Adaptive Bar Implementation and Ergonomics

Matjaž Debevc, Rajko Svečko, Dali Donlagić
Faculty of Technical Sciences
Smetanova 17, 62000 Maribor, Slovenia
E-mail: debevc@uni-mb.si

Beth Meyer
AT&T Human Interface Technology Center
500 Tech Parkway, N.W.
Atlanta, GA 30313, USA
E-mail: Beth.Meyer@AtlantaGa.ncr.com

Keywords: Adaptive User Interfaces, Intelligent User Interfaces, Human Factors, Icons Design

Edited by: Matjaž Gams
Received: March 9, 1994 **Revised:** June 20, 1994 **Accepted:** September 22, 1994

Self-adjusting, adaptive user interfaces offer automatic customisation of the computer-based working environment by checking users' procedures and typical decisions, eventually offering them adaptations or enhancements designed to make their individual work patterns easier and more efficient. This means that the users don't need deeper understanding of the application environment or its procedures, since the adaptive user interface itself recommends solutions and possible adjustments.
This article classifies user interfaces and their roles. Positive and negative aspects of adaptive user interfaces are also discussed. Using the adaptive bar as an example, we discuss the implementation and ergonomics of the adaptive bar, which represents the adaptive part of the interaction. During the working sessions, but without disturbing them, the user interface suggests the addition or removal of command icons and their resizing depending on the priority, which is based on the frequency of use. The article also offers a convenient solution to present the priority of icons.

1 Introduction

Self-adjusting (in following text called 'adaptive') systems offer automatic adaptation of the working environment by keeping track of user's procedures and decisions, and eventually offer the possibility of adjustment or easier ways to doing something (Geiser 1990). The user therefore doesn't need detailed knowledge about the working environment and possible procedures, since the adaptive system itself recommends best solutions and optimisation of repeating procedures. The same goes for adaptive user interfaces in human-computer interaction, since, for computer applications, the user interface is the user's working environment.

Graphical user interfaces, which represent an important component of current computer systems, are often the most sensitive elements of the communication. Advances in computer controls and displays make it possible for computers to be more widely accessible and lead to new, innovative interaction methods. One of these methods is interface adaptation, which could widen the group of users who would potentially switch to computerised processing or controlling. An ideal computer system would automatically adapt to the present user by identifying problem areas and offering help for the present work, therefore lowering the stress level and necessary concentration.

As a result, there are many research projects going on in this field (e.g., Browne *et al* 1990,

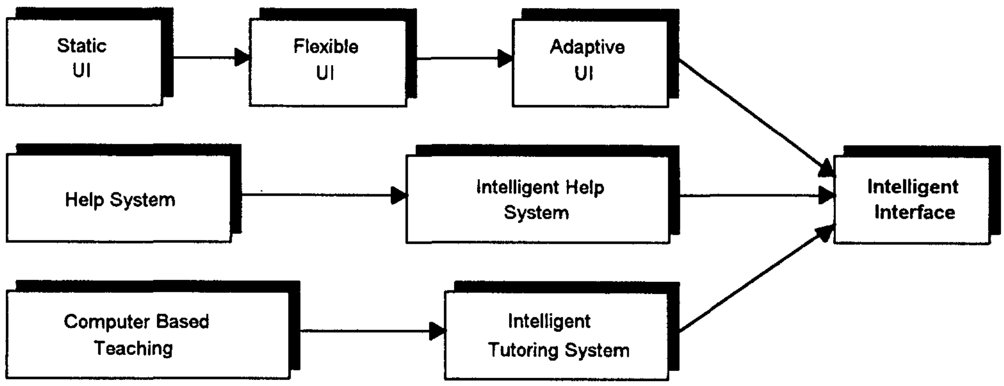

Figure 1: Adaptive user interface in relation to Intelligent Interface (in Kühme 1992)

Benyon and Murray 1988, Kossakowski 1989). The research is especially promising if we regard all the advantages of such a user interface in computer systems and applications, which are getting more and more complex. Studies such as those reported in Benyon and Murray 1988 and Kossakowski 1989 have demonstrated or implied significant human performance advantages for adaptive interfaces.

