VISA-comp system that is better suited to VIPs.

2. Experiment 1

2.1. Introduction

Broad description of this experiment

The purpose of this experiment is to discover how GUI information should be presented to VIPs, as described in the previous chapter. To achieve this purpose we analyzed in this experiment how sighted people work in a Window environment. The subjects were novice users in relation to a GUI, but they had some experience with the use of computers.

Our analysis was performed in order to answer the following questions:

- What kind of information on the screen (window objects) is used? For instance, do the subjects first look at the application group icons, the menuline or the title bar[1].
- How frequently is every part of information used? For example, every time the subjects are searching for some kind of information they might look at the menuline and look at the option keys only when they cannot find the information there.
- How do the subjects perform specific tasks? For instance we are interested in a detailed analysis of the subjects perform the queries, and a GOMS analysis can help with this. An example of the use of a GOMS analysis to describe the information use strategies is given in appendix 4.
- How much time is needed to perform a certain task? This is of interest for a comparison with the results for VIPs in the later experiments.

Window objects used i the Windows 3.1 environment

Before we give a detailed analysis of this experiment, we have to explain the window objects and the windows used.

The objects of the Window 3.1 environment can be divided in four classes: Icons, Menus, Dialog boxes and Windows.

Icons are little graphical pictures with or without labels. An icon can be used to choose an application (called application icon') or some different applications (called 'application group icons'). Icons are grouped together on

---

[1] These terms will be explained below.

screen ('collection of icons').

We specify two different kinds of menus: the menuline and a pull down menu. Pull down menu's can show a list of actions or options ('called menu items'). These menu items are grouped by vertical lines. A pull down menu can appear after selecting an item in the menuline. The menuline consists of some menu items presented on one line.

A plain window consists of some basic information and some optional information. A window always contains a closebutton in the top-left corner and a caption on top of the window (called 'titlebar'). If the closebutton is chosen, the window disappears. A window can also contain a vertical scroll bar (positioned on the right side of the window) and a menuline (underneath the titlebar). A scroll bar is a graphical representation of the real size of the window. If the current window is to small to present all information of the window, the scroll bar gives the possibility to scroll up and down through all information presented in this window.

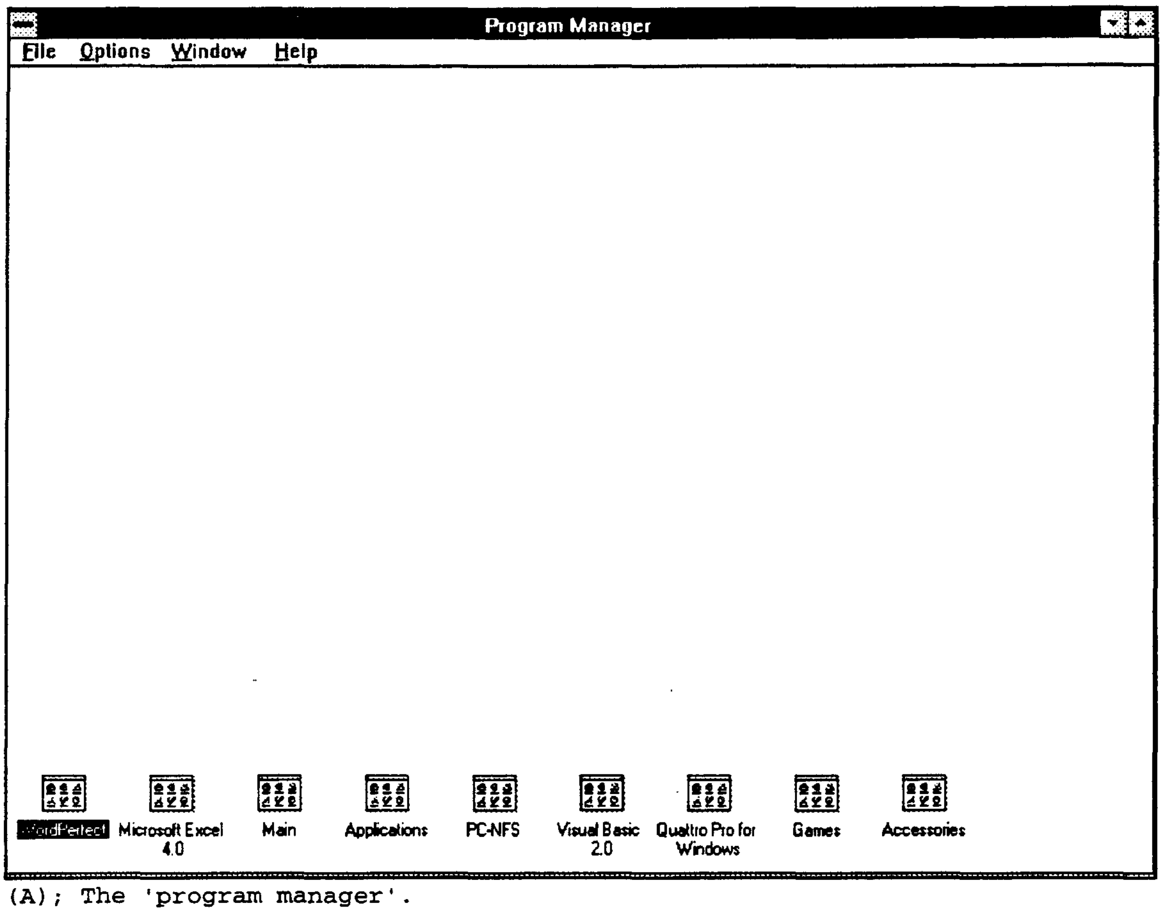
A static window can contain icons, text labels, empty and filled rectangles that contribute to the general function of the dialog box.

A special type of window is a dialog box. In fact it is a normal window that contains control tools. There are two types of dialog boxes: standard dialog boxes and captioned dialog boxes. A standard dialog box contains a message that prevents the user from interacting with other windows in a program. A captioned dialog box lets the user choose the window to work with.

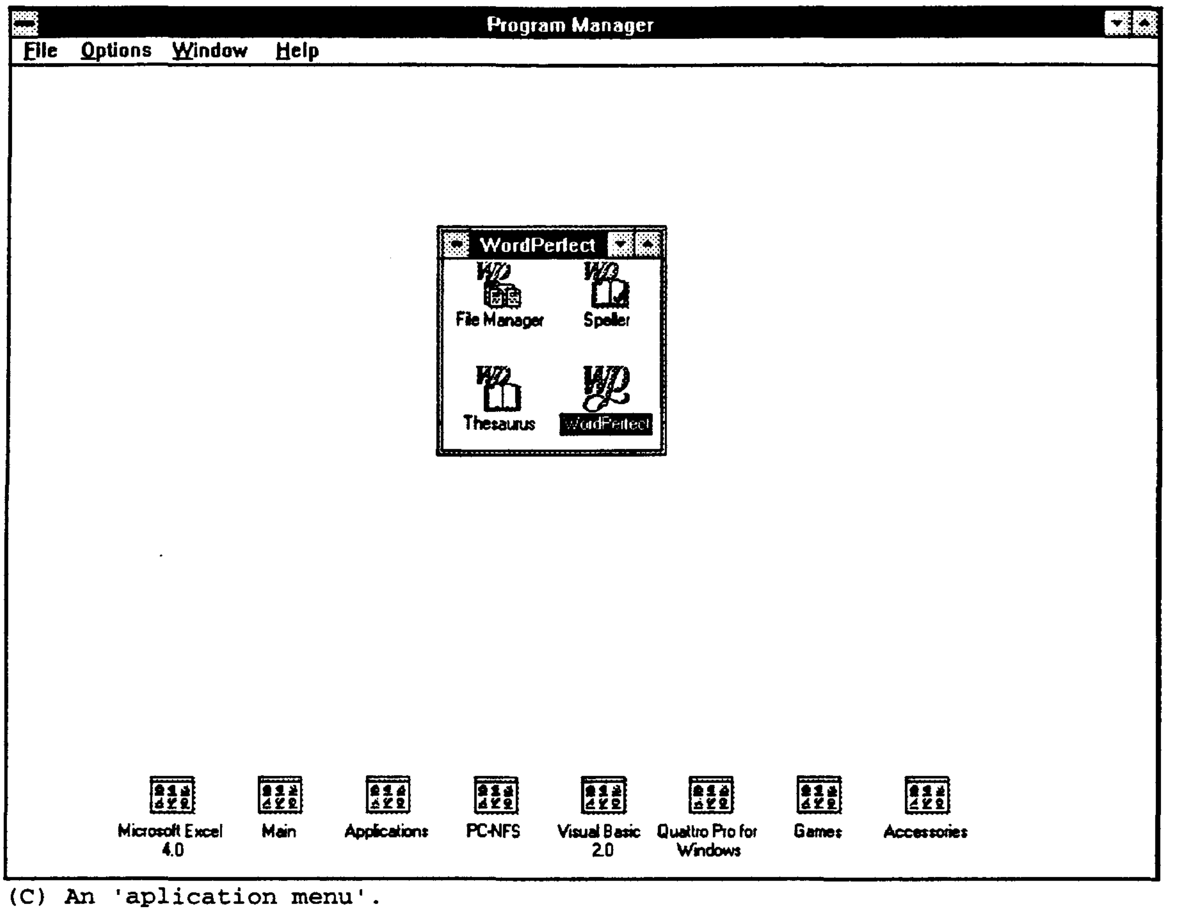
A dialog box can contain either different kinds of buttons (which can be chosen) or edit control. Edit control is a box that allows editing text.

Some graphical information is presented in figure 2.

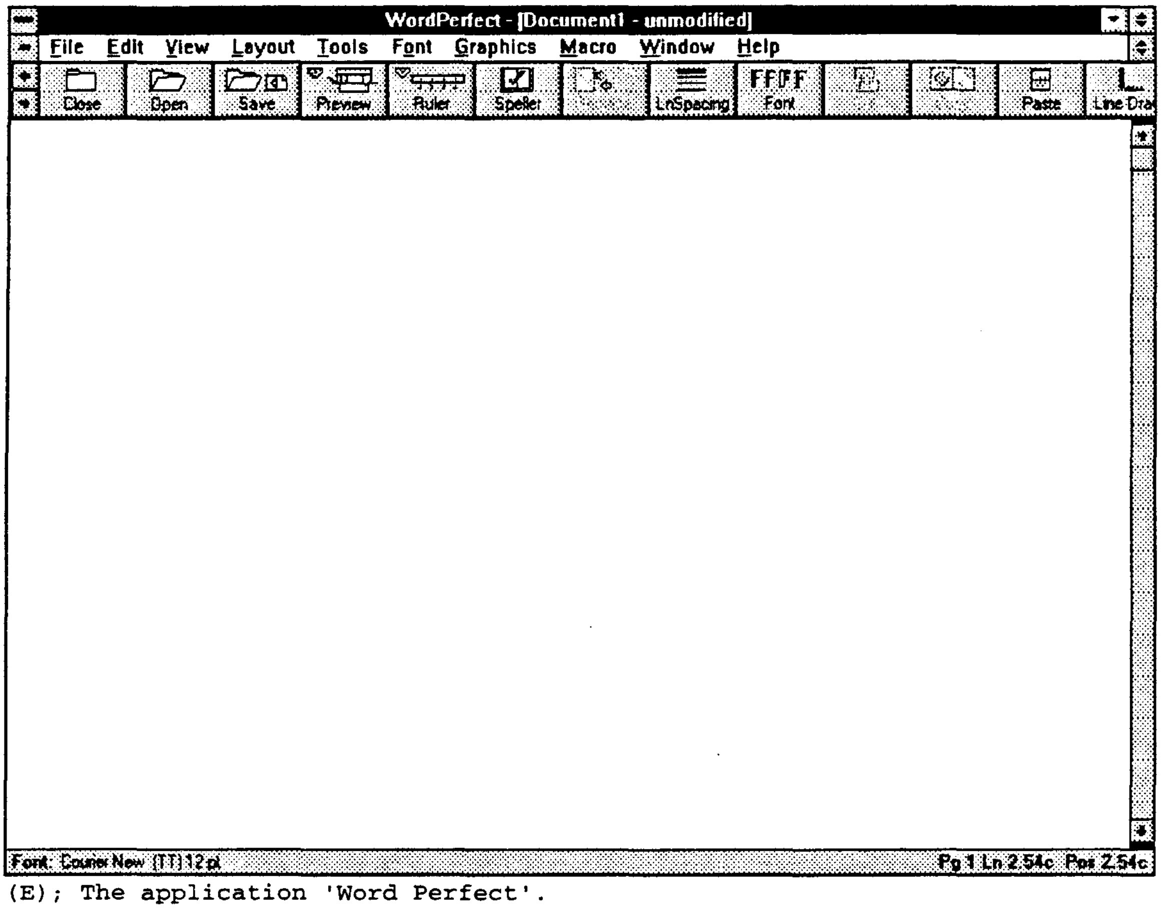
Figure 2: Window objects in the Windows 3.1 environment.
A: The program manager is a static window with application group icons.
B: The application menu is a static window which contains application icons
C: An application is a static window
D: A pull down menu
E: A captioned dialog box



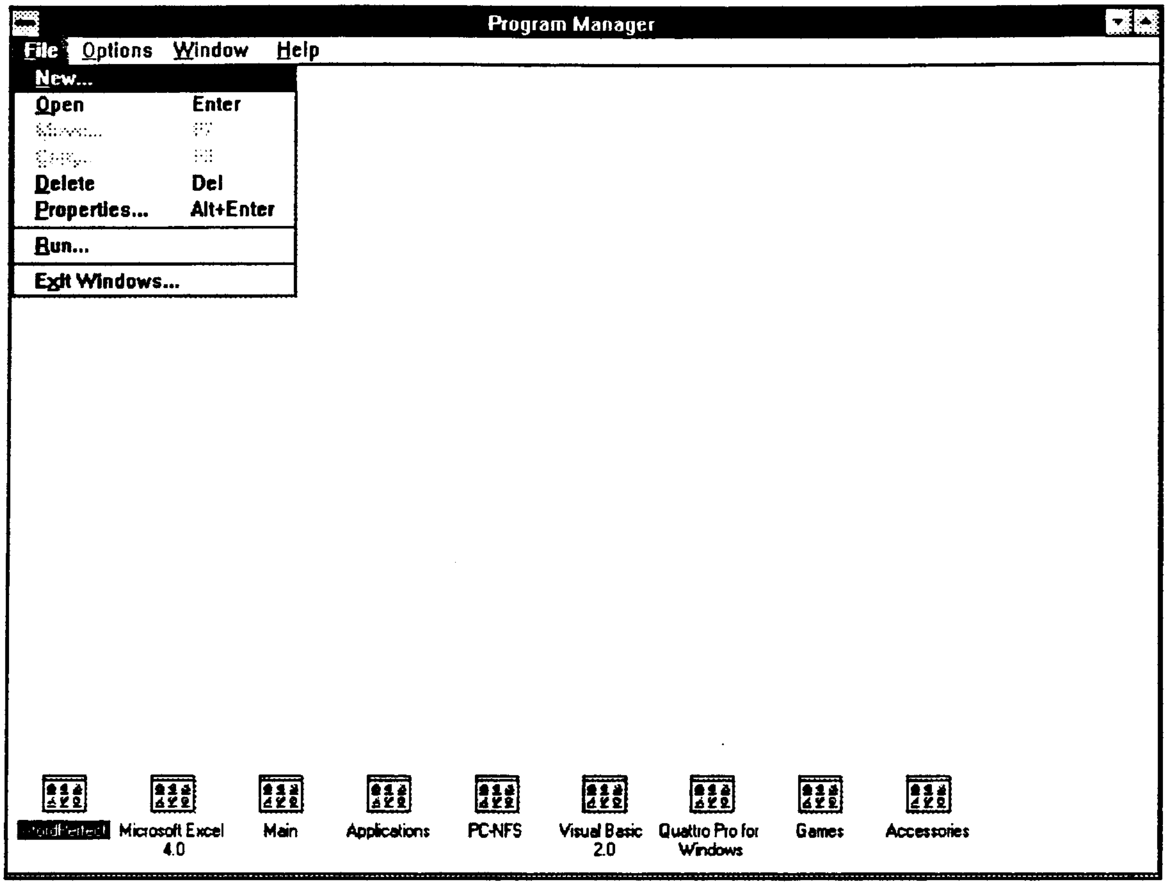
(A); The 'program manager'.