2 Classification

The user interface can be classified depending on the methods of adaptation. Kühme (Kühme *et al* 1992 and Hufschmidt *et al* 1993) suggests a scheme that represents multi-dimensional classification and is more convenient for representation of all existing viewpoints and prototypes of systems.

Figure 1 shows the scheme of classification of various existing user-interface concepts. The first generation of user interfaces was static; the system developer designed and implemented the user interface, and the user had to learn how to use it.

Later, flexible user interfaces began to appear. These user interfaces allow the user to make changes, although these changes are initiated and managed by the user. Today, many user interfaces are adaptable to at least some extent – for example, many offer the possibility of changing colours or resizing and moving the windows.

Adaptive user interfaces actively change according to conditions and user needs, either automatically or with user input. For example, an adaptive system might suggest and provide a special tool to perform a set of tasks that the user frequently performs together. If the functionality and demands of the application also adapt, the system can get quite complex.

According to Figure 1 the intelligent user interface is an assembly of adaptive user interfaces, intelligent help systems and intelligent learning tools.

Kühme has also defined **stages** and **agents** which influence the flow of the adaptation process. In every process of adaptation various tasks are executed. We can identify them as stages, which are carried out during the process of any single adaptation.

The **stages** are:	The **agents** are:
– Initiative	- System
– Proposal	- User
– Decision	
– Execution	

If the system is the agent that carries out all the stages we call it **Self-Adaptation**. In this case the system observes the communication, creates and evaluates various possible adaptations, and in the end selectss one of them and accomplishes it. Another type of adaptation is **User-Controlled Self-Adaptation**, in which the user determines whether the adaptation occurs and the system performs all of the other functions.

Systems that offer self-adaptation or user-controlled self-adaptation should also make it possible for the user to manually initiate the process of adaptation. These variations are called **User-Initiated Self- Adaptation** and **Computer-Aided Adaptation.** In the first case the user initiates the adaptation process and the rest is carried out by the system, whereas in the second case the user both initiates the process and decides whether to accept the suggestion.

In the last two combinations the user performs all of the stages except, perhaps, initiation. Depending on the agent that performs the first stage we have **System-Initiated Adaptation**, in which the system initiates the process, and normal **Adaptation**, in which all stages are carried out by the user. The simple adaptation makes it possible for the user to arrange the system according to his preferences and goals. Almost every window manager offers, for instance, colour palette customisation and changing of window size and menu display.

Generally, an information system may function as an **adaptive user interface** if the system acts as an agent in any stage of adaptation. This definition is taken from the user's point of view, so it does not specify anything about implementation or necessary information (Haaks 1992, Zeidler 1992).

3 Negative and positive aspects of adaptive user interfaces

An adaptive user interface adapts itself according to the present user and the present common procedures. Because of continuous changing of working environment and user's tasks there are negative and positive aspects of adaptive user interfaces (Norcio and Stanley 1989).

a) negative aspects

The most inconvenient aspect of adaptivity is its potential to prevent the user from developing a clear model of a system, if the system is changing all the time. For example, the system could require the user to learn a new procedure to accomplish a task, just as the user was beginning to learn the initial procedure. This aspect could reduce the user's productivity and confidence in the system, since it could keep the user from understanding how the system will behave.

Another problem with adaptive user interfaces is that the user can lose the feeling of competence or of being in charge. It can even happen that the user's actual goals and demands are concealed by attempts to control the user interface behaviour. Therefore the user interface should make it possible for the user to exert control over the working environment. The user interface also must not take the initiative away from the user, but should give him the best and most proper assistance with the present task.