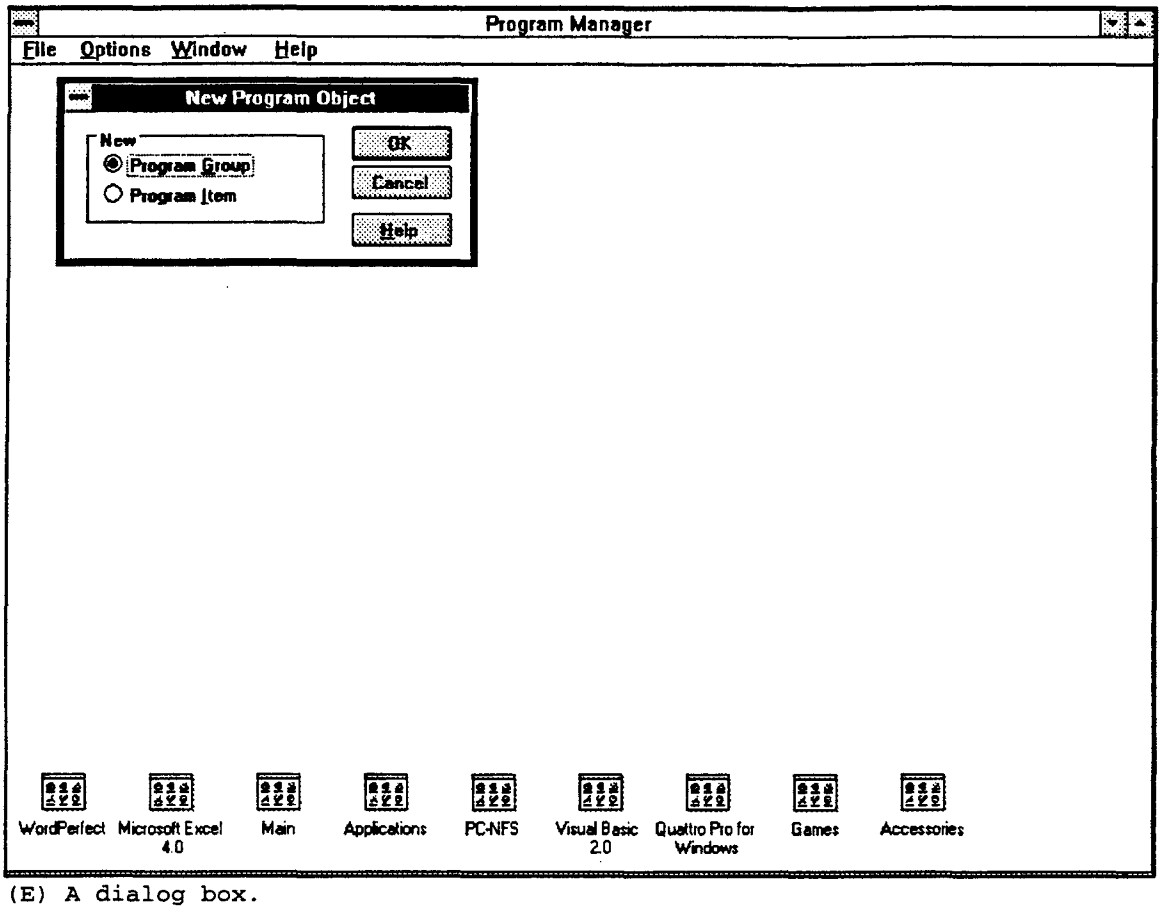
(C) An 'aplication menu'.



(E); The application 'Word Perfect'.



(D); A pull down menu.



(E) A dialog box.

2.2. Methods

Subjects

We used ten subjects who were used to working with a computer, but had no experience with a GUI (Graphical User Interface) environment. The age of the subjects ranges from 22 to 40. They were not paid for participating in the experiment, because they were all acquaintances. Two of the ten subjects were used for the pilot experiments, to see if all materials and software were working and if the queries were understandable to the subjects.

Hardware and software

The experiment was video taped. We used a 386 personal computer with MS Windows 3.1.

We started with a trial session which consisted of two queries. The first query was not related to Windows, the second query was.

The actual test contained nine queries. Four queries were application search tasks, which means that the subjects had to search for an application (e.g. 'go to Word Perfect'). One query was a filler task, so that people would not be influenced by the repetition of certain actions (repetitive actions can lead to a certain routine and this routine will influence the results of the experiment). The others were Option search tasks, queries related to the window environment (e.g. 'Search for the item 'save settings on exit'). The queries can be read in appendix 1, an option search task is shown in figure 3, an application search task is shown in figure 4.

The queries were performed in a randomized order. The randomization was done according to the Latin square with the constraints mentioned below.

The constraints which had to be taken account of are as follows:

- Begin with an application search task. In this way the subject will be motivated, because these queries are relatively easy.
- End with an application search task, for people like to end with an easy query.
- Two queries of the same type should not occur in succession (application search tasks or option search tasks).
- The filler task should be given in the middle of the queries.

Procedure

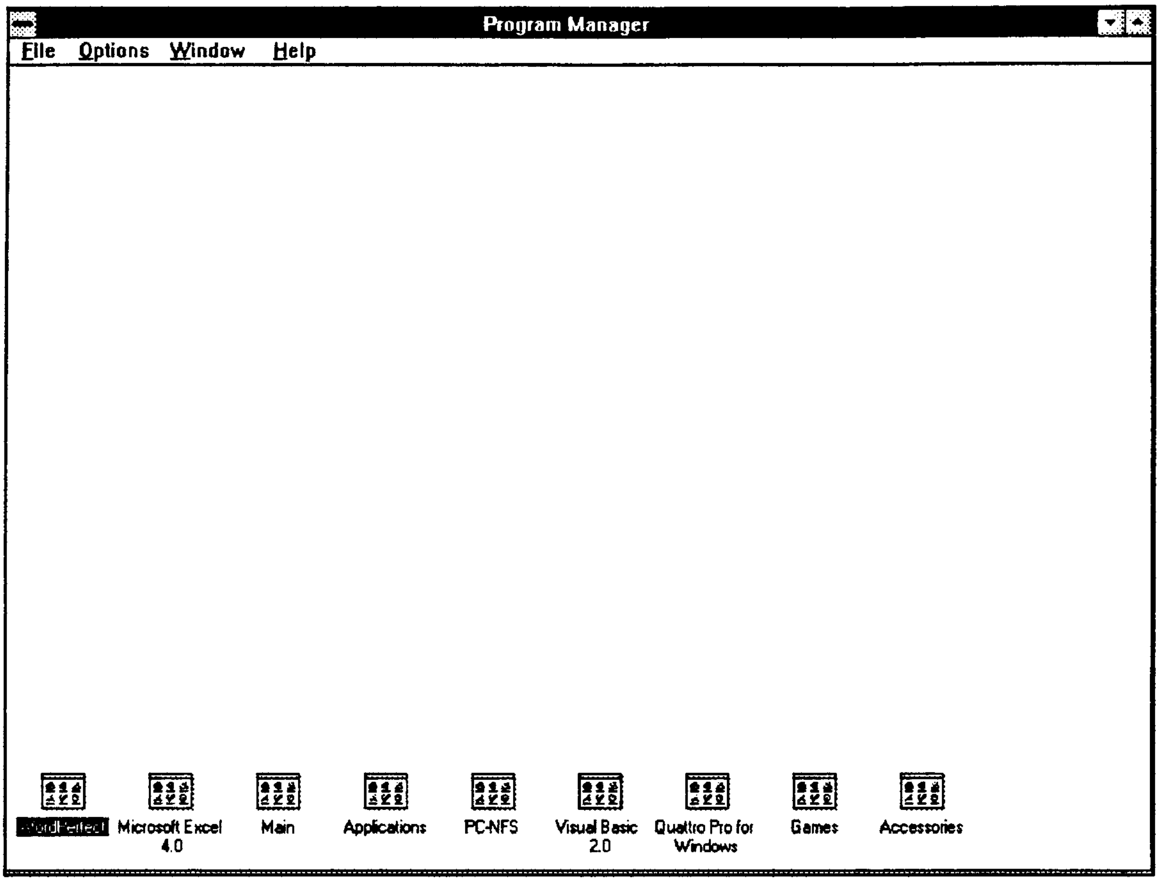
The experiment started with giving instructions for a Thinking aloud protocol. The subjects were stimulated to talk continuously, and we asked them what they thought when they were silent for a while. One query was given to train the Thinking aloud protocol. After this, the subjects had to perform another query related to Windows, to get used to the GUI environment (see appendix 1) and for further training with the Thinking aloud protocol.

After the trial session, the video tape was started and the actual test began. We used a video reorder because in this way we can analyze non-linguistic actions performed by the subjects.

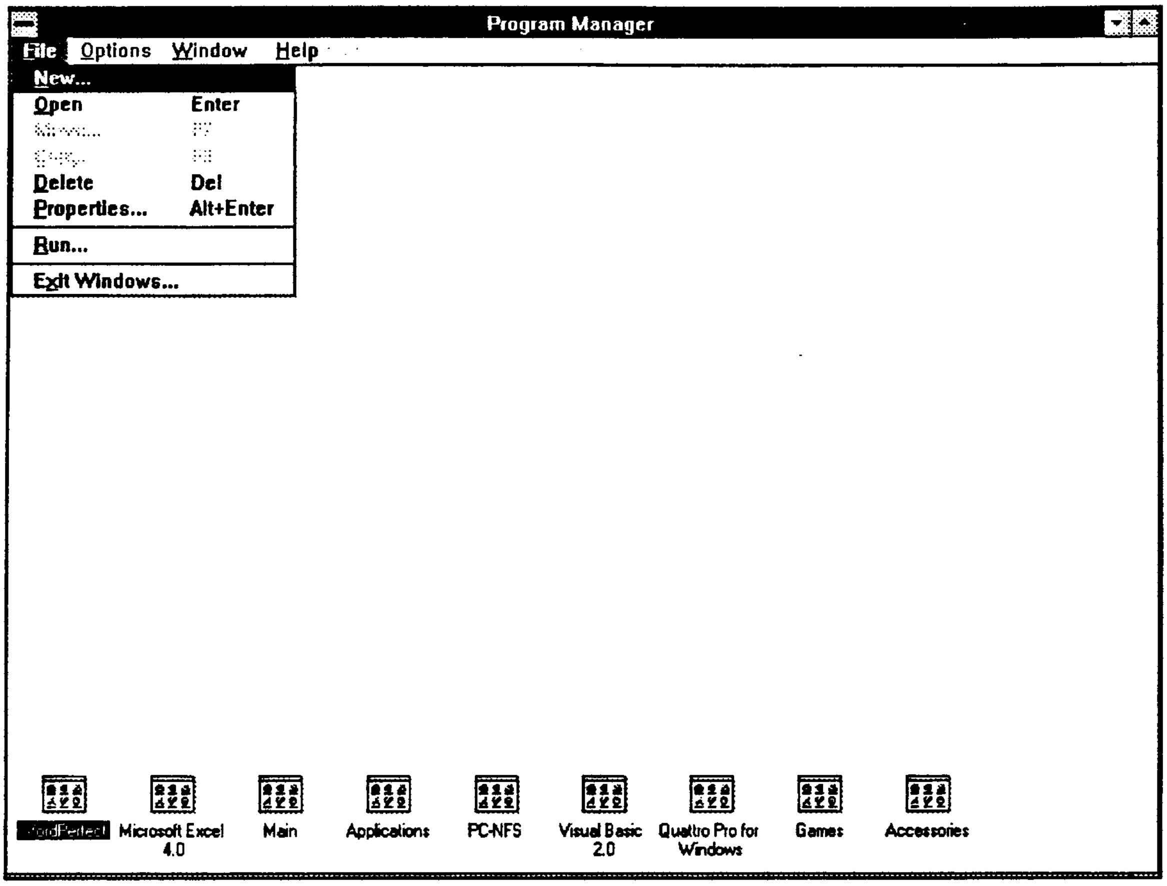
The queries were performed in accordance with the randomization described above. When the subjects stopped talking, they were stimulated to start talking again. They were asked, for example: 'What are you thinking of?'

After the experiment we had a short talk with the subjects. This talk was meant to easy their minds about their performances and to tell them about the purposes of the experiment.

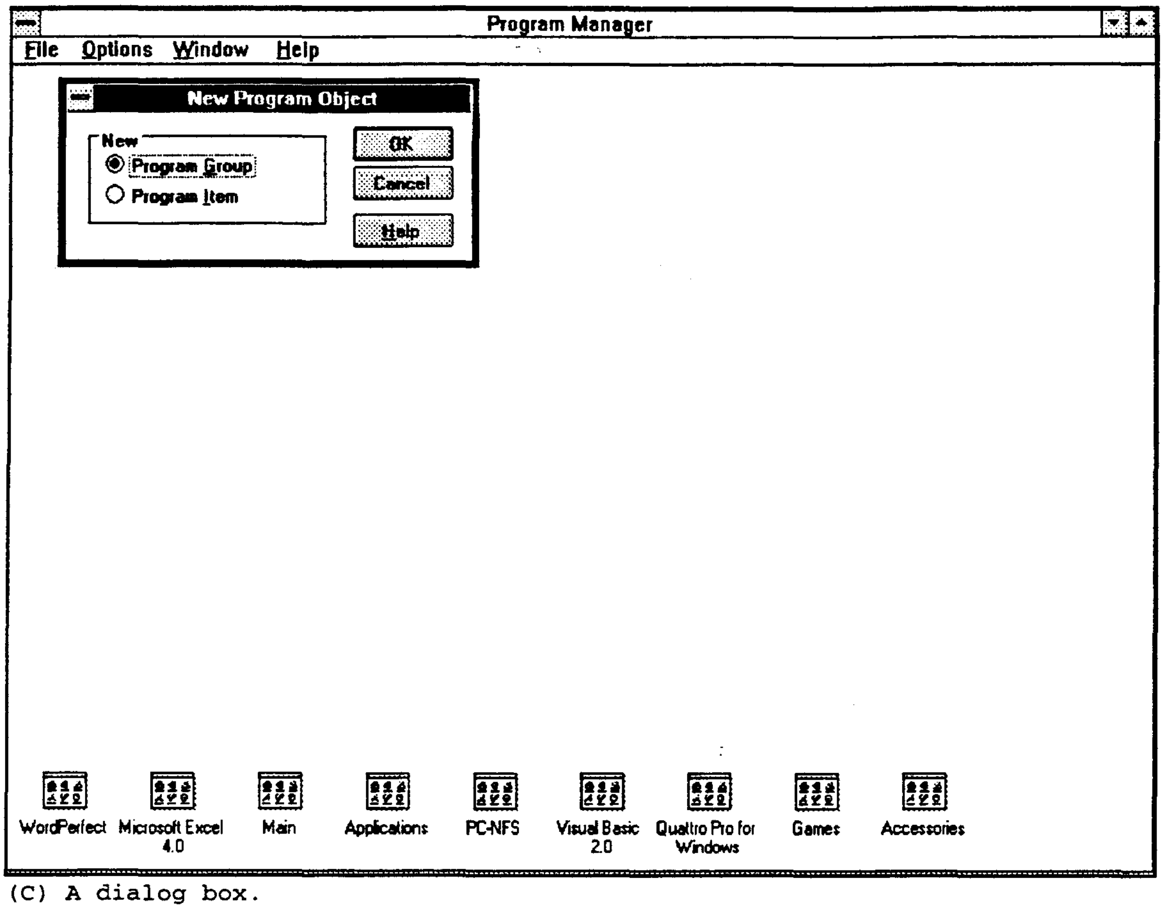
Figure 3: Windows in the Windows 3.1 environment.
The windows which should be passed when performing the first option search task.



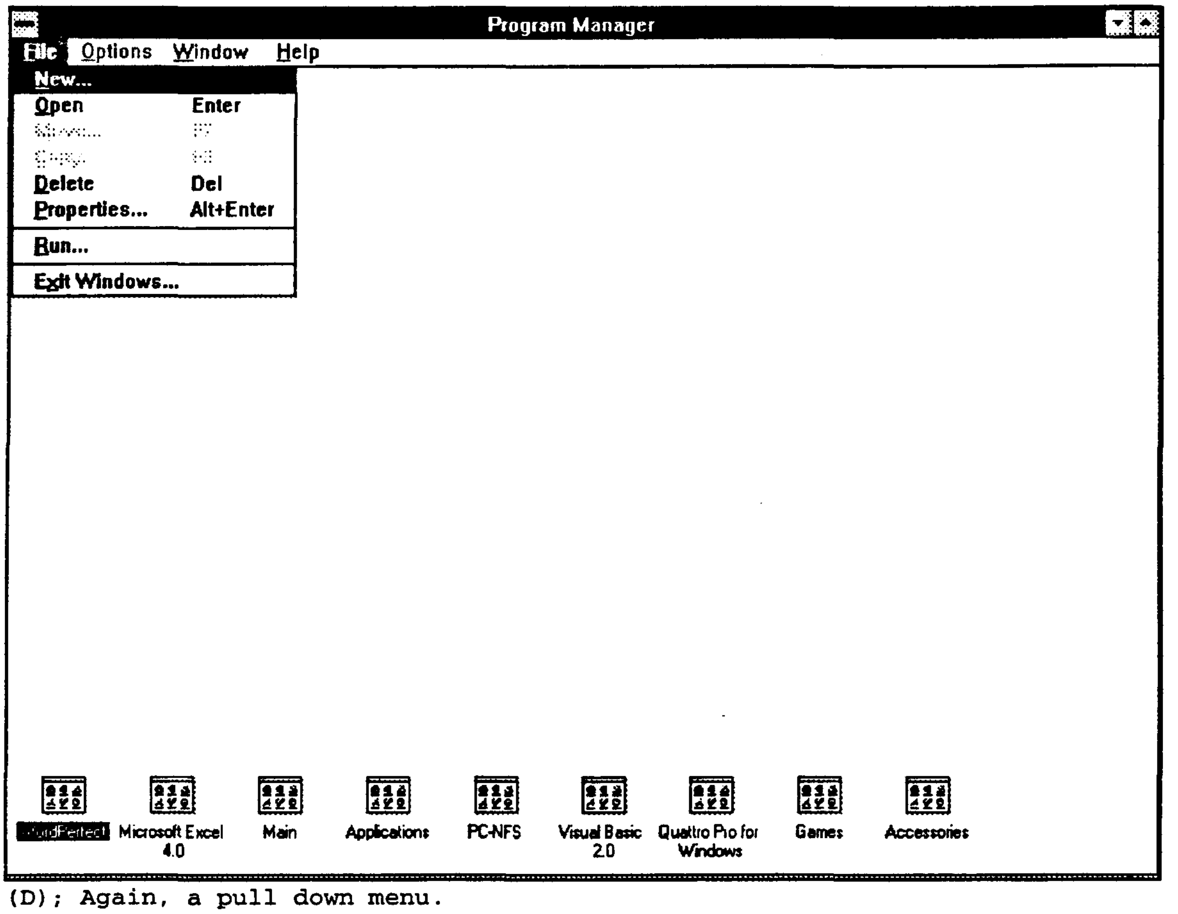
(A); The 'program manager'.



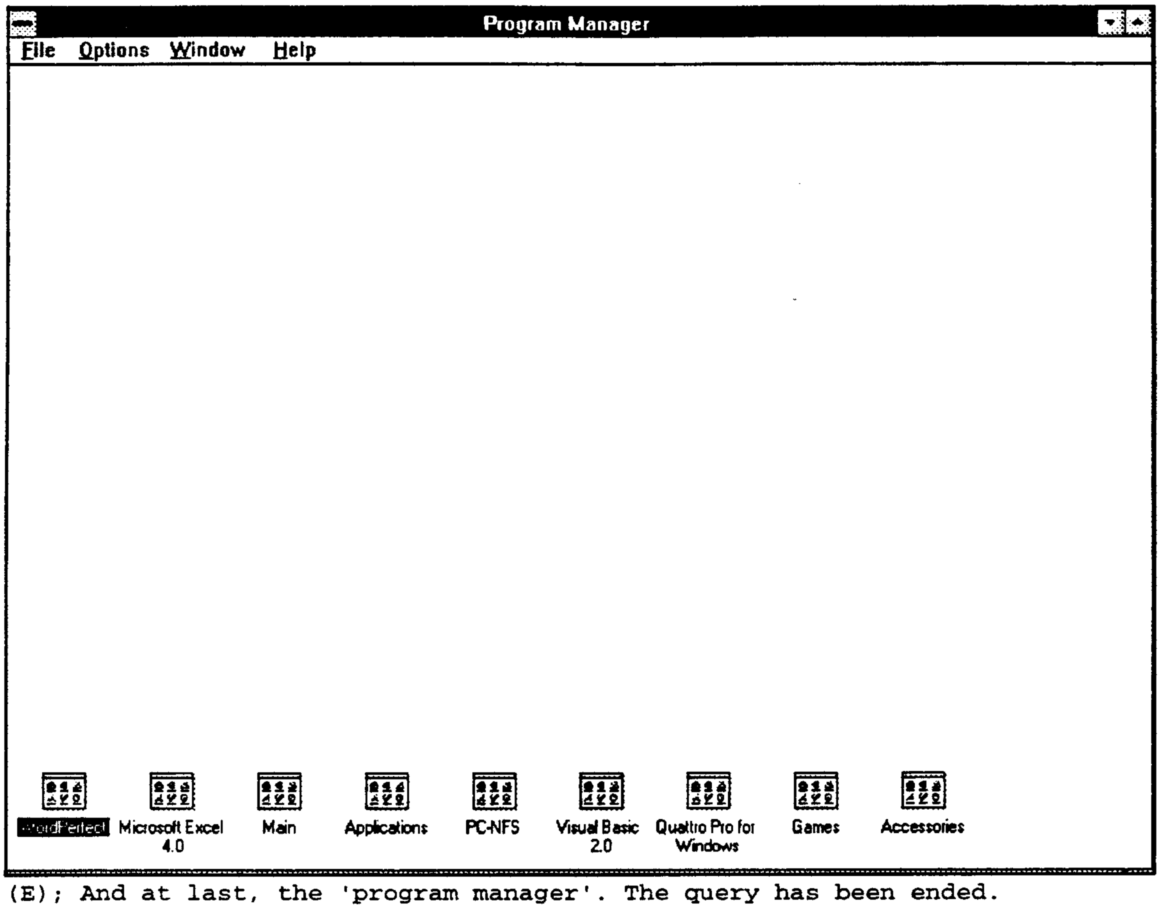
(B); A pull down menu.



(C) A dialog box.

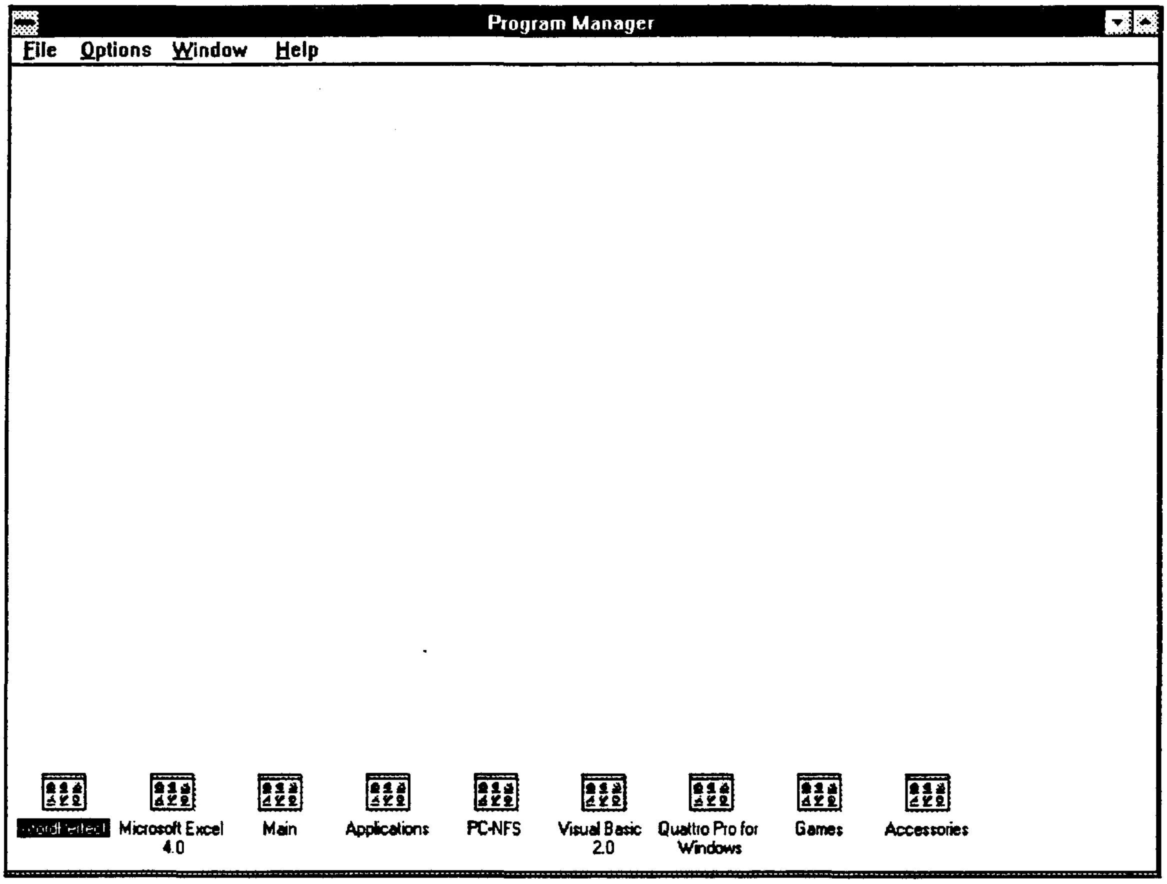


(D); Again, a pull down menu.

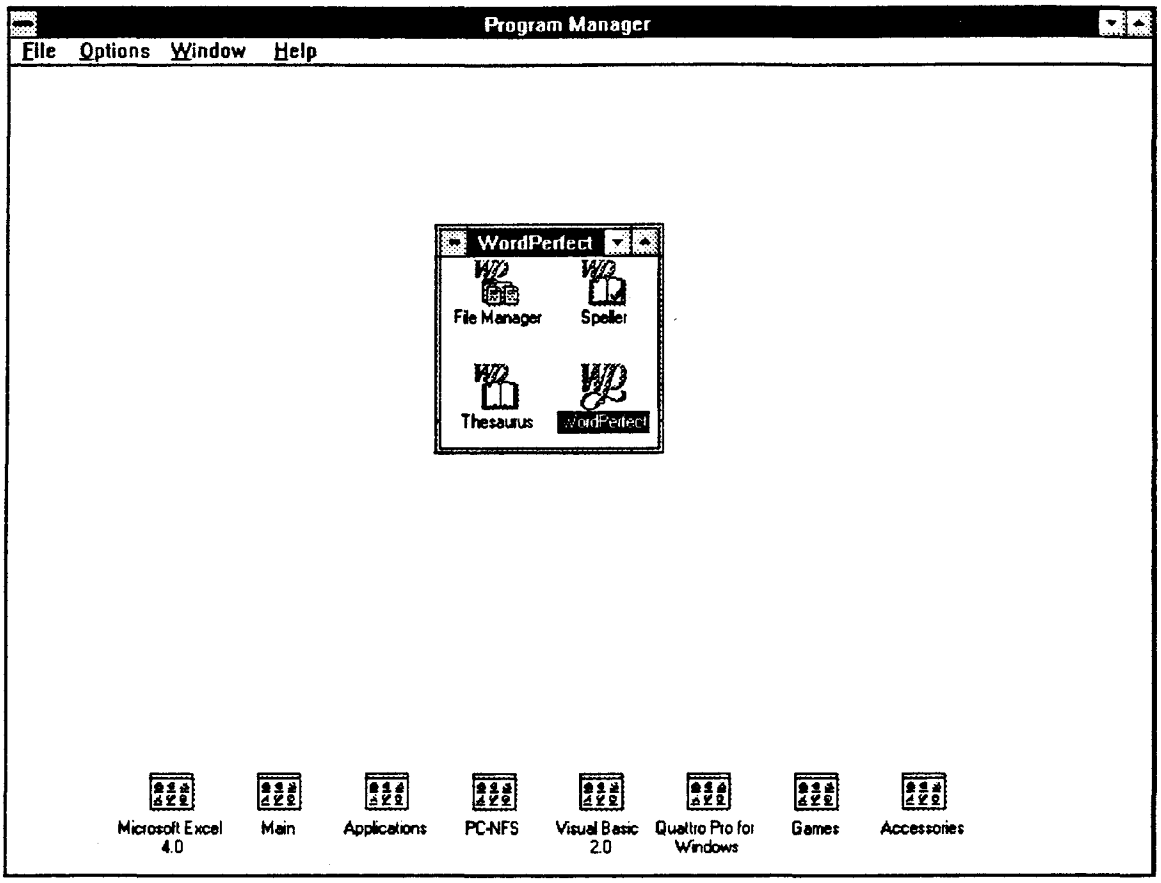


(E); And at last, the 'program manager'. The query has been ended.

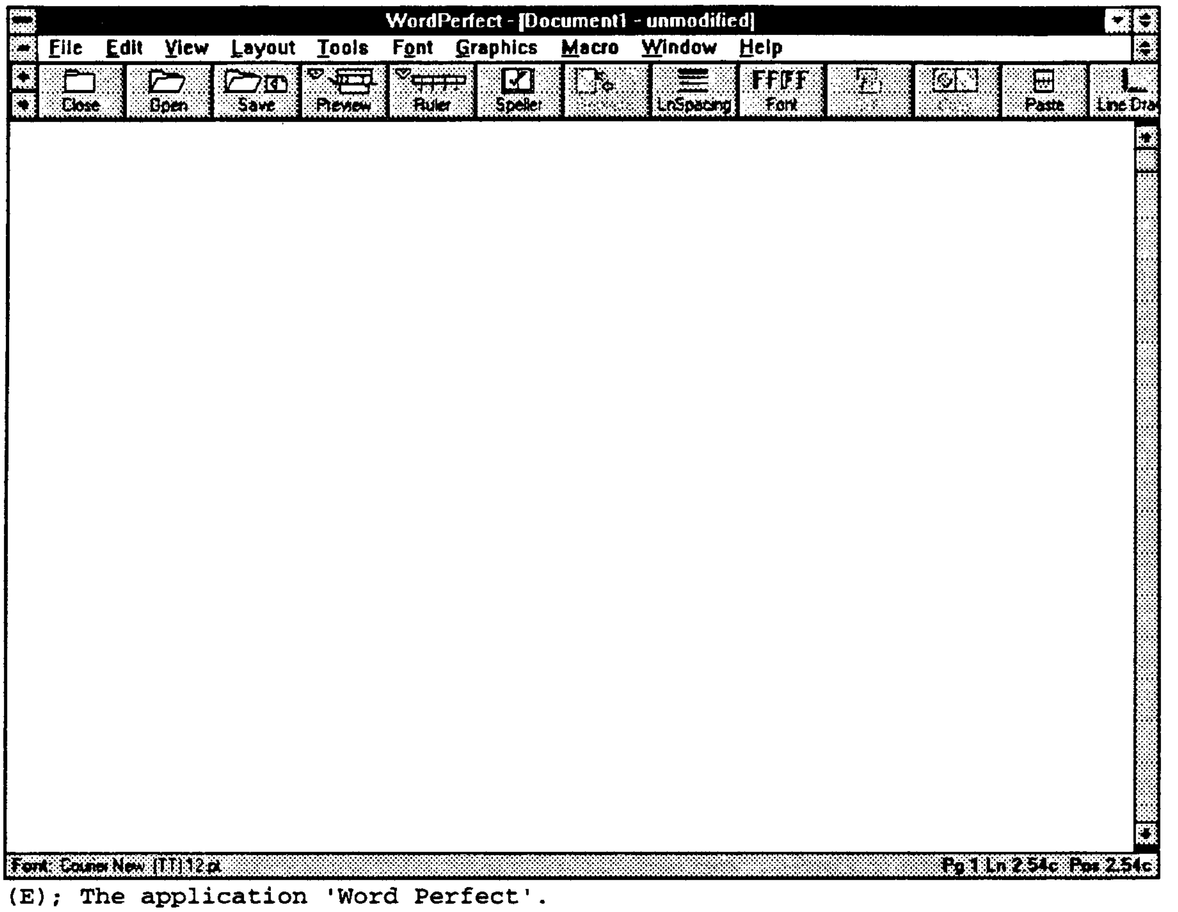
Figure 4: Windows in the Windows 3.1 environment.
The windows which should appear when performing the first application search task.



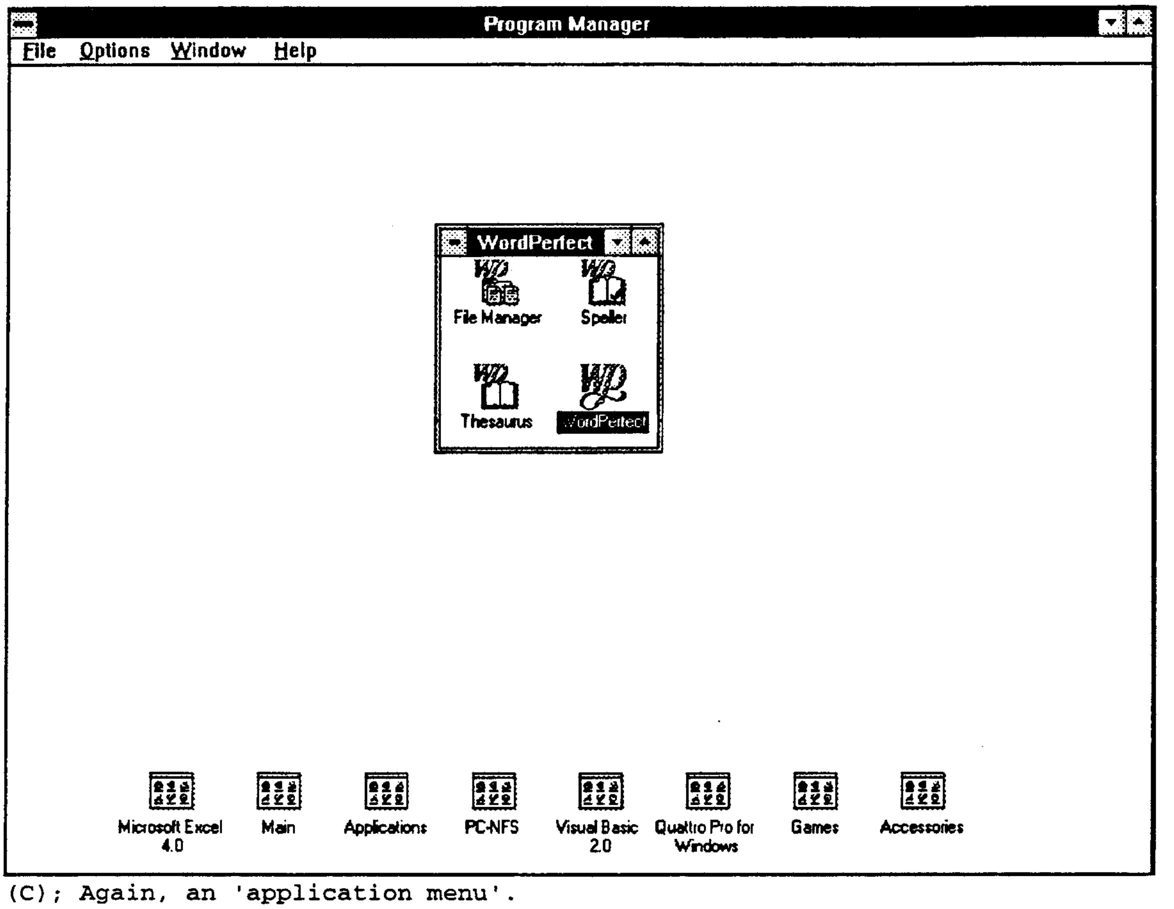
(A); The 'program manager'.