A typical negative effect is when the system suddenly changes and the user is forced to interrupt his work and determine what happened (Shneiderman 1992). The user may start worrying, because he is unable to predict what will the next adaptation look like, when it will happen and whether he will be able to restore the original state. This is why it would be better for the user if the general appearance would remain the same and only a part of it would change, and even that only on his request (Debevc 1993, Debevc *et al* 1993).

b) positive aspects

On the other hand we can also list some positive aspects of introduction of adaptive user interfaces. For example, automatisation of systems is a domain which expresses certain needs for adaptive interfaces. A system that dynamically changes its tasks must be able to adapt to individual users. The tasks should be assigned to the user as well as to the computer. The way in which such an assignment would take place depends on who has better overview upon the information and procedures at work with the automatisation system.

The user who has a great amount of information available at the same time has to make a decision and choose one of these pieces of information. But we know that the user is not always capable of making decisions equally well and quickly. In order to be able to assure general optimal performance of the system it is necessary for the computer to be able to link the user's previous decisions and eventually show him only a few pieces of information, for which the probability of being interesting for the user is the highest at the moment.

Use of adaptive user interfaces is especially convenient for the growing group of users that do not have the time to gain deeper understanding on either the computers generally or the particular application they are working with. The adaptivity is most convenient for novice users who have to adapt to the new working environment as quickly as possible. Besides, the user is enabled to complete his task faster and more efficiently. Plus, the possibility of disappointment is lowered. For example, if an adaptive word processor user interface detects that a user is trying to print onto envelopes for the first time,, it could provide additional instructions for printing envelopes in unobtrusive help windows. These instructions could then be omittedafter the user has successfully printed onto envelopes several times.

The positive aspects of adaptive interfaces can be summarised in the goals of these interfaces. These goals are (Kühme 1992):

- easy, efficient, effective use
- making complex systems usable
- presentation of what the user wants to see
- faster use
- a user interface that fits heterogeneous user groups
- a user interface that considers increasing experience

The question exists, whether the positive aspects can outweigh the negative ones and to what extent are the adaptive user interfaces actually needed? If they are correctly and carefully designed, they can help in building more useful systems for treatment of larger amounts of information. In this way both beginners and experts can easily use the system. Adaptive interfaces help various types of users in making their work more efficient by showing them better ways to complete certain tasks and offer them the proper amount of support according to their individual needs.

4 Reasons for using the adaptive bar

One of the successful ways to adapt to the user can be found in the toolbar, which offers graphical representation of most often used commands and macros (Figure 2). The icons can be accessed quickly and simply; there is no need to spend time browsing through menus.

Figure 2: Toolbar in the Microsoft Word 2.0 for Windows^{TM}

The use of command icons is a relatively simple way to access various commands, but there are also some problems related to their use. The basic problem is in the size of the icons and poor overview of them, since we can barely distinguish between them on some screens because of their small sizes.

Although in some programs it is possible to adapt the bar to personal needs by adding or removing icons, changing the bar requires additional understanding of the structure of the program and the procedure for changing icons. Especially for the beginners, it can be a very difficult task to perform, since in many cases they cannot decide which commands will be used frequently and which they won't even need.

Especially for them and for others who often need a smaller number of repeating routine operations, we have designed the **adaptive bar**. It follows the frequency and manner in which the user performs individual commands, and, without interrupting the user's work, it suggests **addition** or **removal** of certain icons and shows their priority.

Perfect adaptivity is unfortunately impossible, but auto-adaptivity in our case means that the user interface tries to determine the priority of available commands. The bar is then adapted to the differing priorities, but the interface never acts on its own accord; it merely informs the user that an adaptation is possible. The user can then either confirm, deny or even ignore the suggestion. According to Kühme's classification, it is a **User-Controlled Self-Adaptation** system.

During the testing and empirical research (interviews, questionnaires) (Mayhew 1992) we determined that it is better for the end-user if the general appearance of the user interface remains static and only one part, which is simple and easy to use, is subject of change.