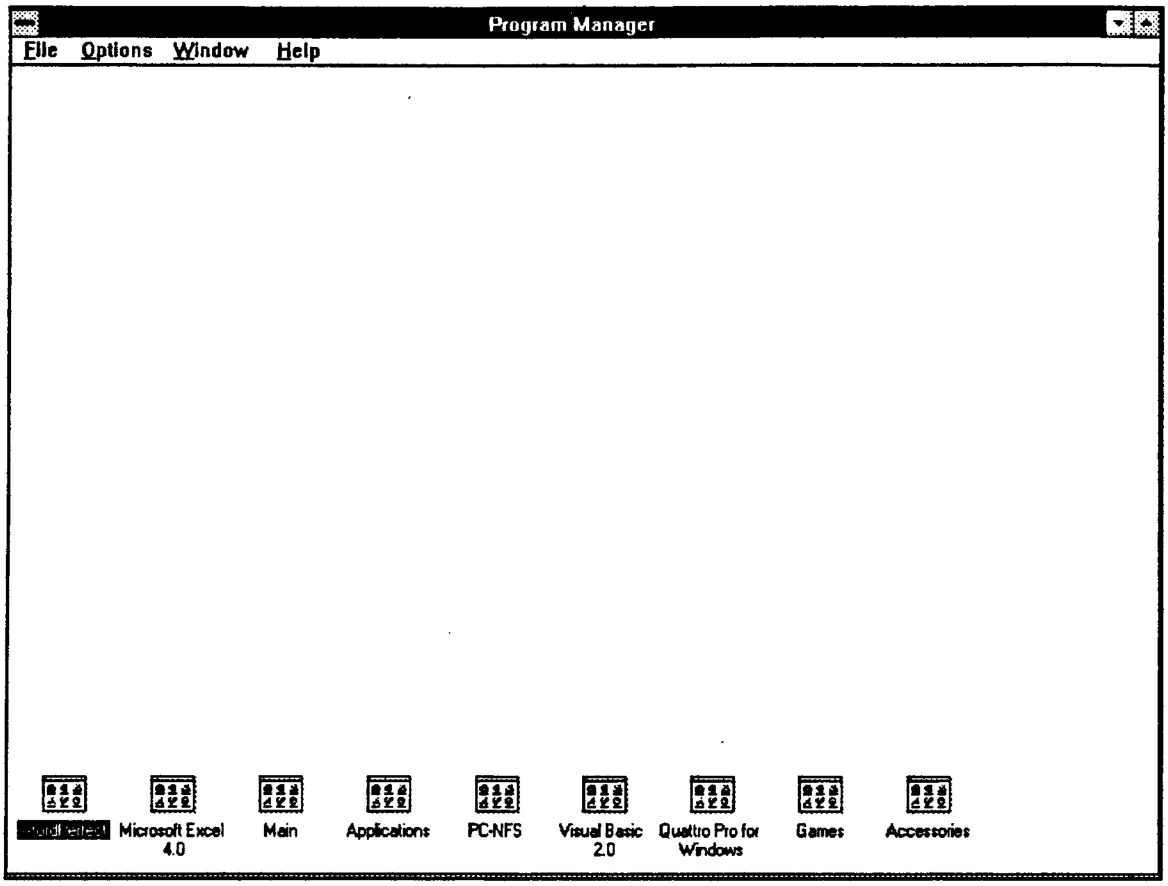
(C) An 'aplication menu'.



(E); The application 'Word Perfect'.



(C); Again, an 'application menu'.



(A); and at last, the 'program manager' again.

2.3. Results

Methods of analyzing

The video tapes of the sessions were analyzed. On the basis of the video tapes, a GOMS analysis was written that gives a close description of the subjects' behavior. In the remainder of this section we will characterize the search and information use strategies of the subjects, on the basis of this GOMS analysis. An example of the behavior of the subjects is that if subjects saw a pull down menu, they read it from top to bottom.

We also used the video tapes to make some time and frequency measurements: measure the time the subjects took to perform a task, to determine how frequently an action was evaluated and how results were evaluated (looking at the title bar or at the overall screen design) and to measure in the program manager how often items were looked at. Some statistical support will be presented using the results in Table 1.

We will sometimes report on the significance of a difference between two variables. Because these differences do not concern hypotheses that were formulated in advance, these statistics should be viewed as an indication of how clear a difference was in our sample and not as strong evidence that the difference would be found in another sample.

In the description of the behavior of the subjects we will use the measurements mentioned in the table.

We always give a mean followed by the associated standard deviation.. For instance: "The total duration of a query is 943 +/- 146 seconds", means that the mean is 943 seconds, with a standard deviation of 146 seconds.

Description of the search and information use strategies

We observed when subjects searched for an item or application in a certain kind of window (e.g. a pull down menu) many subjects used the same search and information use strategies. They also used the same strategy to leave a window.

Therefore we will describe the search and information use strategies in all different kinds of windows. We will describe how subjects leave windows, how they react on unexpected screen changes and messages of the system, and how they evaluate the results.

table 1 Mean frequencies of recorded event types in Experiment 1.

Event type	Mean	Standard Deviation	Standard Error
Evaluate automatically	11.3	3.2	1.1
Look at the title bar	2.0	1.3	0.5
Look at the overall screen design	8.6	3.9	1.4
Search among the application group icons	6.5	1.4	0.5
Search among the menu items	4.4	1.4	0.5
Repeat the action	8.8	4.5	1.6

Time needed

The time subjects needed to perform the queries was measured from the moment subjects started to read the query up to the moment they gave some response that they had completed the query. Because we had some interruptions during the experiments caused by lack of concentration, sounds in the background that disturbed the subjects or by questions asked by the subjects that were not relevant to the queries, we had to correct the result by subtracting the time of these interventions from the time they needed to perform the task. This corrected time is called 'duration of the queries'.

The subjects handled 8 queries with a total duration of 943 +/- 146 seconds, i.e. a single query typically took about 2 minutes. The results concerning the frequencies of the various event types will be discussed below.

Search in the program manager

All queries started in the program manager. When a subject was asked to perform a task, he looked sometimes directly (and sometimes after thinking) at the information desired. The `information desired' would be a piece of information that matches best with the query that had to be performed (e.g. a certain menu, an item or an application). Because the subject could only look at the information desired if he knew what he had to look at, he always used previous experiences. The relevant experiences most used by the subjects were:

- The information the subject was searching for had been seen on screen earlier.
- The subject had seen an item on screen, which (he thought) was a generalization of the item he was looking for. He thought he would find the item desired if he selected this item.

We observed that when no relevant experience was applicable, the subjects systematically searched through the most conspicuous menu. The most conspicuous menu of the program manager is the collection of application group icons. They searched among the application group icons 6.5 +/- 1.4 times. This is 0.8 time per query. Even when the subjects thought they had to search the menuline and started there, they returned to the collection of application group icons if they could not immediately find the item desired in the menuline.

When the subjects told they could not find the information desired among the application group icons, or when they thought straightaway that they had to search in the menuline - illustrated by remarks like: "I have seen the desired item in the menuline, so I will search over there." or by simply reading the menu items aloud - they searched among the menu items. The subjects searched among the menu items 4.4 +/- 1.4 times which is 0.6 times per query.

However, most of the time subjects searched the most conspicuous menu. The difference between the frequencies of looking among the application group icons and looking at the menuline is significant by a paired t-test ($p < .05$).

We observed the subjects did not look at the title bar nor the closebutton when they searched for an item or application. We should held this into account when we design windows for VIPs.

The systematical search among the application group icons and in the menuline was the same for all subjects. We had expected this, because they used their normal reading methods, as when reading a book, to search a menu. People train their eyes to read faster and more easily. As speakers of Western languages read from left to right and from top to bottom, this training leads to a better development of a little part on the right side of the macula and a part directly under the macula (the macula is the part of the eye which people use to read) than the other parts of the eye bordering on the macula (Pollatsek et al. 1989). Thus, the subjects read the menu from top to bottom and from left to right.

Search in a pull down menu

During the experiment we observed that all subjects had the same search strategy in a pull down menu. When they were not able to use earlier experiences, they searched from top to bottom. Sometimes they looked directly at the desired item, so they used earlier experiences (see earlier).

The use of a dialog box

When people saw a normal dialog box they always read and answered it. A captioned dialog box was handled differently. Seven subjects used the scroll bar first to search for the item desired, and when they could not find the information (in two cases) they used the edit control (they typed the word on the command line). The other subject typed the word on the command line rightaway, without using the scroll bar.

Search in an application menu

We observed that the subjects searched among the application icons in the same way as they did among the application group icons in the program manager. Again, they did not look at the title bar or the closebutton.

Search in an application

When the query asked the subject to find a particular kind of information on a static window - e.g. an 'option key, this is a button which can be chosen (see appendix 1 the third option search task) - most subjects (7 subjects) first looked at the text of the static window. The search strategy used by the other subjects could not be determined. We have also noticed that seven subjects looked at the menuline to find the information desired, when they had read the text in the window. If they could not find the information desired there, they took a long time for thinking, and looked around desperately. If there were option keys on screen they were read.

Evaluating the results

When (part of) the query was completed, the subjects had to evaluate the result. We say that a subject 'evaluated the result automatically' if he or she gave some response that indicated that he or she thought the result of an action was good or bad. For instance, a response was "I see a screen which contains the title "Write' , so I know that I reached the editor 'write'. However, it often happened that the subjects did not automatically evaluate the results they had achieved. In 16 situations we checked if subjects evaluated automatically. In eight situations the desired information had been found or a certain action had been performed; in eight situations a query had been completed. An automatic evaluation was performed 11.3 +/- 3.2 out of 16 times.

We think that the reason why subjects did not always make an automatic evaluation was that an observer was sitting beside the them. This might have given them the impression that the observer was evaluating their actions directly, and they may have thought the observer would tell them when they had made a mistake. We tried to overcome this problem by asking why they thought they had performed the actions well. However, this gave a slight deviation from the normal situation outside the experiment.

Subjects could evaluate their results by checking if the screen they were looking for was produced by the desired application, for instance a game. In the table this is called 'look at the overall screen design' and was performed 8.6 +/- 3.9 times, while this situation could maximally exist 16 times.

The subjects looked at the title bar of the window to see if a desired application was reached 2.0 +/- 1.3 times. Again, this situation could maximally occur 16 times per experiment.

In general the subjects tried to find out whether the window could be produced by the desired application. The difference between the frequencies of looking at the overall screen design and looking at the title bar is significant by a paired t-test ($p < .01$).

Return to the program manager

After performing a task, the subjects returned to the program manager by leaving all windows they passed.

Leave a window

A pull down menu disappears when one clicks the mouse button outside the window. A dialog box which contains a message disappears when the question is answered. The closebutton should be chosen in order to leave all other kinds of windows.

We observed that all subjects left the windows in the same way. We expected this, because, as has been described above, there is only one way to do so.

Control all screen changes

The description above explains how subjects reached the desired information that matched with a task. But this analysis is not complete. People make mistakes (e.g. they choose the wrong item) or the system gives a reaction they do not expect. That is why people - mainly during the time they learn a new system (Hammond 1987) - check all screen changes. We observed that if the screen change was what they expected it to be, they proceeded to the next part of the task. If a screen change was not expected or they thought a false screen change had occurred, they tried to understand it and, if needed, to correct their actions. In the following paragraph we will describe the behavior of the subjects in this experiment.

If the system gave no screen changes, the subjects repeated the action just performed. What we had not expected was that the subjects also repeated the actions they had just performed when they were confronted with an unexpected screen change, because they had made a mistake. For instance, when they saw a new window instead of a pull down menu. The subjects stopped this type of repetition after a while, when they saw that the desired screen change had not occurred. Then they would start thinking.

In the experiment they repeated an action 8.8 +/- 4.5 times. For example, they repeated the activation of an application. The reason why they did this was that they usually thought the action had not been performed correctly if they saw no screen changes.

Messages of the system

Most subjects automatically gave the desired reaction to messages of the system. An example is that, when they saw the mouse pointer changed into a small hourglass, seven out of the eight subjects said: 'I'll have to wait, because I see an hourglass.' The other subject asked what he had to do now. Another example is that the same reaction was shown when the mouse pointer changed into a small hand, a sign that you can point to (select) an item.

Check the screen changes when an action is performed

Often the result of a performed action was not clear to the subjects (because they had no experience with Windows). However, most of the time the subjects still evaluated the result of a performed action in the same way. If the screen changed at the moment the action was performed, the subjects were satisfied. This reaction was confirmed by a subject who was very satisfied when the screen changed after a performed action, whereas normally this would not have been the case in that situation. The other subjects were not satisfied in this situation, because they had not seen a screen change. Most of the time, if the screen did not change, subjects returned to the item they had selected. They wanted to discover a change closely connected to this item. However, this situation seldomly occurred and therefore we have no statistical evidence on it.

2.4. Conclusion

Summary of the search strategies

The search and information use strategies of sighted people is defined very clearly by this experiment. The search and information use strategies depend on two factors. The first factor is, that another search and information use strategies depend on the type of query (option search task or application search task). The different search and information use strategies are described below. The second factor is that the search and information use strategies depend on the window type (e.g. a pull down menu or an application menu). Again, the different search and information use strategies are described below.

Five different Window types were used in the experiment: the program manager, pull down menus, application menus, applications and dialog boxes. When subjects had to perform a Option search task, they tried to use previous experience to find the information desired. If no relevant experience was available, they looked at the most conspicuous information and searched it systematically. The most conspicuous information in the program manager it is formed by the different application group icons, in the application menu it is formed by the application icons and in the application it is formed by the most conspicuous information in the text. Somewhat less conspicuous information in the application is the menuline, followed by the least conspicuous information, the collection of option keys. In a pull down menu there is no conspicuous information, therefore the subjects searched it systematically if they had no relevant experience.

If the subjects had to perform an application search task, most of the time they tried to establish whether the window could be produced by the application desired. And when they could not see if the application desired had been reached, they looked at the title bar of the window.

Reacting on screen changes

We have seen how the subjects reacted on changes on screen. When they saw no screen changes after performing a certain action, they repeated their action(s). Most of the time unexpected screen changes were also interpreted directly. For example, in a dialog box which contained a message, subjects answered the question directly. Another example is that messages from the system usually were understood directly (e.g. an hourglass).

Way of use in the second experiment

In the second experiment we will use these search and information use strategies to develop a screen design for VIPs. We will test these screen design by asking VIPs to perform the same queries as in thee previous experiment and we will see which changes should be made to improve the screen design.

3. Experiment 2

3.1. Introduction

Relations between the first and second experiment

In the first experiment, we have seen the strategies sighted people use while working in a Window environment. We saw which information was used most and which was used least. We used these strategies to design Window screens for the second experiment in which VIPs can find their information in an easy way.

This means that all information is put in a column, so that VIPs can find the information without using spacial information. The order in which the information is put in a column depends on different factors.

One factor is the search and information use strategy. We saw that, when subjects were searching for an item or application, they looked at the items and applications on screen, and not at the titlebar or the closebutton. So it is important that the items and the applications are put on another part of the list, apart from the closebutton and the title bar. Because we want to develop a system that can be read easily with the sliders, we put the items and applications on top of the column and the closebutton and title bar at the bottom, so that subjects can easily distinguish between searching for an item or application and leaving windows.

Another factor is formed by the frequency of use discovered in the first experiment. For instance, the application group icons are used more frequently than the menu items. Therefore the applications is placed above the menuline.

Other factors which should be taken account of are for instance, when information is put in a column, less information can be presented on the screen. Therefore we have to make some generalizations of menus. When subjects choose this generalization, they can read the menu. Another factor is formed by the design guidelines of screen layout should be taken account of. For instance, objects that are used on different windows should have a fixed location on the screen (for instance in our situation the item 'close') (Card et al. 1980) (Paap et al. 1980).

Broad description of this experiment

In this experiment we did not use a Window environment with the special screen design which has just been described, but we used an ordinary Window environment in which VIPs had to perform some queries in a special way. The subjects were to ask a person who simulated the Windows 3.1 environment (henceforward the `simulator') to inform them about the information on the current screen. The simulator gave this information according to the instructions mentioned in appendix 2. These instructions (called principles) take the factors mentioned above into account. Figure 5 describes some screens used by the simulator.

An example of a principle is for instance "If a window contains menu items and one or more other kinds of information (e.g. text), the simulator gives a summary of the information on screen in the following way.

- If there is an application menu, he says 'applications'.
- If there is a menuline, he says 'menuline'.
- If a window has a name, he says 'titlebar' and tells the user the name of the window.
- At the end of the menu, he says 'close'.

Example:
applications
menuline
title bar: program manager
close

When a subject wanted to know what was on screen, he would ask for instance: "What does the screen show?" Then the simulator would say: "file, options, windows, help and back." (see figure 5). When the subject asked "Select the item 'file'; what does the screen show now?", the simulator said: "new, open, delete, move, copy and back." Only after this experiment the construction of the VISA-computer simulation could be made.

The experiment had several aims. First, we would get an impression of how VIPs search for information. Do they have the same search strategies as sighted people? For example, do they also revert to previous experience? Second, we wanted to find out whether our new interface is adequate. If we found faults in our principles, we would have to correct them for our next (third) experiment.

3.2. Methods

Subjects

We used one subject for the pilot experiment and four subjects for the experiment. They were paid for participating the experiment. Their age was between 18 and 45 years. All subjects were visually impaired, and all were experienced computer users. No subjects had any experience with a GUI environment. All subjects were interested in making graphical modes accessible to the blind.

Figure 5: Windows which the simulator reads aloud. These windows should be passed when performing the first option search task. The windows (A) and (G) are 'the program manager'. The windows (B) and (F) are the menuline, the windows (C) and (E) are a pull down menu, and window (E) is a dialog box.

```
applications
menuline
titlebar 'program manager'
close
```

(A) 'The program manager'

```
file
options
window
help
back
```

(B) the menuline in the 'program manager'.

```
new
open
delete
properties
run
exit
back
```

(C) A pull down menu in 'the program manager'.

```
radiobutton 1 new program item
radiobutton 2 new program group
button yes
button cancel
button help
titlebar: new program object
```

(D) A dialog box.

```
new
open
delete
copy
back
```

(E) Again, the pull down menu.

```
files
options
windows
help
back
```

(F) The menuline,

```
applications
menuline
title: 'program manager'
close
```

(G) And 'the program manager'.

Hardware and software

We used the same queries as in the first experiment, because this would optimize the comparison between the two groups of subjects (people with normal eyesight versus VIPs). Moreover, faults which had been found in the first experiment were repaired, for instance we changed a word in a query, because the word had proven to be unclear (see appendix 1).

The queries were randomized according to the Latin square with regard to the following constraints:

- start with a option search task, because the trial session contains an application search task.
- end with an application search task (as in experiment 1)
- two questions of the same kind (application search task or option search task) cannot be performed after each other (as in experiment 1).
- skip the filler task, because this query is a visual task and cannot be performed auditorially.

After the queries, we asked every subject if he liked working with the system and how it could be improved. We wanted to get new suggestions about how our system might provide better accessibility to windows for VIPs.

Information on screen was given according to the principles given in more detail in appendix 2.

We used a Phillips taperecorder in this experiment, because a videorecorder would not give us any additional information. Furthermore, we used a personal computer 386 AT with DOS and Windows 3.1.

Procedure

When the experiment started, the subjects received information about the structure of the Window environment and about the way they had to perform their queries. There was one application task, which was meant to familiarize them with the system, and the way of questioning the screen by means of the simulator. The tape recorder was started and the experiment began. After having performed eight queries, the subjects were asked the questions mentioned before.