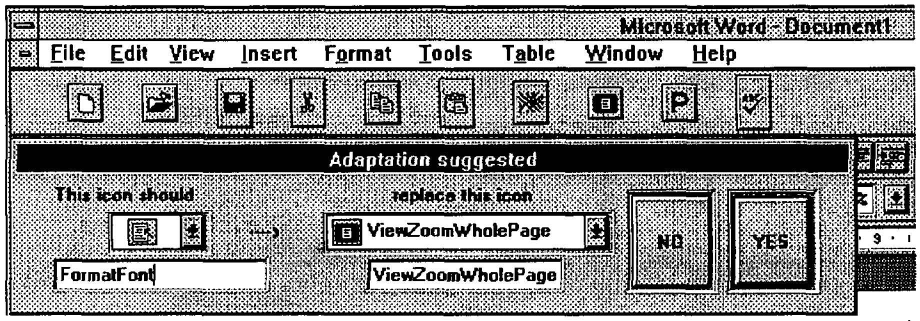

Figure 3: Adaptive bar with the dialog window

5 Adaptivity of the bar

The adaptive bar (Figure 3), which we have designed for Microsoft Word for Windows 2.0 (Microsoft 1991), has following characteristics:

- automatic addition and removal of icons
- automatic resizing of icons according to the frequency of use
- storing of present arrangement of the bar

The procedure of the **removal of a command icon** starts when the user interface, while measuring the frequency with which icons are used, determines that a particular icon hasn't been used for a pre-specified period of time. An indicator-icon, accompanied by a sound signal, appears and informs the user that a change to the bar is recommended. The **indicator of change** is represented by the background of the bar. The user eventually chooses the dialog window, which can be accessed by double-clicking on the background of the bar, and decides whether to accept the proposition. In this way we do not hinder the user's work dynamics. It is up to him to decide whether and when to change the bar. If the user ignores the proposition, the background remains in the warning state. This state is represented by the relief and intensive colour, distinguishable from the others. Further suggestions of changes, if not accepted, are stored and inserted in the waiting queue until the user decides about them.

For the user interface, the most complex and difficult operation is the **installation of command icons** in the bar. The system has to determine the frequency of commands and options from the menus and to decide which ones have to be inserted in the bar. A similar procedure is the determination of macro commands. In order to be able to determine the need for a change the user interface requires information about the application and its capabilities, the present user, previous events and the real-time system being controlled (Debevc *et al* 1992). On the basis of this knowledge it is able to check and compare user's actions. After a certain **frequency of repetition** it uses the indicator of change to inform the user that an adaptation would be suggested. After the adaptation is finished, which happens during the surveillance mode, the indicator of change returns into the passive state, which is represented by the flat appearance and lack of colour. Nevertheless, the bar-modification dialog box can still be accessed by double-clicking on the background of the bar.

The dialog window is designed to look simple and offer a clear overview of all its functions. The user can either place the suggested icon where adaptive bar recommends or he can instruct the bar to replace a different icon, simply by choosing a different destination icon in the dialog window.

The look of the bar can often change due to the adaptivity. Therefore the **storage of the present state** in a file has to be provided. In this way the adaptive bar can be used on the same system by various users with the possibility that everyone uses the bar developed during his own sessions. Of course, if there is a new user, the system opens the default type of bar, which can be either accepted or adapted to user's needs right away.

a)
b)
c)

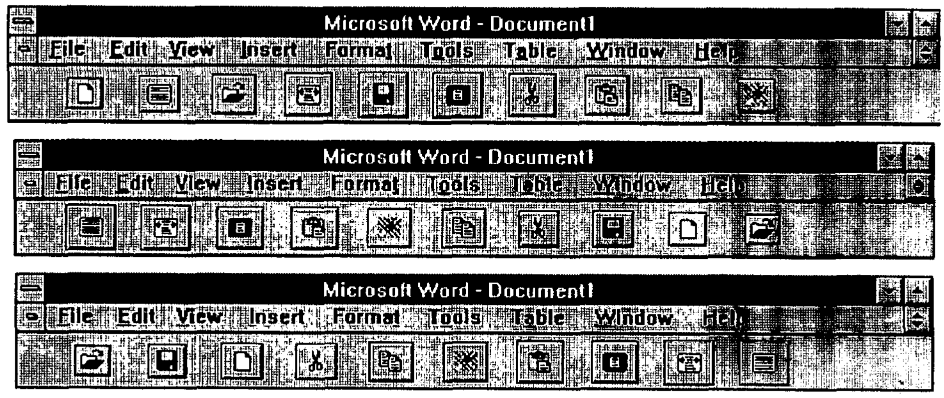

Figure 4: Models of the adaptive bar with varying positions of icons

6 Ergonomic point of view

In order to satisfy the ergonomic demands, which are described in ISO-standards (ISO-9241, 1992), we designed the bar in following two ways:

- Varying positions of icons:
- bigger icons in the centre of the bar, smaller ones on the sides
- bigger icons on the right side, smaller ones on the left side
- bigger icons on the left side, smaller ones on the right side
- Fixed positions of icons:
- varying widths and heights of icons
- varying widths of icons
- varying heights of icons

Even though we had to implement many different models of the bar, we used a single one and designed others quickly and simply by changing only four parameters:

- positions of icons: fixed or varying
- height range of icons (min. and max. height)
- width range of icons (min. and max. width)
- icon arrangement pattern

The testing sessions were followed by research using various prototype testing techniques (Mayhew 1992) (Structured observation, Benchmarking, Classic experiments). We also used an empirical research method with a questionnaire, written according to ISO-9241 Part 10 norms (Prümper 1993).

6.1 Varying position of icons

In the first test we introduced three models with varying positions of icons (Figure 4) to different kinds of users (novice users, students taking computer science classes, experienced users of interactive systems and experts). The first model has the most important icons placed in the centre of the bar and these are also the biggest ones, whereas other less important ones are getting smaller and are arranged towards both sides. The arrangement of these icons is very similar to the Gauss' random curve. The second model (Figure 4b) has the most important icons on the right hand side and the priority decreases from right to left side of the bar. The last model (Figure 4c) has the most important icons from left to right. In all three models the size of the icons decreases with their decreasing priority. The spacing between the middle points of icons are constant throughout the bar, which means that spacing between edges of bigger icons is smaller than the spacing between edges of smaller icons.

The results of the empirical research are shown in Figure 5. It is clearly visible from the table that toolbar "b" was the most convenient. The numbers represent the average number of points that the particular model gathered during the testing. The majority of subjects shared the opinion that this bar seems to be the most convenient because of its icon-arrangement from left to right, just the way we are used to read. In our findings, 24% of the subjects liked the changeable behaviour of the bar, while 72% of them preferred fixed positions of icons. Only 4% did not express a preference (Figure 6).

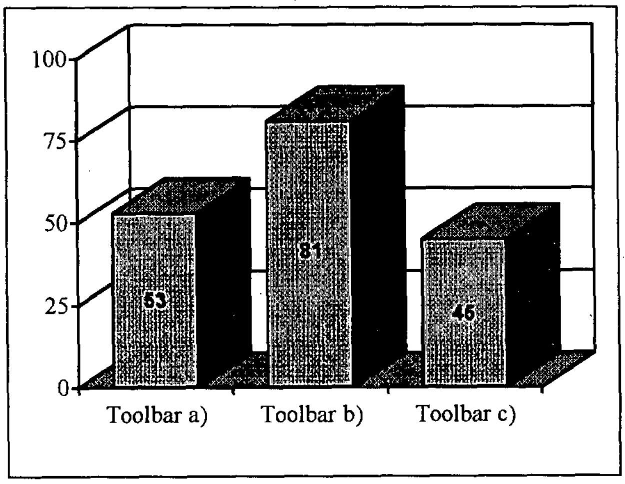

Figure 5: Results of the empirical research (collected points)

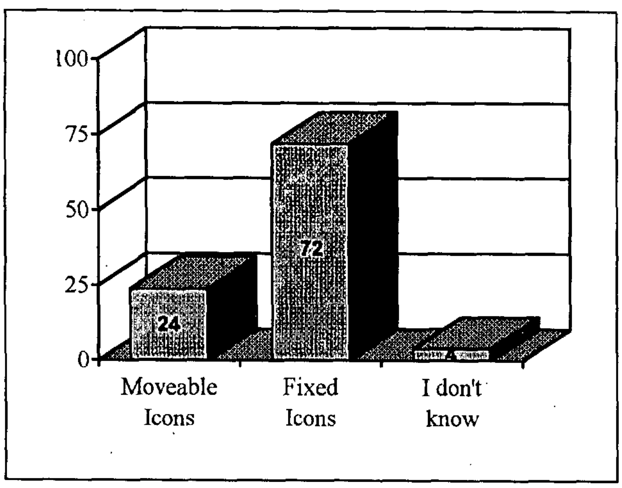

Figure 6: Results of the questionnaire about positions of icons (in percentages).