3.3. Results

3.3.1 Search and information use strategies

Method of Analysis

We used the tapes to make a GOMS analysis, which describes the behavior of the subjects in this experiment. We used this analysis to see which Window objects are used in which situation.

As in the first experiment, we described the search and information use strategies with the support of statistics. The significant measurements should be treated the same way as in the first experiment, i.e. as an indication of how big the differences were. The measurements that are described in the text are written down as in the first experiment.

We used the GOMS analysis of the first experiment as a basic framework for comparing the search strategies of VIPs and sighted persons. Furthermore, the GOMS analysis was used to find the faults in the newly designed screen design.

Table 2: Mean frequencies of recorded event types in experiment 2.

Event Type	Mean	Standard Deviation	standard Error
Extra information needed	4.0	3.6	1.8
Evaluate automatically	5.5	2.4	1.2
Look at the title bar	2.3	1.5	0.8
Interrupt information	5.8	4.5	2.3
Look at the overall screen design	4.3	1.9	0.9
Repeat the action	0.5	0.6	0.3
Slip between 'back' and 'close'	6.3	3.9	1.9

Time and extra information needed

In this experiment 8 queries with a total duration of 1256 +/- 285 seconds, were performed. Which means that the average query took about 2.5 minutes to handle.

It should be noticed that VIPs used extra information offered by the questioner during the queries, because sometimes they did not know how to read a screen or they interrupted the simulator too soon, and then they did not receive the information they needed. For instance, when a subject asked for the application group icons, he would hear: "editors, calendar, write, ..." and then he might intervene and opt for `editors', whereas he should have selected `Word perfect', but this application group would only have been mentioned after `write'. Once he had arrived in `editors', he would not see 'Word Perfect' and became confused. Extra information was needed 4.0 +/- 3.6 times.

Search in a menu

In our screen design we only had menus that looked like the structure of a pull-down menu in Windows 3.1. All window objects (e.g. menu items) were put in a column. The difference is that the menus designed according to our principles put the items that could be selected on top of the column and the title bar and an item which gave the possibility to leave the menu were situated at the end of the column. Because all menus have the same structure the search strategies are the same in all menus.

We observed that all subjects spent time trying to remember the desired item. When they had read the query they would be silent for a while, or they would think aloud about how they could find the item.

If they could remember a desired item, they did not want to hear all items, but interrupted the simulator 5.8 +/- 4.5. times. This method was not used frequently by all subjects.

If they had no relevant experience they searched through the menu from top to bottom. The search from top to bottom was trivial, because the information was presented in this way.

Evaluation of the results

The VIPs were very careful while choosing an item. They took their time if they did not know precisely what they had to do and did not try without thinking first. They expected their choices to be the right ones and often did not evaluate the results automatically. As in the first experiment, we measured in 16 situations if the evaluations occurred automatically, but in the experiment they evaluated the results only 5.5 +/- 2.4 times.

When the subjects had either reached an application or performed an action or when they returned to the program manager, they evaluated this by looking at the title bar 2.3 +/- 1.5 times (with a possible maximum of 16 times) or at the screen design 4.3 +/- 1.9 (this also had a possible maximum of 16 times). The difference between these kinds of evaluation is not significant by a paired t-test ($p < .05$),

We observed that the subjects reacted directly to unexpected screen changes. For example, all subjects reacted on a dialog box directly. We had not expected this, because we had given no information about unexpected screen changes. However, when we heard the tapes after the experiments, we heard that the voice of the simulator slightly changed when the screen changed. The subjects noticed this changed voice and they interpreted it as something abnormal. Therefore they asked the simulator what had happened.

Leave a menu

All menus in our system have an item 'back' or 'close'. The item 'close' should be chosen when leaving a window and return to the previous window. The item 'back' is chosen when a menu disappears from the current window. We have discovered that the difference between these items was not clear to the subjects, because they mistook 'back' for 'close' and vice versa 6.3 +/- 3.9 times. This is the reason why in our next experiment we decided to use a universal name for this item: 'closebutton'. We should point out that the frequency of this error varied very strongly from subject to subject. More precisely, we noticed that two subjects were able to choose the correct item straightaway and two subjects could not.

3.3.2. Comparing the first and second experiments

Search strategy

The way in which subjects searched for desired information in the second experiment can only be compared with the searching methods in a pull down menu as used in the first experiment because the second experiment used a menu that looks like the structure of a pull down menu (see the paragraph 'search in a menu').

We find no differences of approach in searching an item in a menu in the experiments. In both cases, subjects first try to use previous experience and secondly, they search systematically.

We expected this to be the case, because in the first experiment subjects used the normal reading strategy (reading from top to bottom and from left to right). In the second experiment the simulator of the VISA-com system also used this reading method, so the information on screen was presented to the user in the same way.

Time used to perform the queries

In the first experiment, subjects needed 75% of the time to perform the queries in comparison with the time subjects needed in the second experiment.

We think this effect is caused by at least two factors. First, VIPs had to hear the information before they could choose it (Visual perception is much faster than hearing). And second, they had no overview of what happened; they did not see changes on screen (e.g. when an application had been reached, they had to ask first what had happened).

Evaluating the results

The way in which subjects evaluated the results in both experiments was different. The frequency of evaluation was also different.

In the first experiment most evaluations were done by looking at the overall screen design (8.6 times), which means that the mean frequency of looking at the overall screen design has a mean frequency over all subjects from 8.6 times. This was done more often than by looking at the title bar (2.0 times). In the second experiment, however, we did not find such a large difference between the two ways of evaluating results. The screen was either evaluated by looking at the title bar (2.3 times) or at the overall screen design (4.3 times).

This difference can be explained if one considers that the sighted subjects of the first experiment had an overview of the whole screen, so that they could use the information offered by an overall view of the screen. In contrast, the VIPs of the second experiment could not use this information.

Two confounding variables could have influenced these results. In the second experiment, the simulator read the whole screen automatically when subjects asked to hear the screen, and therefore they could not distinguish between the different types of evaluation clearly. Moreover, as the subjects looked at the title bar automatically in the first experiment, they knew they had reached the desired information. Nevertheless, when they were asked how they knew this, they were usually looking at the whole screen and therefore thought that was -we think - why they knew it, unless they had consciously evaluated the screen by looking at the title bar.

Evaluations that were given automatically (explained earlier) were more frequent in the first experiment than in the second experiment (evaluate automatically in experiment 1 11.3 times, evaluate automatically in experiment 2 5.5 times). We think this is caused by two factors. The first is that the subjects in the second experiment had to make explicit evaluations: they could not look at a particular kind of information, but had to say something like 'Read the screen.'. The second is that the simulator read the screen and they thought that they would be warned if he made a mistake.

Control the screen changes

Screen changes were handled identically in both experiments. The reason is that in both cases the subjects knew that a screen change had taken place (in the first experiment because they saw a screen change and in the second experiment they heard that the voice of the simulator changed which make them thinking that something unexpected has happened). Moreover the simulator read the dialog box just like the sighted subjects of the first experiment did.

3.4. Evaluation

The effectiveness of this design

An important point to be made is that for two reasons, one cannot say exactly whether the screen design we aim at (explained in the third experiment) will function properly.

The first reason is that normally people do not read all items, but they immediately know where to find the item and select it without reading all items first. For instance, they know that the information they needed is positioned on the third line, and they move the vertical slider to the third line and read the item over there. The second reason was given by one of the subjects after the experiment: "I want to work with your colleague (the simulator) instead of a computer, because it is easier to instruct a person than a computer." Thus the entire interface would available for facilitating the performance.

The subjects used the same search strategy as sighted people do in a pull down menu. They first tried to think of a desired item, and if no relevant experience was available, they searched trough the menu systematically. Using relevant information was done by interrupting the simulator and asking him to select the desired item. Systematical search was done by asking the simulator to read the item on screen.

VIPs sometimes needed extra information to know more about the screen design or the items presented on screen.

Part of the time the result of a (partially) executed action was not evaluated automatically. When evaluating the result, they looked at the title bar or the screen design as a whole. No significant differences were measured.

Two imperfections have been discovered. The first is the difference between the item 'back' and the item 'close'. The second imperfection is the remark 'titlebar: ...'. Three subjects wanted to select this remark a couple of times.

The first imperfection has been corrected by changing the item 'close' and 'back' into the item 'closebutton'. The second has been corrected by changing the remark into an entire sentence, namely: 'The titlebar is the ...'

Remarks of the subjects

All subjects said that they would like to work with the system, because they would be able to find their information in it, and that is the most important issue to them.

They all made some remarks about the advantages and disadvantages of this presentation of the screen. They mentioned that "The structure of the system is very good, because there is not so much information on the screen, and the information is clearly structured. In this way, you do not miss information and you know where you can find the information". The disadvantages mentioned were: "You get a navigation problem: if you have to do too many selections, you do not know where you are. Furthermore, when you know which item or application you want to have, it is annoying you have to pass different screens. And it is also a problem that your sighted colleague has another screen design than you have yourself." It would be a good idea to test both systems in the next experiment and look into the imperfections and the advantages of both interfaces.

Thus, we will also get an idea about how to extend the systems. Two subjects have told us that a command structure is lacking. "When I want an item, and I know that the current screen contains that item, I have difficulties to select it. If I could give one or more letters, I could select this item more quickly." These remarks give also an idea how this command line should work. Instead of moving the sliders to a desired item to select it, a command can be given - typed on the keyboard - which moves the cursor directly to the desired item. An example of such a command is 'ctrl f' to select the item file.

4. Experiment 3.

4.1. Introduction

Broad description of this experiment

In this experiment we tested two prototypes of a window simulation. The difference between these two prototypes is their screen design.

The first prototype contains a window simulation which has the same screen design as an ordinary Window environment. The difference between this version and the ordinary window environment is its interface. Now items can be read if the horizontal and vertical sliders are moved to the position on which the item is located and the vertical slider is pushed. Then the item can be selected by pushing the horizontal slider. We called this version 'the breadth version' and we will describe this version in more detail below.

The second prototype contains an interface which has been built according to the principles of the second experiment, including the corrections on them, which we explained in the results of the second experiment. In the following section we will give some extra principles that should be taken into account to build this interface.

Globally the second interface works as follows. The Window environment can be seen as a book of which the pages have the structure of a column containing information (e.g. menu items or applications). This column is put at the left margin. All pages basically have the same structure. When one chooses an item on top of the page, it is turned over, and one arrives at the next page. For instance, when 'menuline' is chosen, the menu items appear (e.g. in the program manager are these items 'file', 'options' 'windows' and 'help'). At the bottom of the page the title bar can be found and the closebutton is beneath it. When the closebutton is selected, the page is turned back. The menus can be read by moving just the vertical slider to the desired line (item) and pushing it. The item can be selected by pushing the horizontal slider as in the 'breadth version'.

The aim of the experiment was to find out if VIPs could work with the two interfaces. For instance, if they knew where they could find the information, how they had to leave a window, etcetera. We wanted to discover the shortcomings and advantages of both systems, like: `Could people easily find the information they needed?' We want to discover the similarities and differences between both screen designs. For example, in both versions of the system subjects had to move the vertical slider, but in the breadth version, the horizontal slider had to be moved as well. We also wanted to know in which situation which system was be preferred to the other. After the experiment we asked the subjects a few questions in order to improve the evaluation.

4.2. Methods

Subjects

We used ten visually impaired subjects. Two subjects are used to participate in the pilot experiment. All are experienced computer users without any experience with Windows. The subjects were paid for participating in the experiment. Their age ranges from 15 to 40 years old.

Hardware and software

The same queries have been used as in the first and second experiments. The queries were randomized as follows:

- Begin with an option search task (as in experiment 2).
- End with an application search task (as in experiments 1 and 2).
- Two queries of the same kind cannot be performed after each other (as in experiments 1 and 2)
- Randomize the items according to the Latin square (as in experiments 1 and 2).
- The first four questions should be made with one version of the system, and the last four questions with the other version.
- Four subjects should start with 'the depth version' and four subjects should start with 'the breadth version'.
- All application search tasks and option search tasks should be performed with the same frequency in both versions of the system. All queries (e.g. the query 'Go to Word perfect') were performed by four subjects in each version.

We asked some questions so as to evaluate the system. We used questions like: `What are the shortcomings of this system?' and `How do you think the system can be corrected?' We also asked some questions that used a scale of seven points. An example is: 'Which system would you prefer to work with?' (See appendix 1 for a list of the questions).

We used the same Phillips tape recorder we used in the second experiment, and a MS DOS computer AT 286. The software used in this experiment is a program that simulates the MS Window environment 3.1. Some new hardware was constructed to make the selection of an item with the horizontal slider when the sliders are positioned on this item possible.

The interface of the 'breadth version'

In the breadth version the general code of the objects which are presented on a line can be read if the vertical slider is pushed (see figure 6 A). Every code is a single letter. For example, when on the first line of a window 'c t' can be read by pushing the vertical slider. This means that two window objects are on that line, but nothing is known about which objects are presented.

Because it is important for the user to get a clear object name (Paap et al. 1988) this is given in the following way. After moving the horizontal slider to a letter, e.g. the 'c', (if a low beep is heard, the letter is reached), the information which is presented by this letter can be read by pushing the vertical slider. 'Closebutton' is heard (figure 6 B). Now the horizontal slider can be pushed, and the closebutton is activated. If the closebutton is not activated and the horizontal slider is moved to the right, a low beep indicates that the next window object has been reached. When this information is read, again by pushing the vertical slider, 'titlebar: program manager' is heard (figure 6 C). This part of the information cannot be selected.

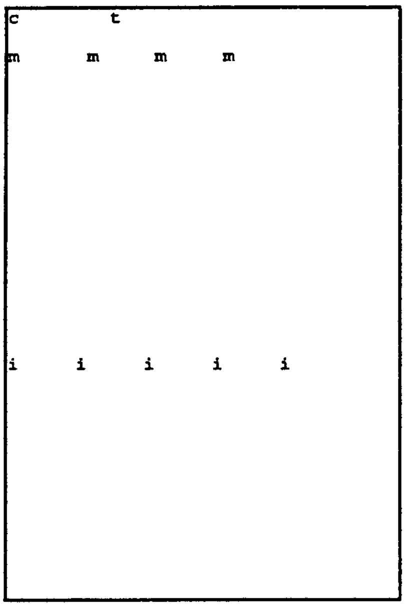
The menu items (figure 6 D E F and G) and the application group icons (figure 6 H I J K and L) can be read in the same way. As has been explained earlier, the information on screen is in the same position as in the ordinary window environment. But we have made some simplifications in this system, for instance that icons and the closebutton can only be activated (push the horizontal slider) and not selected. items in the menuline and option keys can only be selected, also by pushing the vertical slider. In figure 7 some screens are presented which should be passed when performing an option search task.

The interface of the 'depth version'

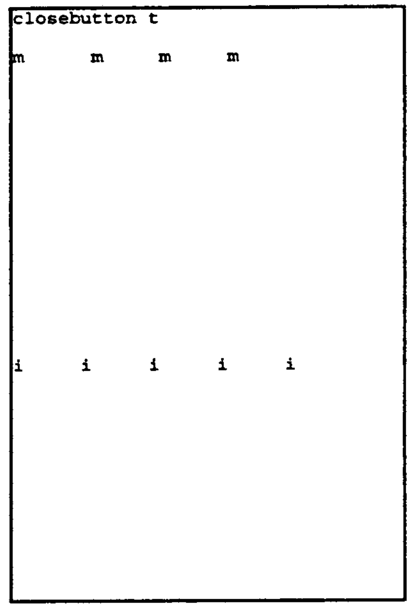
In this interface all information is presented in a column. If one moves the vertical slider down, one can read the window objects with their names directly, without using the horizontal slider. E.g. if you are in the program manager, you can read: 'applications' on the first text line,. 'menu bar' on the second text line, 'the titlebar is the program manager' on the third text line and 'closebutton' on the fourth text line. In figure 8 and 9 we give some examples of the screens in 'the depth version' of the VISA-comp simulation.

In a 'normal' Window 3.1 environment, menu items can be selected - to activate a menu - and applications can both be selected - - then, a new selection (in a pull down menu) leads to the activation of the application - or an application can be activated directly. Because there is no difference between activation and selection in both simulations of the VISA-computer simulation (pushing the horizontal slider) we will call all activations and selections 'selections'.

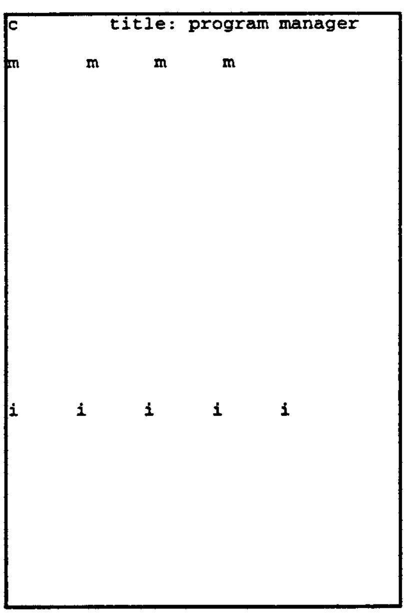
Figure 6: Reading the 'program manager' in the 'breadth version' of the VISA-comp simulation.