The results of the testing revealed this important fact: adapting the position of the icons was disliked by users and probably would not work well. It may be that users who are working with icons that change positions within the bar, even with their consent, would take more time to perform certain tasks and make more errors than the people working with icons in fixed positions. We make this prediction because such an adaptation requires the user to change his actions. It requires the user to move in a totally different direction to select an icon that he is already accustomed to finding elsewhere in the bar. This is a kind of adaptation that could cause problems for the user.

For example, if the user wants to select a certain icon that has just changed position, he would probably move the cursor to its former position, not to its current position. He then either selects the wrong icon, causing a time-consuming error, or he must make a second movement to get to the correct icon. Either way, the cost of moving the icon after it has been frequently used would be greater than the cost of leaving the icons unordered in terms of their priority. Whatever the order of the icons, it is likely that users will develop habits of moving quickly to the icons that they often use, perhaps without even looking at them.

Therefore the model of icons in fixed positions as long as they are in the bar is clearly better.

6.2 Representation of the priority of icons

The next step with testing was trying to find the best way to represent the priority of the icons, since it is not visible from their positions in the bar. The idea occurred to us, that it might be easier for the user to select his "favourite" icons if they are larger.

There are three possibilities:

– varying both width and height of icons

– varying width of icons

– varying height of icons

The most appropriate seems to be to vary the height of icons. The reasons for that are:

– When we change the width in any way, we must either change the position of icons as well, or we must have the spacing large enough in advance, which means potential waste of space.

– Increasing the height of an icon would probably make it easier to select it than changing its width.

However, these statements are based on the study of horizontal toolbar behaviour. For a vertical bar, the height is the dimension that should remain constant and width should be subject to change.

Returning to the horizontal bar, we can see that changing the widths of icons is impractical if we wish to keep them on the same positions. If we start with relatively small icons, and then increase their widths, we may have to move them as shown in Figure 7:

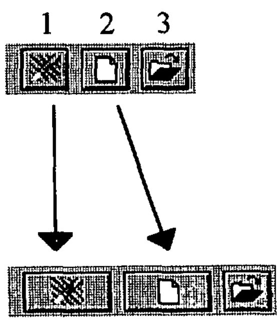

Figure 7: Changing widths and positions of icons after icons 1 and 2 are used frequently

It is clearly visible that icon 3 is almost hidden next to icons 1 and 2, which are much bigger, and is also moved far to the right.

Another possibility is to start with icons with very large spacing, in order to give them room to expand (Figure 8):

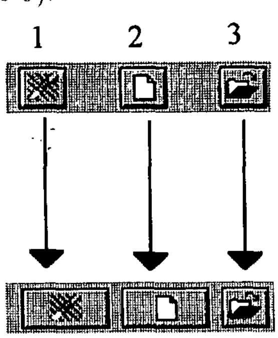

Figure 8: Changing widths of icons after icons 1 and 2 are used frequently

The later approach, while it does prevent the icons from changing their positions, greatly reduces the maximum number of icons in the bar. That is a problem, since a very useful feature of the adaptive toolbar is the possibility to add icons for frequently used commands.

This is the first reason for introduction of the third model. The second one is that the increase in height is likely to be more useful in making the icon easy to select than an increase in width. This prediction is based on Fitt's Law, the rule by which the difficulty of a hand movement can be predicted (Fitts 1954).

Fitt's Law states that the time required to make a movement is related to both the distance of the movement and the size of the target in the direction of the movement. The formula may be expressed in this way (from Bullinger, Kern, and Muntzinger 1987):

$\mathbf{MT = a + b^2 \log (2A/W)}$, where

MT is movement time,

A is "amplitude", or the distance from the starting to the centre point of the target,

W is the length of the target **in the direction of movement**(in the case of a circular target it would be twice its radius, not its area),

a and b are constants.

This means that, from the two movements below, which both start in the point 's' and finish in the centres of the targets, the one on the left is more difficult and takes more time to perform than the one on the right (Figure 9):

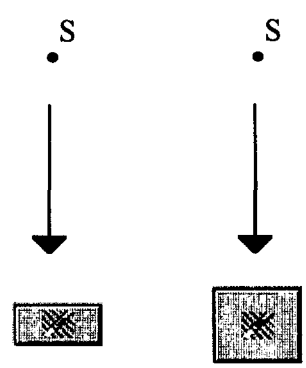

Figure 9: The right icon is easier to select that the left one

This is because the value of W for the target in the case on the right is larger, thus making the movement time (MT) smaller.

Furthermore, the two movements below should be equally difficult to perform (Figure 10):

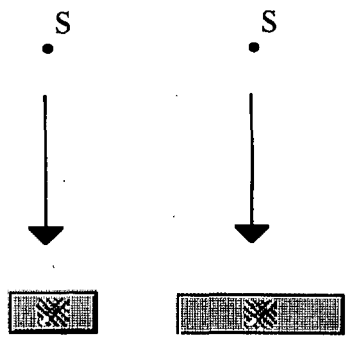

Figure 10: Equally difficult movements

This is because W, the length of the target in the direction of the movement, is the same for both cases. The extra width of the target on the right does not help when the movement starts directly above it. (It would help, however, if the movement started from the side of it.)

Someone moving the mouse cursor to an icon in the toolbar on the top of the application window, most probably starts moving it from a point on the screen well below the bar. In most cases, the movement will be primarily vertical. That means that increasing the height of an icon should have a greater effect on the ease of use than increasing the width of an icon.

One effect that increasing the width of an icon could have is that of reducing the distance that a user's hand must move to reach the icon. It could do this by decreasing the angle at which the user must move. However, this effect is unlikely to be significant. It is not practical to change the width of icons very much, for reasons given previously. Therefore, the vertical component of most movements is likely to be much greater than any possible change in icon width. For example, a movement from the center of the screen to a toolbar at the top of the screen might require 13 cm of vertical movement, while the maximum change in icon width might be about 1.3 cm. In a case where the center of an icon was 13 cm to the right or left of the center of the screen, and the user was moving to the icon from the center of the screen 13 cm below the toolbar, increasing the width of the icon by 1.3 cm would decrease the necessary movement distance by at most only 0,5 cm, or 2.5 percent. This figure is obtained by using the Pythagorean Theorem on two right triangles, one with sides of 13 cm (vertical distance) and 12.3 cm (horizontal distance), and the other with sides of 13 cm and 11.7 cm, respectively. This calculation compares two movements aimed at the very edge of the icon; if movements are consistently aimed at the center of the icon (the more probable case), the effect of a change in width is even smaller, since the center points of icons in fixed positions would always be at the same locations. Hence, it is reasonable to predict that increasing the width of an icon in a horizontal toolbar will not significantly increase its usability. However, this prediction should be tested in future experiments on the adaptive bar.

7 Conclusion

Generally speaking, adaptive user interfaces can be convenient for novice users, who have to learn quickly how to perform in an unfamiliar working environment. Since the user interface adapts itself to the user's knowledge and work, the user will soon lose the anxiety about the new environment and will start working on the actual problem more quickly. If the user interface is carefully and accurately planned it can be very useful for a wide group of users. These interfaces offer better efficiency of work by giving the proper amount of suggestions and help tailor information to different kinds of users and their individual preferences. Given the increasing complexity and functionality of computer systems, the benefit and the need for adaptable and adaptive user interfaces will also increase.