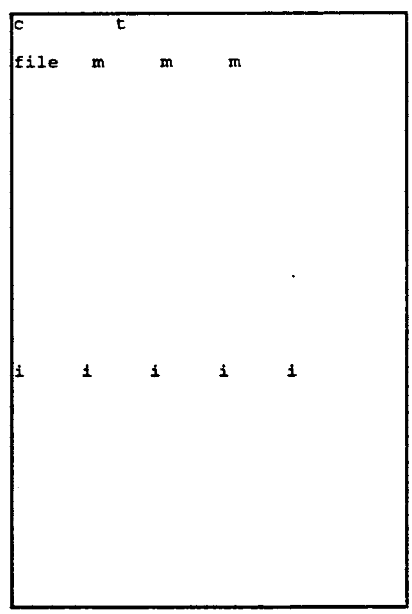
(A): The code of the window objects



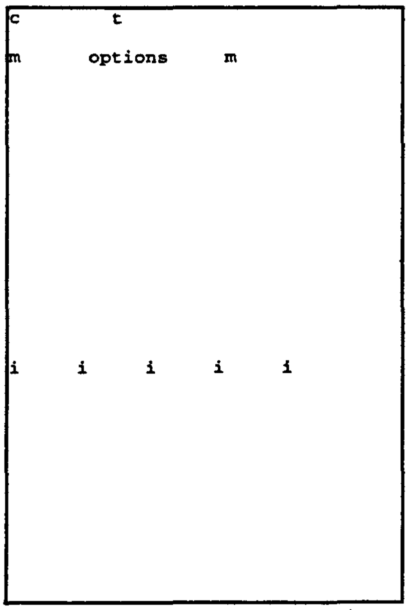
(B): the 'closebutton'



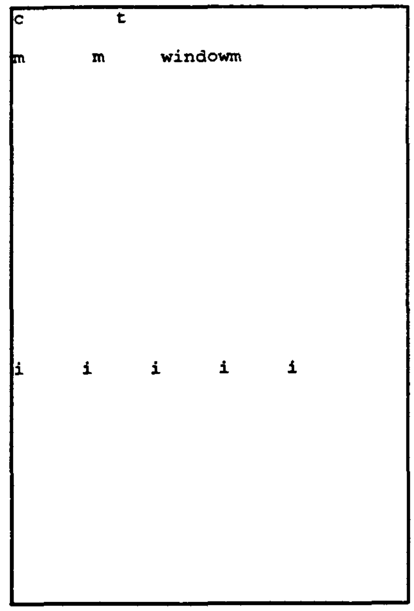
(C): The 'titlebar'



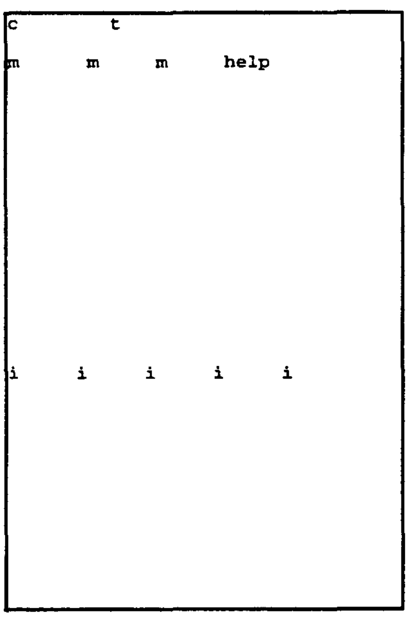
(D): The menu item 'file'



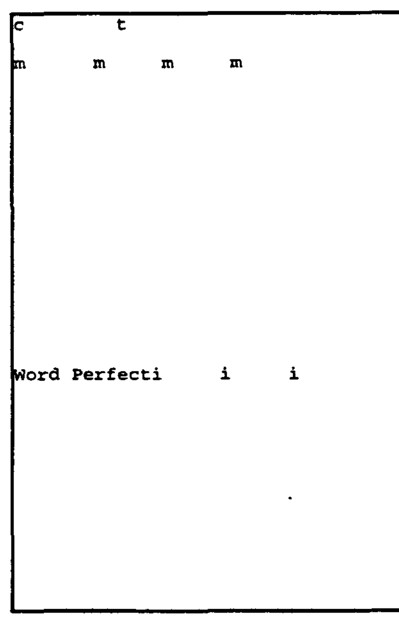
(E): The menu item 'options'



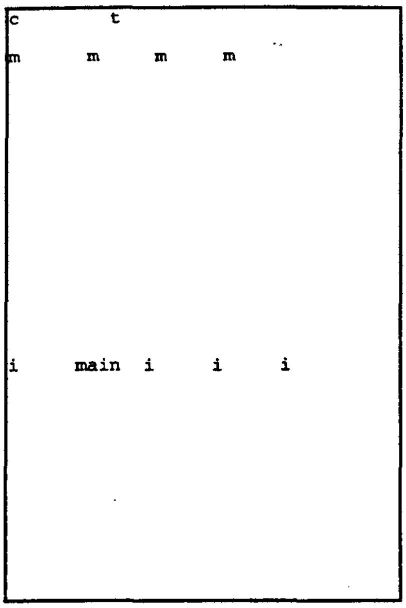
(F): The menu item 'window'



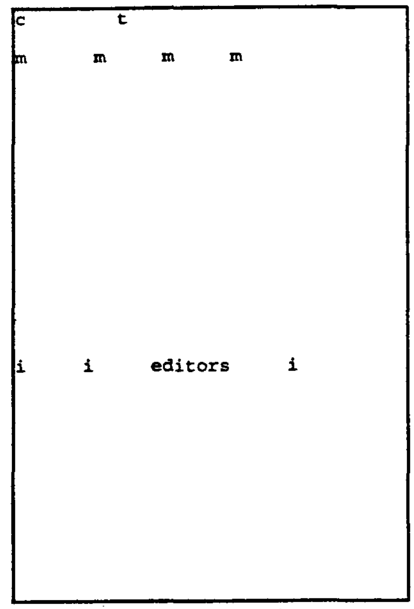
(G): The menu item 'help'



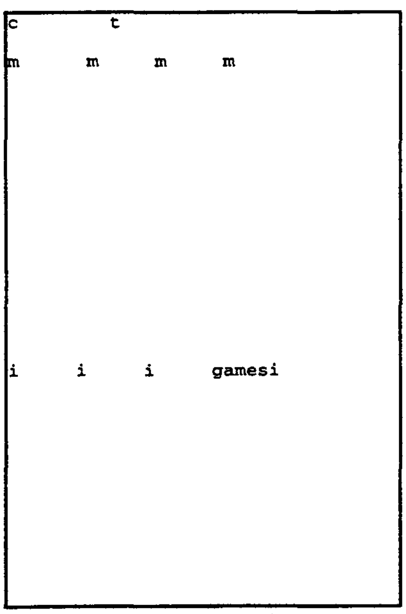
H): The application group icon 'Word Perfect'



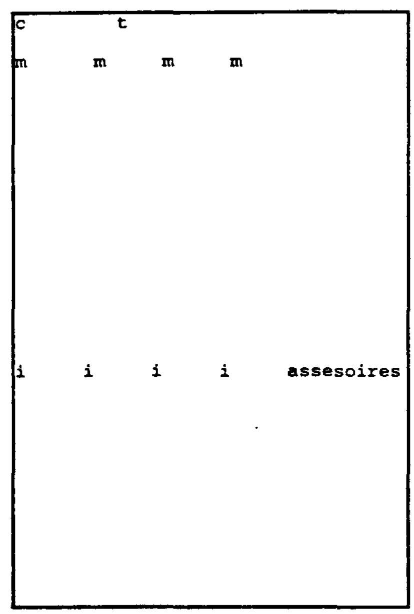
(I): The application group icon 'main'



(J): The application group icon 'Editors'

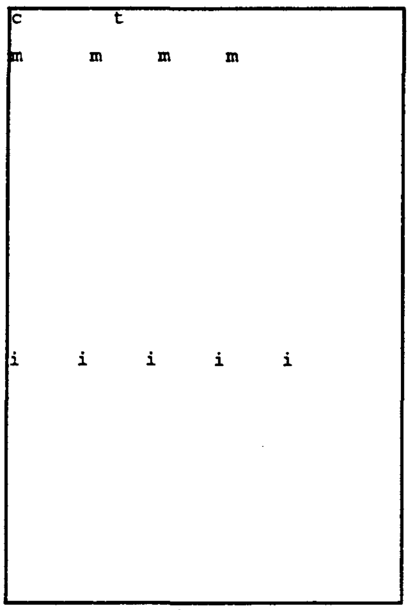


(K): The application group icon 'Games'

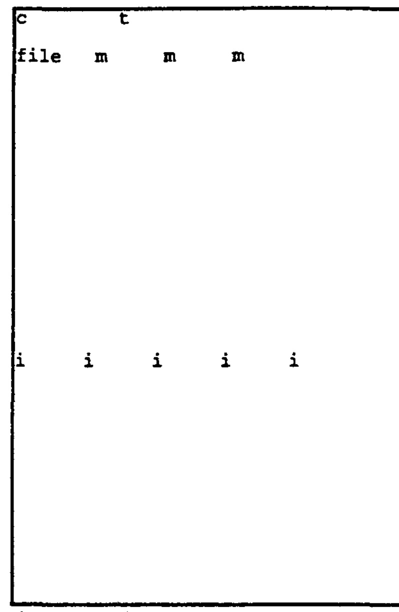


(L): The application group 'assesoires'

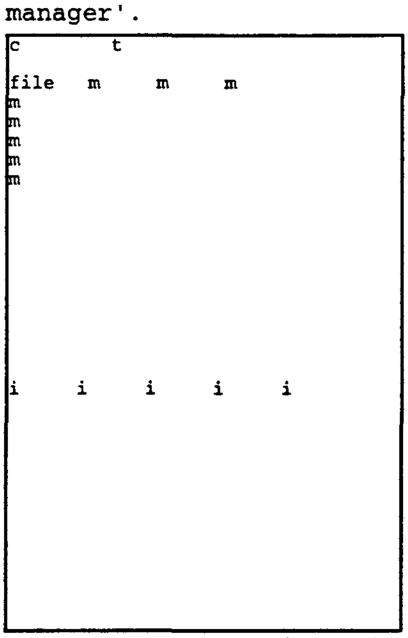
Figure 7: Windows in 'the breadth version' of the VISA-comp simulation.
The windows which should be passed when performing the first option search task. The following windows are the program manager: (A), (B), (H) and (I). The windows (C) and (D) are a pull down menu in the program manager. A dialog box is given in window (E), (F) and (G).



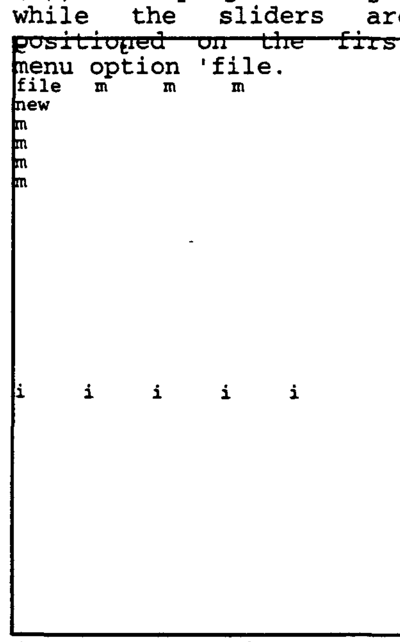
(A); 'The program manager'.



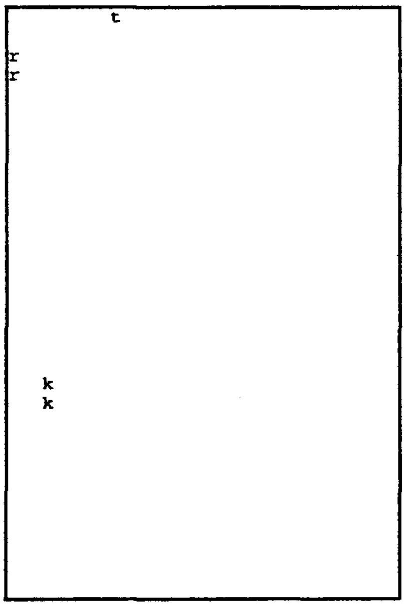
(B); 'The program manager while the sliders are positioned on the first menu option 'file.



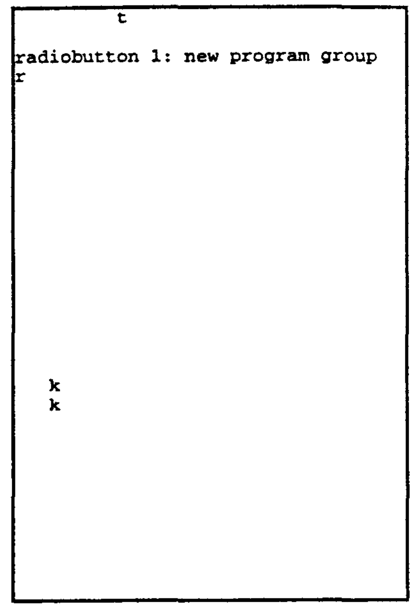
(C) after selecting the option 'files the pull down menu appears.



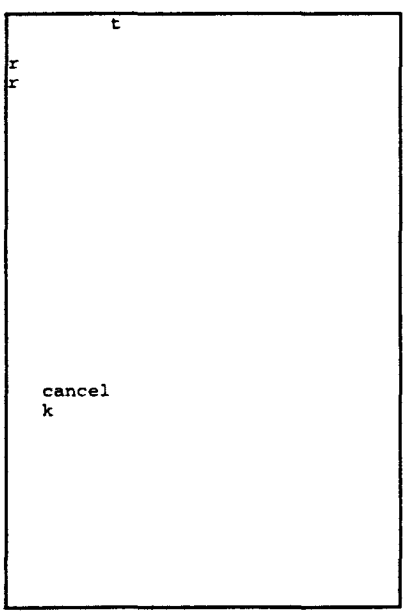
(D); The sliders are positioned on the option 'new'.



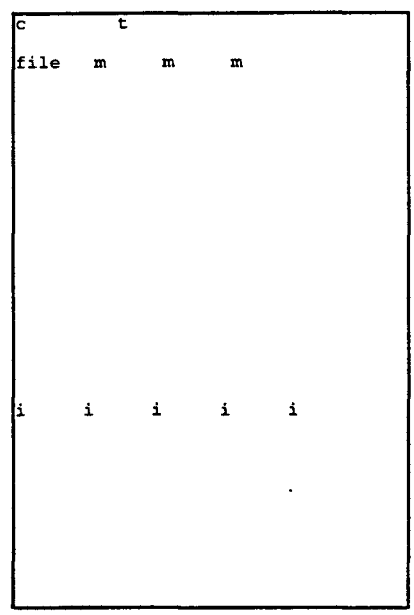
(E); After selecting the option 'new' the dialog box appears.



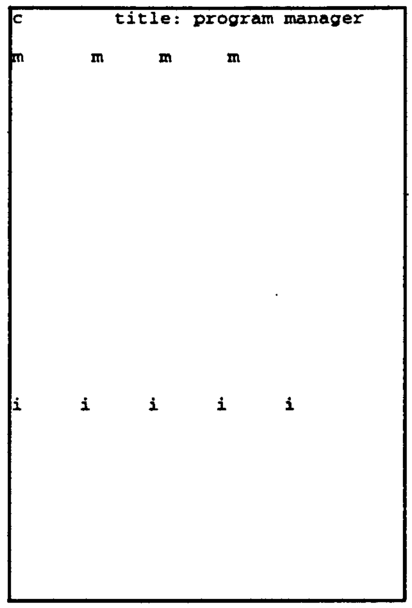
(F); The sliders are positioned on the first radio button.



(G); The sliders are positioned on the 'cancel' button.



(H); after selecting 'cancel' the pull down menu in the program manager is reached again.



(I) The sliders are positioned on the 'titlebar'

Figure 8: Windows in 'the depth version' of the VISA-comp simulation.
The windows which should be passed when performing the first option search task. The windows (A) and (G) are 'the program manager'. The windows (B) and (F) are the menuline, the windows (C) and (E) are a pull down menu, and the window (E) is a dialog box.

```
applications

menuline



the title is the program manager

closebutton
```

(A); 'The program manager'

```
file
options
window
help



close button
```

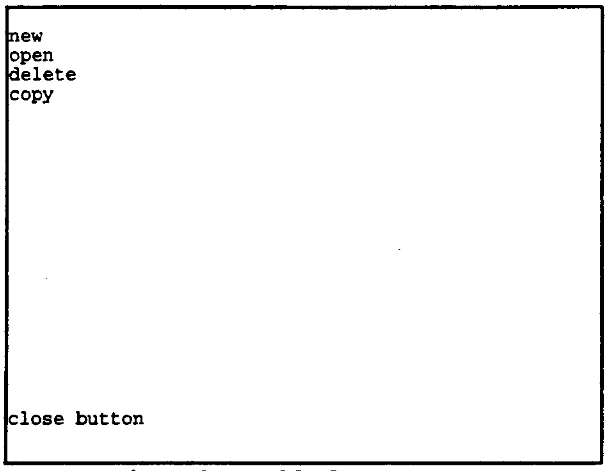
(B): the menuline in the 'program manager'.

```
new
open
delete
properties
run
exit



close button
```

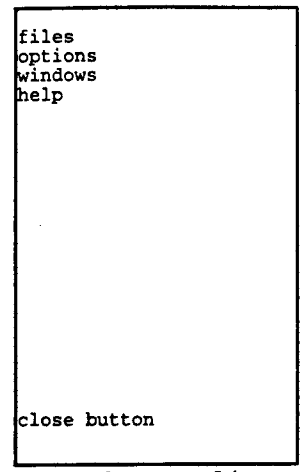
(C); A pull down menu in 'the program manager'.

```
radiobutton 1 new program item
radiobutton 2 new program group

button yes
button cancel
button help



The title bar is new program object
```

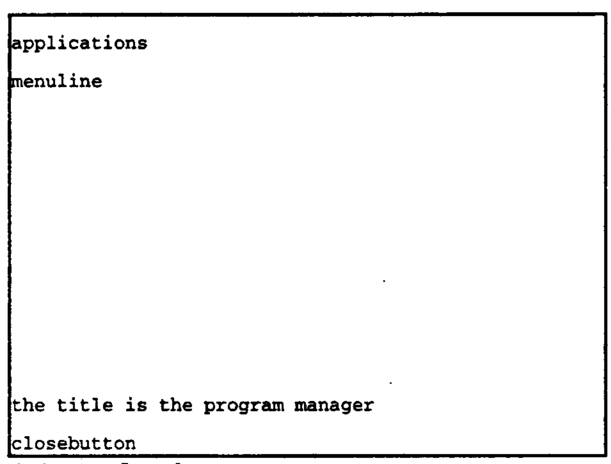
(D): A dialog box.