At testing of the adaptive bar, which represents the adaptive part of our user interface, the results have shown that the models which included changing positions of icons were unsuitable for the user's needs, since they were unable to remember and automatize movements for all the changes. The best solution appeared to be the model with fixed positions of the icons, in which priorities of the icons were represented by their height.

With help of the adaptive bar, we offer the user an opportunity to easily use the advantages of a self- adaptable user interface without being confused by changes, since the general appearance of the user interface remains unchanged. We also defined several principles for making the adaptibar ergonomically acceptable. These principles can be specifically tested in future studies of user performance with the adaptive bar.

References

[1] Benyon, D.R., Murray, d., 1988, Experience with Adaptive Interfaces, *The Computer Journal,* Vol. 31, No. 5, 465-473.

[2] browne, D.,Totterdell, P., Norman, M., 1990, Adaptive User Interface, *Academic Press*, London.

[3] Bullinger, H.-J., Kern, P., Muntzinger, W. F., 1987, Design of Controls, in *G. Salvendy (Ed.), Handbook of Human Factors, 577-600, John Wiley & Sons,*New York.

[4] Debevc, M., Donlagić, D., LEŠ, M., 1992, User Interface Styles, *Informatica*, Ljubljana, 16, 49-54.

[5] Debevc, M., 1993, Adaptive Bar, *INTERCHI '93, Conference on Human Factors in Computing Systems, INTERACT '93 and CHI '93, Adjunct Proceedings*, Amsterdam, 24-29 April, 117-118.

[6] Debevc, M., Donlagić D., SVEČKO R., 1993, Adaptive User Interfaces, *IEEE Electrotechnical and Computer Conference*, Portorož, Slovenija, September 27-29, 91-94.

[7] Fitts, P.M., 1954, The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, Vol. 47, No. 6, 381-391.

[8] Geiser, G., 1990, Mensch-Maschine-Kommunikation, *R. Oldenbourg Verlag*, München.

[9] Haaks, D. 1992, Anpaßbare Informationssysteme, *Verlag für Angewandte Psychologie,*Götinngen, Stuttgart.

[10] ISO-9241, 1992, Ergonomic requirements for office work with visual display terminals (VDTs), Part 10:*Dialogue principles – Second commitee draft.*

[11] Kossakowski, M., 1989, Adaptive Man-Computer Dialogues, in*F. Klix, N.A. Streitz, Y. Waern, and H. Wandke (Eds.), Man-Computer Interaction Research, MACINTER-II*, 201-212, *Elsevier Science Publishers*, Amsterdam.

[12] Kühme T., Dieterich, H., Malinowski, H., Schneider-Hufschmidt, M., 1992, Approaches to Adaptivity in User Interface technology: Survey and Taxonomy, *Proceedings of the IFIP WG2.7 Working Conference on Engineering for Human-Computer Interaction,*August 10-14, Ellivuori, Finland.

[13] Mayhew, D.J., 1992, Principles and Guidelines in Software User Interface Design, *Prentice Hall, Englewood Cliffs*, New Jersey.

[14] Microsoft Corporation, 1991, Microsoft Word for Windows 2.0, User's Guide.

[15] Norcio, A.F., Stanley, J., 1989, Adaptive Human-Computer Interfaces: A Literature Survey and Perspective, *IEEE Transaction on Systems, Man, and Cybernetics,*Vol. 19, No.2, 399-408.

[16] Prümper, J., 1993, Software Evaluation Based upon ISO 9241 Part 10, *Proceedings of the Vienna Conference Human Computer Interaction*, Austria, September 20-22, 255-265

[17] Schneider-Hufschmidt, M., Kühme, T., Malinowski, U., 1993, Adaptive User Interfaces: principles and practice, *Elsevier Science Publisher*, Amsterdam.

[18] Shneiderman, B., 1992, Designing the User Interface, Strategies for Effective Human-Computer Interaction, Second Edition, *Addison-Wesley Publishing Company*, New York.

[19] Zeidler, A., Zellner, R., 1992, Software-Ergonomie, Techniken der Dialoggestaltung,*R. Oldenbourg Verlag*, München.