(E); Again, the pull down menu.

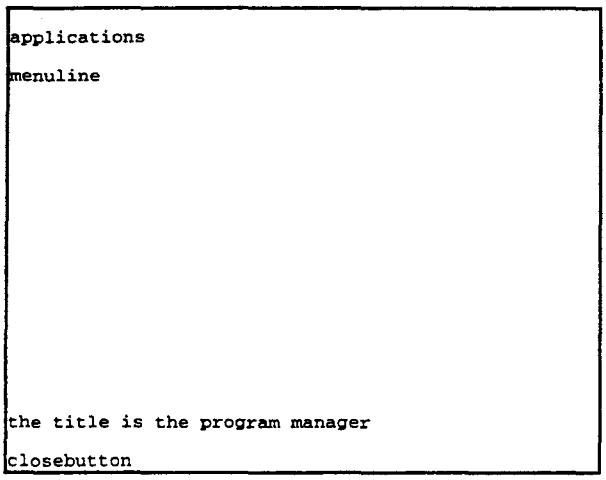


(F); The menuline,

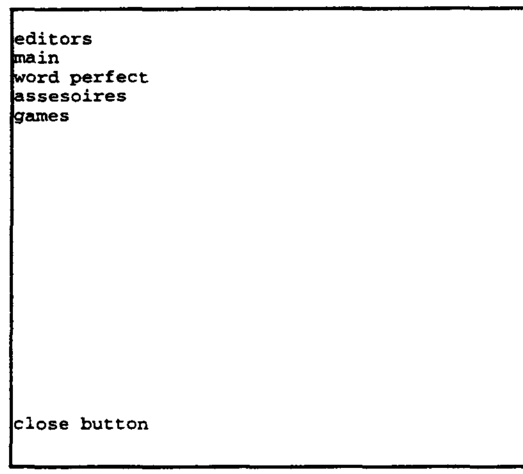


(G); And 'the program manager'.

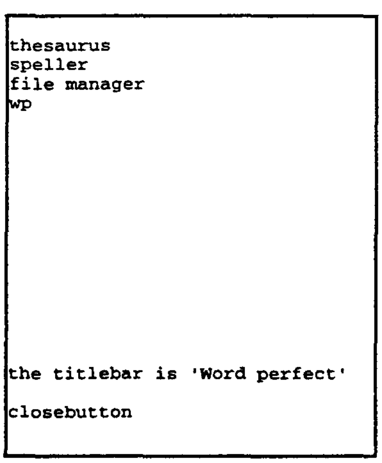
Figure 9: Windows in 'the depth version' of the VISA-com system.
The windows which should appear when performing the first application search task. As told before, the windows (A) and (G) are 'the program manager'. The windows (B) and (F) are the 'application group icons, the windows (C) and (E) are an 'application menu' and the window (D) is an application ('word perfect').



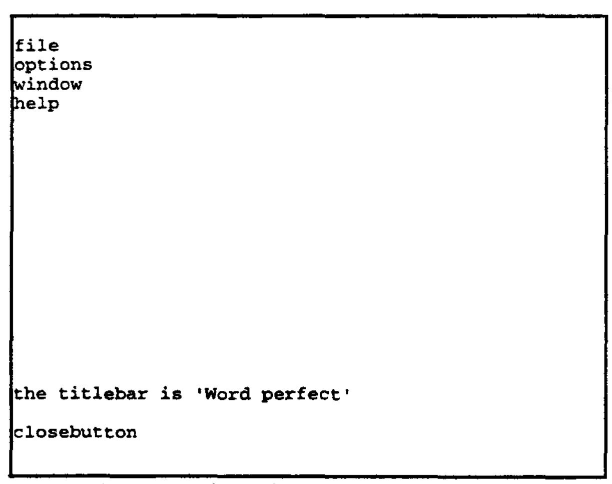
(A); 'The program manager'



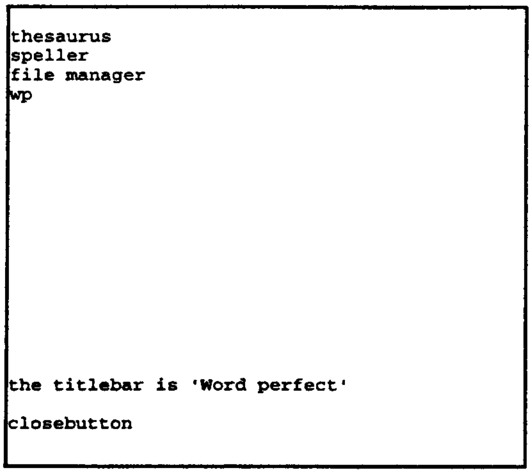
(B): the application group icons in the 'program manager'.



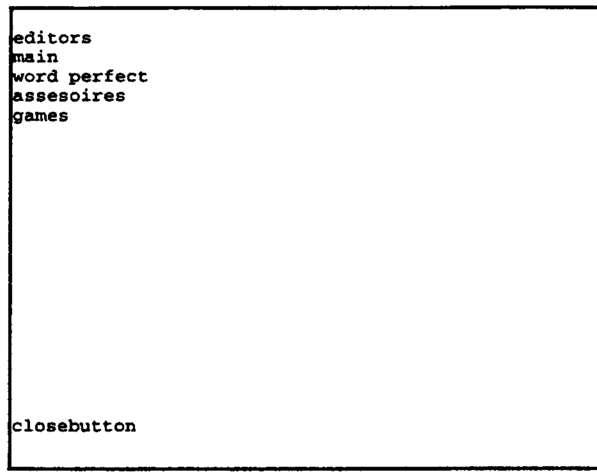
(C); the application menu.



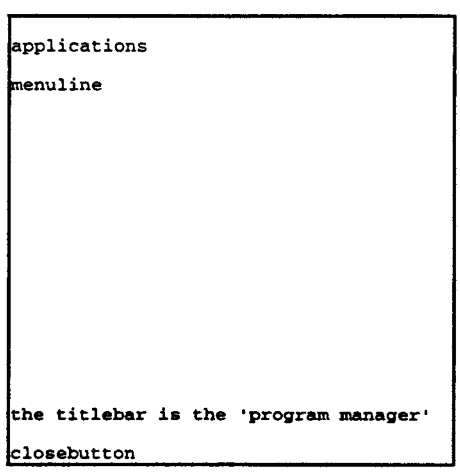
(D); The application 'word perfect'.



(E)' The application menu.



(F); The application group icons.



(G); And again, the 'program manager'.

The principles used in the 'depth version' of the system

Most principles and the correction of these principles have been explained earlier. But if we build a prototype we have to use some extra principles. These principles are mentioned in appendix 3. An example of these principles is:

When there are different kinds of information on screen, separate the types of information with a blank line. Different kinds of information are, for instance, the menuline and a menu of applications. In this way it is clear what information does not belong to some other information.

Procedure

We started with an explanation of the window environment and gave some information about the interface of the system. After this instruction a trial section followed in which subjects could practice with the interface. In this trial section one application task was performed in the version of the system they had to start with. After this query the subjects could ask some questions about the interface or the Window environment. Then four queries were performed. The questions without scaled answers to evaluate the system concerned were asked after the performance of the four queries. After a little pause (depending on the subjects) a new trial session was started to make the subjects experiment with the other version of the system. Again, the subjects could ask questions and the last four tasks were completed. After having performed these queries the subjects were given the same questions without scaled answers to evaluate the version of the system they had just seen. Finally, they did the question with the scaled answer to compare the systems. There was also room for some remarks, if subjects wanted to say something about the interfaces of the systems.

4.3 Test results

Comments of subjects

Some remarks about both screen designs are the following (see appendix 3). Seven subjects said that they would rather work with the depth version, because they can find the information more easily and they understand the structure of the system better. However, three subjects argued that they would want to work with the breadth version, because they want to work with the same screen design as their sighted colleagues, for then they would be able to get assistance from them.

Methods of analysis

We analyzed the behavior of VIPs working with the VISA-computer simulation with the tapes and then analyzed these results with the GOMS analysis of the first experiment. We had to correct this GOMS analysis slightly for both versions of the system and we will explain these adoptions.

The information search strategies of VIPs in 'the breadth version' and in 'the depth version' will be discussed different sections. We made a section in which we compared the 'breadth version' with the `depth version' and a section to compare the first and third experiments.

However, in both 'the depth version' and 'the breadth version', the same measurements were made. These measurements concerned the following: how much time did they need to perform a certain query (measured like in the first and second experiment), how frequently did they evaluate automatically, how did they make an evaluation (by looking at the title bar or at the overall screen design), how frequently did they interrupt the information spoken by the speech module, how much extra information from the tutorials did they use, how much time did they need to explore the screen and how frequently did they do so. We will describe these frequency measurements in table s 3 and 4.

4.3.1. Search and information use strategies in the 'breadth version'

In this section we will discuss the search strategy of visually impaired people in 'the breadth version' of the VISA-computer simulation.

Table 3: Mean Frequencies of recorded events in 'the breadth version'

Event Type	Mean	Standard Deviation	Standard Error
Exploring the screen	5.4	2.1	0.8
Evaluate automatically	5.0	0.5	0.3
Look at the title bar	4.0	0.8	0.3
Look at the overall screen design	0.8	1.9	0.9
Extra information is needed	16.0	10.7	3.8

Query time

The time subjects needed to perform the four queries in 'the breadth version' had a total duration of 1409 +/- 250 seconds i.e., almost 3 minutes per query.

Extra information

VIPs also used extra information (see experiment 2) in this version of the system. Extra information was needed 16.0 +/- 10.7 times.

Search in the program manager

The subjects always started to think about a menu or menu item desired (as in experiment 2). For instance, if they had thought of a desired menu first, they were capable of moving the vertical slider straight to the desired text line (A subject has described this as follows: "I know that the menu desired is the third text line, so I move the vertical slider to the third low beep.").

A line is defined as the area in which you can move the horizontal slider while the vertical slider is in the same position. If the information can be found with the horizontal slider, the line is called a text line. If no information can be found, the line is called an empty line.

If the subjects had no relevant experience and therefore could not think of a menu desired, they started searching with the sliders. In this article we called this 'explore the screen'. We will describe this later on in this section.

We observed that the search strategies in a menuline and in the application menu are identical, because they have the same structure (described earlier). This is why we observed that the subjects searched with the horizontal slider from left to right.

Search in a pull down menu

When a menu item was selected, a pull down menu appeared. It should be mentioned first that the subjects had many difficulties understanding this change of the screen. We will explain this problem later on in the discussion of the experiment. When we gave the subjects some more information about the change on screen they understood the structure of the pull down menu.

The subjects used the following search strategy. They tried to use earlier experiences, like searching an menu itme in the program manager. If the subjects had no relevant experience, they searched the menu systematically. They moved the vertical slider one position down and read the item. When the item was the desired one, they selected it. If not, they moved the slider to the next item and so on, until the desired item was found.

The use of a dialog box

When a dialog box appeared unexpectedly the subjects had big difficulties to handel it. They did not see a screen change, but they were confronted with a changed screen when they read the screen with the sliders. The subjects were surprised, which can be illustrated by remarks like: "Such funny things happen!' and: "Is this a pit fall?".

They react on a unexpected dialog box like well sighted people. When they had found a question, they looked further, and found the buttons and selected the desired one.

In some cases (e.g., when subjects were searching for a glossary) the dialog box was not unexpected. In this situation they had no problems to handle the dialog box. They read and chose this information like the search strategy in a pull down menu.

When a scroll bar was on screen, they had a lot of difficulties to use it because They could not find this scroll bar (the position of the scroll bar is illogical for a VIP).

Search in an application menu

An application menu looks like the program manager, except for the menuline (because no menuline exists in the application menu). Therefore it is not strange that we observed that this window is treated just like the program manager.

Search in an application

We also observed that the same search strategy was used in an application as in the program manager. The difference was that, when searching a menu, they could not come across applications, but text or option keys. Text could be read directly by pushing the vertical slider, if it contained one line. When text contained more lines, they moved the vertical slider. We observed that the search among option keys is similar to the search among menu items.

Evaluating the results

VIPs evaluated the results of a (partially) performed action in the same way as sighted people by looking either in the title bar 4.0 +/- 0.8 times or to the overall screen design as a whole 4.3 +/- 1.9 times, while both could happen with a frequency of 8 times maximally. They evaluated automatically 5.0 +/- 0.53 times, while the frequency of evaluating automatically could happen 8 times maximally.

Exploring the screen

The subjects explored the screen when they did not know where to find the information. But also some subjects explored the screen when they had seen the desired item, because they wanted to know how the screen looked like.

We observed that the subjects could explore the screen in two different ways. The first was a broad search, where only the vertical slider was moved and only could be heard which information was presented on a certain text line. The second was a precise search, were both the horizontal slider and the vertical slider was used to read all information presented on screen. Because no clear structure is found between these two methods, we have not distinguish these two methods in our measurements. Exploring the screen was done 5.4 +/- 2.1 times, with a duration of 267 +/- 120 seconds, i.e. about 50 seconds per exploration.

Return to the program manager

When the query was executed, the subjects returned to the program manager, as was told in the query. To do this, they had to leave some menus and windows.

A pull down menu is left by clicking outside the menu. The subjects had difficulties with this action, because they did not know immediately if the menu had disappeared.

A window disappeared if the closebutton was activated. We observed that this caused little difficulties, because the subjects could evaluate the result of this action quickly by moving the horizontal slider and then read the title bar.

Control all screen changes

We observed that the subjects had very manu difficulties to control the screen changes, because they had no overview. They asked constantly if a screen change had occurred. But they checked the screen changes in the same way as their well sighted colleagues (e.g. evaluation of (partially) performed queries and leaving windows which were not preferred).

4.3.2. Search and information use strategies in the 'depth version'

Here we will discuss the search strategy of visually impaired people in the depth version of the VISA-computer simulation.

Table 4: Mean Frequencies of recorded events in the 'depth version'

Event Type	Mean	Standard Deviation	Standard Error
Exploring the screen	2.5	1.7	0.6
Evaluate automatically	6.0	1.9	0.7
Look at the title bar	4.6	0.9	0.3
Look at the overall screen design	1.1	1.1	0.4
Extra information is needed	2.3	2.1	0.8

Time used

As in 'the breadth version' of the VISA computer simulation 4 queries were performed in 'the depth version' with a total duration of the tasks of 981 +/- 333 seconds.

Extra information

As in the 'breadth version', the subjects needed extra information 2.3 +/- 2.1 times.

Search in a menu.

Because in this version the windows and menus have nearly the same structure (described above), like a pull down menu in 'the breadth version', we observed that we can describe the search strategies in 'the depth version' in terms of a menu. We also observed that the applications, application groups, menu items and option keys were handled in the same way. Only the way to leave a menu in the `depth version' was different (described below).

The search in a menu was done by selecting an menu item or application directly (trying to use earlier experiences). If they had no relevant experience, the search was done systematically: the subjects moved the vertical slider one position down and read the item. If the item was the desired one, they selected it. If not, they went to the next item by moving the slider one position down, they read this item, and so on.

Use of a dialog box

In this 'depth version' the subjects also had difficulties to handle a dialog box which appears unexpectedly with a message. The reason is the same as in the 'breadth version', namely that they were confronted with a changed screen instead of actually seeing a screen change.

The subjects handled a dialog box the same way as described in the previous section, except that a scroll bar was searched differently; it looks like the search of an item in a menu. We observed that subjects had no difficulties to handle the scroll bar.

Evaluating the results

We observed that the evaluation of the results took place in the same way as in the 'breadth version', namely by looking at the title bar (4.6 +/- 0.9 times) or looking at the overall screen design (1.1 +/- 1.1 times), where they evaluated automatically 6.0 +/- 1.9 times. The maximal frequency of evaluating automatically was 8 (as in 'the breadth version') and the maximal frequency of looking at the title bar or looking at the overall screen design could both be 8 (also as 'in the breadth version').

Exploring the screen

We observed that in this version the screen was also explored. The first method was a broad search, just to find out where the text lines are on the screen. In this case they moved the vertical slider and listened to the beeps. A low beep indicates a text line and a high beep an empty line. The second method was a precision search, reading all items on the screen. Again we have not made a distinction between these two kinds of search for the same reasons. The mean duration of exploring the screen is 40 +/- 30 seconds and the mean frequency to explore the screen is 2.5 +'- 1.7 times.

Return to the program manager

When the task had been executed, the subjects returned to the program manager by selecting the closebutton. Every window (menu) contains a closebutton. The closebutton was searched in the same way as a menu item.

Control all screen changes

In this system the subjects also had difficulties with screen changes, but less than in 'the breadth version', because all screen changes looked the same. They controlled the screen changes in the same way as in the breadth version.

4.3.3. Comparing the search strategies of both versions

Method of analysis

In this section, we will compare the measurements of 'the breadth version' with the measurements of 'the depth version' by subtracting the averages of the two versions in table 5. We will look at the differences between the frequency measurements by subtracting the measurements of 'the breadth version' from the measurements of 'the depth version' and thus finding the significant difference between the two versions.

Again, the level of significance indicates how clear a difference there was in our sample and does not serve as strong evidence that the difference could be found in another sample.

Table 5: comparing 'the breadth version' and 'the depth version'.

Event Type	Mean	Standard Deviation	Standard Error	Prob>|T|
Exploring the screen	2.9	3.1	1.1	0.030
Evaluate automatically	-1.0	2.2	0.8	0.240
Look at the title bar	-0.6	1.5	0.5	0.279
Look at the overall screen design	-0.3	1.3	0.5	0.598
Extra information is needed	13.7	12.0	4.3	0.015

The performance time in 'the breadth version' is not significantly higher than in the 'depth version'. However, the standard deviation in 'the depth version' is very high and this decreases the chance that there are significant time measurements. This may be the case in another sample.

The time and frequency of exploring the screen is significantly higher in the breadth version with a paired t-test ($p < 0.05$). The frequency in which the subjects needed extra information also differs in both versions. In 'the breadth version' subjects needed significantly more information than in 'the depth version'.

4.3.4. Comparing the second and third experiments

Search strategy

The most important difference between the search strategies in the second and third experiment was that the simulator of the second experiment determined the search for information for an important part. When subjects asked him to read the screen, he read all information from top to bottom, while the search strategy in the third experiment was determined by the sliders and the different screen designs. Furthermore, in the third experiment subjects could explore the screen not only by reading all items (as in the second experiment) but also to make a broad search by moving the vertical slider and hearing all beeps.

Methods of analysis

We measured differences in the way subjects evaluated the results in the third experiment twice as often as in the second experiment. Maybe this result is caused by the fact that in the second experiment subjects had no interaction with a computer, but with a real person - which the subjects thought would evaluate their actions directly - in contrast to the subjects of the third experiment.

The evaluation of the results was also different. In the second experiment no big difference was measured in the way in which subjects evaluated the results: looking at the title bar (2.3 times) or looking at the overall screen design (4.3 times). However, in the breadth version there was: subjects looked at the title bar 8.6 times and at the overall screen design 2.0 times.

4.3.5. Comparing the first and third experiments

Search strategy

The most important difference in the search strategies of these two experiments is that sighted users looked in the most conspicuous menu when they had no relevant experience. In contrast, the search strategy of VIPs depends on the sliders; they explored the screen with the sliders if they had no relevant experience.

Because in the third experiment subjects got no information about screen changes, they were surprised to see the screen changed sometimes, and then had to explore the screen, whereas in the first experiment the subjects could handle a screen change the moment this screen change occurred.

Differences between methods of analysis

The way in which subjects evaluated the results was also different. In the first experiment, subjects looked more often at the overall screen design to evaluate the result (8.6 times) rather than looking in the title bar (2.0 times). But in the third experiment subjects looked more often at the title bar to evaluate the result (8.6 times) rather than looking to the overall screen design (2.0 times).

4.3.6. Comparing the second experiment with the depth version of the third experiment

Because the only difference between the second experiment and the depth version of the third experiment is the interface (in the second experiment a simulator was used and in the depth version of the third experiment a Window simulation of the VISA com was used (with the TASO screen and speech module)} it would be interesting to compare them to see which differences occurred in search strategies, ways of evaluation etcetera.

However, it should be noticed that some interaction effects occurred because only four questions were made with the depth version of the VISA-com simulation, while eight questions were made with the simulator in the second experiment. Another confounding variable was that in the third experiment the two versions of the VISA com simulation were tested and the results of the breadth version could have influenced the results of the depth version. This is the reason why we have not make this comparison. This comparison is good for further research.

4.3.7. Comparing the first experiment with the breadth version in the third experiment

The reason why it is interesting to look at the results of this comparison is that the only differences between these experiment were firstly, the interface (using the VISA-com simulation in the third experiment and using the Windows 3.1. environment with use of a mouse) and the second difference is that in the first experiment sighted subjects were used and in the third experiment visually impaired users. We could check how these differences have their influence on the search strategy and way of evaluating the results.

But the same confounding variables took place as in the comparison between the second experiment and the depth version of the VISA-com simulation. Again this is the reason why we have not made this comparison and leave this for future research.

4.4. Conclusion

In this experiment, we saw the two versions of the VISA-computer system, both with their own screen design. We saw that VIPs search their information in a different way in both versions. In 'the depth version' we saw a more systematic search than in 'the breadth version', because the subjects did not know at which location they could find the information on screen, whereas in the depth version they did know.

This can be concluded, because in the `breadth version' the screen has to be explored more frequently and it takes more time, and more extra information is needed to find out how the window is divided. The way of evaluating the results is different as well.

Because we observed in the breadth version subjects can only tell how they have evaluated the result after the evaluation, while in the breadth version users tell how they will evaluate the result before the evaluation, we conclude that the evaluation of a result occurs more systematically in the 'breadth version'. However, we have observed these reactions of the subjects during the experiment, so we have not done any systematical research of this phenomena.

So we think that the difference between the evaluation of the result in 'the depth version' in contrast to 'the breadth version' is that in 'the depth version' the subjects knew how they could evaluate an executed action, while in 'the breadth version' they looked around desperately, and then they found some screen changes had taken place or they found the titlebar and they used this for the evaluation.

5. General Discussion

Summary of the experiments

In the first experiment we saw how sighted novice Window users searched for information in a Window environment. We described how they used Window objects, and their frequency of use.

In the second experiment we used both the principles determined in the first experiment and some principles described in literature about screen design (Hammond 1987) (Paap 1988) and we took the way of 'looking at' the screen of VIPs into account to develop a screen design which we expected to be more suitable for VIPs than the normal Window screen design. We saw some problems with some items when observing VIPs in this experiment. We corrected these items after the observations. We also determined the search strategy of VIPs during this experiment.

In the third experiment we determined the search strategies of visually impaired users in two different screen designs of a Window environment. We compared these search strategies and asked the VIPs which screen design they would use in which situation.

Comparing the experiments

When we compared the first experiment with the second experiment, we saw some differences in performance. The way in which VIPs evaluated the (partially) performed queries was different from the way sighted subjects evaluated the (partially) performed queries. Also we measured some differences in the time they used to perform the queries.

Because the same differences were measured between the breadth version and the first experiment and the depth version and the first experiment, the breadth and depth versions can be combined and contrasted with the first experiment. This means that we compared the first and the third experiment. The same applies to the second experiment, which we compared with the third experiment as well. Differences were seen in the way subjects searched information on screen if they had no relevant experience (sighted subjects looked at the most conspicuous information while the information VIPs used was dependent from the position of the sliders) and in the way they evaluated the results (VIPs looked more often at the title bar to evaluate the (partially) performed queries and sighted people looked more to the overall screen design).

Conclusion

In our experiment we found a partial answer to the question `what general principles of graphical screen design are best suited to the way in which VIPs perceive the world?'

We saw that a one dimensional representation gave a better understanding of the structure of the system and subjects were less tired after the experiment. In the frequency measurements we also saw that performance in the one dimensional screen design was better. Less time was needed to perform the queries in the depth version than performing queries in the breadth version. Subjects needed less extra information, spent less time to explore the screen and did so less frequently.

A good example of a bad understanding of how a two dimensional screen representation was given by the subjects were born totally blind. They had much difficulty to understand the working of the sliders of the TASO screen. They both thought that if they moved the horizontal slider, they could read the last line of the screen, and that the vertical slider described the position between the left corners of the screen (because the vertical slider was placed on the left side of the normal keyboard). They could not imagine that all positions on the screen could be described by these two sliders. However, we had only two users who were totally blind from their birth, and therefore it is interesting to make some more research on this point.

Other issues

A good example of a bad understanding of how a two dimensional screen representation was given by the subjects who were born totally blind. They had much difficulty to understand the working of the sliders of the TASO screen. They both thought that if they moved the horizontal slider, they could read the last line of the screen, and that the vertical slider described the position between the left corners of the screen (because the vertical slider was placed on the left side of the normal keyboard). They could not imagine that all positions on the screen could be described by these two sliders. However, we had only two users who were totally blind from their birth, and therefore it is interesting to make some more research on this point.

Many questions have not been answered in this article. Some questions have been mentioned above. Another question is, for instance, how to give information about screen changes, because we have seen that the VIPs did not receive feedback of any screen change in our system. A solution has been given by presenting sound (everyday sounds, or abstract sound; Gaver 1989). Yet another question is how the search of information on screen can be improved, for instance by using commands (Hammond 1987). It can also be checked if auditory icons (called 'earcons') lead to faster performance (Blattner et al. 1989). Auditory icons consist of certain sounds, so that, for instance, a particular sound can be heard when a new window appears. This speeds up the performance time of certain actions executed by VIPs, because they can hear the sound to check some events (e.g. screen changes) instead of having to move the sliders to a certain position (e.g. to read the title of the window) to see what has happened.

Concluding Remark

In this article we showed that a screen design which is well suited for sighted people is not automatically a good screen design for visually impaired users. Namely, a screen design depends on how you 'look at' the screen and this, in turn, depends on the way you 'look at' the world. This should be taken into account when one designs an interface for visually impaired users.

6. Literature

1. Arat", A. & Vasp^Tri T. (1992). An intelligent screenreader integrated with synthetic speech for the blind. *Computers for handicapped persons. Proceedings of the 3rd International Conference* on computers for handicapped persons, 8 - 13.

2. Blattner, M.X. & Summikawa, D.A. & Greenborg R.M. (1989). Earcons and icons: Their structure and common design principles. *Human computer interaction*, Vol. 4, No. 1, 11 - 44.

3. Blenchorn, P.L. (1992). Requirements for screen access software using synthetic speech. *Computers for handicapped persons. Proceedings of the 3rd International* Conference on computers for handicapped persons, pp 31 - 37.

4 Card, S. K., Moran, T. P., & Newell, A. (1980). Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology, 12,* 32--74.

5. Edwards A.D.N. (1990). Adapting the Mackintosh and other graphical user interfaces for blind users. *Proceedings of the 6th. international workshop on computer applications for the blind.* Vol. 4, No. 3, 1 - 9.

6. Edwards A.D.N. (1992). Graphical user interfaces and blind people. *Computers for handicapped persons. Proceedings of the 3rd International Conference on computers for handicapted persons,*114 - 119.

7. Edwards A.D.N. (1989). Soundtrack: an auditory for blind users. *Human Computer Interaction,* Vol. 4, pp 45 - 56.

8. Ekberg, J. (1990). Some notes on European research for the visually disabled. *Access to visual computer information by blind persons,* 11 - 27.

9. Gaver (1986). Auditory icons using sound in computerinterfaces. *Human computer Interaction,* No. 2, pp 167 -177.

10. Gaver, W.W. (1998). The Sonic finder: an interface that uses auditory icons. *Human Computer Interaction,* Vol.4, pp 67 - 94.

11.. Hammond, N. (1987). Principles from the psychology of skill acquisition. In M. M. Gardiner & B. Christie (Eds.), *Applying cognitive psychology to user-interface design.* Chichester: Wiley.

12. Knaapen, M. (1986). *Tasten in het duister, voelen in het licht: Een haptonomische studie over mensen die blind zijn in een wereld van mensen die zien.*(Chapter 7 :'Aspecten bij het waarnemen van een ruimte' , pp. 73-97). Bloemenbergen Sante B.V., Nijmegen.

13. Marshall, C., Nelson, C., & Gardiner, M. M. (1987). Design guidelines. In M. M. Gardiner & B. Christie (Eds.), *Applying cognitive psychology to user-interface design.* Chichester: Wiley.

14. Paap, K. R., & Roske-Hofstrand, R. J. (1988). Design of menus. In M. Helander (Ed.), *Handbook of human-computer interaction[.* Amsterdam: North-Holland.

15. Pollatsek, A. & Rayner, K. (1989).The psychology of reading.(Chapter 4: *'The perceptual span'*). Prentice Hall, Engelwood Cliff.

16. Tzeng, O.J.L. & Singer, H. (1981) Perception of print. Laurence Erlbaum Hellsdile. Nickerson, R.S. Speech, *Understanding and reading: Some differences and simularities* (pp. 257 - 289).

17. Weber, G. (1992). Adapting graphical interaction objects for blind users by integrating braille and speech. *Computers for handicapped persons. Proceedings of the 3rd International Conference on computers for handicapted persons,* 546 - 555.

Appendix 1

1. Queries in the first experiment

The queries can be divided into two groups: the trial session and the main session. The trial session is asked at the beginning of every experiment and consists of two queries. The main session consists of 9 queries: 4 application search tasks, 4 option search tasks and one filler task.
The experiment was conducted in Dutch, so the queries are in the Dutch language.

1.1 Trial session

Query: "How many windows does your parents' house have? We don't want to know just the answer, but we'd like to know how you are counting in your head. You have to talk continuously."
Query: "Activate 'editors' in the 'program manager'."

1.2. Main session

Application search tasks
Query: "Go to the application Word perfect. When you have started Word Perfect, go back to the 'program manager'."
Query: "Go to Borland C++. Choose the workshop of Borland C++. When you have reached the workshop, go back to the 'program manager'."
Query: "Go to the editor 'Write'. When you have started up this editor, go back to the 'program manager'."
Query: Go to the game 'back gammon'. When you have activated this game, go back to the 'program manager'."

Option search tasks
Query: "You want to make a new icon. Therefore, you have to search the item 'New program object'. When you have found this item, go back to the 'program manager'."
Query: "Search for the item, that saves the settings on exit. Activates and deactivates this item."
Query: "You don't know the meaning of the word 'information area'. This is a word which is used in Windows. Therefore, you have to search for the glossary in the help of Windows. In the glossary, you can found this word. What is the meaning of 'information area'?"
Query: "Search for the item 'cascade'."

Filler task
Query: move the icon 'editors'

2. Queries in the second experiment

The queries in the second experiment are like the queries in the first experiment. But some differences can be mentioned. First, we skipped one query in the trial session - "Hoeveel ramen heeft het huis van uw ouders?" - because the subjects in this experiment were forced to speak aloud, because they had to give instructions to the simulator. So they had to verbalize their thoughts directly and an exercise to train this behavior is not necessary.
Second, we skipped the filler task, because this query is a visual task and can not be performed auditorially.
We also changed a word in a query , because the word had proven to be unclear. We changed the word 'icon' into 'file' in the query "You want to make a new icon. Therefore, you have to search the item 'New program object'. When you have found this item, go back to the 'program manager'."

3. Queries in the third experiment

In the third experiment we used the same queries as in the second experiment. We had to change one query, because in that query the application 'Borland C++' had to be activated, and we could not implement 'Borland C++' in our VISA-computer simulation. We also added a checklist to the queries that had to be performed after the queries.
We had to change the application search task query: "Go to Borland C++. Choose the workshop of Borland C++. When you have reached the workshop, go back to the 'program manager'" into "Go to the calendar. You can find the calendar in the group of applications 'accessories'. When you have found the calendar, go back to the 'program manger'".

Checklist

We used the following questions. The first and second questions were asked after the subjects had performed the tasks in the 'depth version' and were again asked after they had performed the tasks in 'the breadth version'. The other questions (3 and up) were asked after the subjects had performed all queries, in both 'the depth version' and 'the breadth version' of the VISA computer simulation.
Again, all questions were asked in the Dutch language. Questions 1 to 4 were open questions, which means that they could give a short answer using their own words. Questions 5 to 10 were questions with a scale. In this case subjects could choose between 7 answers:
1. much better in version 1.
2. better in version 1.

3. a little bit better in version 1.
4. no difference.
5. A little bit better in version 2.
6. Better in version 2.
7. Much better in version 2.

The open questions are:

1. "What are the shortcomings of this version?"
2. "What do you like about this version?"
3. "What do you think of the sliders?"
4. "How do you think the system could be improved?"

The `scale'-questions are:

5. "What version is easiest to work with?"
6. "What version requires most effort?"
7. "In what version is it easiest to find the applications required (= programs, like WordPerfect and Back gammon)?"
8. "In what version is it easier to work with the sliders?"
9. "In what version is it easiest to perform the item search tasks (like 'save settings on exit') ?"
10. "The structure of which version do you think you see through best?"

Appendix 2

Principles regarding the simulator

The 'simulator' should verbalize the windows in the Windows 3.1 environment according to the following principles:

1. If a window contains menu itmes and one or more other kinds of information (e.g. text), he gives a summary of the information on the screen in the following way:
1.1. If there is text on the screen, he says 'text'.
1.2. If there is an application menu, he says 'applications'.
1.3. If there is a menuline, he says 'menuline'.
1.4. If there are option keys, he says 'option keys'.
1.5. If there is some information from other windows, he says 'information from other windows'.
1.6. If a window has a name, he says 'title bar' and tells the user the name of the window.
1.7. At the end of the menu, he says 'close'.

2. If a window contains only one kind of information (e.g. menu itmes, text or option keys) he gives the information in the following manner:

2.1. He tells all items, applications or option keys.
2.2. If there is a 'cancel' or 'o.k.' button, he says 'cancel'or 'o.k.' respectively.
2.3. If there is a title bar, he says 'title bar' and reads out the name of the window.
2.4. He mentions the item 'back'.

Appendix 3

Principles regarding the depth version

The principles which should be taken into account when building the screen design of the 'depth version' are:

1. When there are different kinds of information on screen, separate the types of information with an empty line. Different kinds of information are e.g. the menuline and a menu of applications. In this way it is clear which information does not belong together, because people expect information that belongs together to be grouped together.
Example:

----

applications

menuline

information from inactive windows.

----

2. When there are different pieces of information of trhe same kind on screen, make no empty line between them. The same kind of information is e.g. information in a pull down menu or information in an application menu.
Example:

----

editors
games
Word Perfect
assesoires.

----

3. According to FIT's law, we use the last position of the vertical slider as the close button. The time to travel (to reach the close button) is long, but the position is easy to find.

4. Above the close button is an empty line and above it the title bar of the window. Thus, you can look in which window you have arrived very quickly.
Because the distance between the title bar and the close button is very small, the time required to find the title bar is limited (FITs law).
Example:

----

work field

menuline

The title bar is Word Perfect

closebutton

Appendix 4

An example of a GOMS analysis.

Search in a pull down menu

We will give a GOMS analysis that describes how well sighted people search in a pull down menu. The aim of this example is to get an idea how a GOMS analysis can be used to describe the information use strategies of the subjects.

Subjects tried to use earlier experience, by selecting the information desired directly). When they were not able to use earlier experiences, they searched systematically trough the pull down menu by reading all menu items from top to bottom until they had found an item desired. then they selected this item

Goal: Search for an item within a pull down menu and select it.;if an pull down menu has been selected.
- Goal: Think of a desired item and select it. ;if you have relevant experiences.
-- Goal: select the desired item .
- Goal: Search trough the menu from top to bottom and select the item. ;if you have no relevant experiences.
-- Goal: look at the first item.
-- Goal: select the item. ; if it is the item desired
-- Goal: Look at the next item. ;if the last item is not selected.
-- Goal: analog to goal: select...item.'
-- Goal: analog to 'Goal: look...item.' ;till success or till all items are read.