Graphical User Interfaces: Mess them up!

Eric Dubuis*
Institut für Informatik und angewandte Mathematik,
Universität Bern, Neubrückstrasse 10, 3012 Berne, Switzerland.
Email: dubuis@tech.ascom.ch

Abstract
Today's graphical user interfaces simulate a perfect world. Windows have always the same look and buttons have identical shapes and colors and are aligned to regular grids. The only distinction between graphical elements is their label. The current visual design of user interfaces ignores the capabilities of modern displays for an effective visual coding and the resulting uniformness makes it very difficult for the user to recognize graphical elements at first sight.
This paper presents some ideas and propositions for the visual design and the 'look & feel'-concept of next generation user interfaces. New concepts for the layout and the visual coding of graphical elements are proposed. The suggested new layout and coding rules should help the user to recognize and remember graphical elements easily and to distinguish them from other elements of the same type.

Keywords
Graphical interaction, graphical user interfaces, human-computer interaction, visual design, visual coding, look & feel.

So, so you think you can tell Heaven from Hell,
blue skies from pain.
Can you tell a green field from a cold steel rail?
A smile from a veil?
Do you think you can tell?

D. Gilmour, R. Waters

1 Introduction

Graphical user interfaces (GUIs) or windowing systems are widespread and common for today's computers. In the beginning of the 1980s, the evolution of user-friendly interfaces started with simple windowing and menu concepts on text based displays. The main idea of these simple user interfaces was – and still is today – to offer the user a set of options and not forcing him to remember and type in each command.

Several different GUIs are in use today and all of them emerge from the same concepts which have been implemented on the Xerox Star and the Apple Lisa first. Both machines were based on pioneering research performed at the Xerox Palo Alto Research Center in the 1970s. Basically, today's GUIs – which are essentially windowing systems – still look and work the same way as the early prototypes. They consist of windows, menus, controls and dialog boxes.

*Now at: Ascom Tech Ltd., Morgenstrasse 129, 3018 Berne, Switzerland

The look of these components is nearly the same in all common GUIs: Windows are rectangular, have a title bar, close boxes and sliders, menus are arranged in a menu bar and located at the top-border of an application or document and commands are small rectangular buttons with a label. The design of conform applications are described in style guides, i.e. [1], [10], [13].

Although the basic concepts have not changed during the last 20 years, detail improvements have been made. A lot of work has been put into the look of GUIs and into the natural appearance of its graphical components.

This paper concentrates on the visual design – the graphical appearance – of user interfaces. Improvements for the layout and the design of menus and controls are proposed. The functionality and the fundamental concepts like Direct Manipulation [12] or Information Retrieval [11], [9] are not addressed.

2 The Purpose of Graphical User Interfaces

In order to determine the appearance of graphical user interfaces, their purpose has to be defined as clearly as possible. The main idea is clearly the simplification of the human-computer interaction. GUIs should make it easier to use a computer or an operating system. GUIs were designed to help the user by presenting him the familiar view of his desktop on the screen. Whatever function a user wants to perform, he should be able to find it instantly. This implies, that the functionality of a GUI has to be well designed and intuitively understandable.

In order to reach this goal, the graphical elements of GUIs, e.g. windows, menu bars, menus, dialogs and buttons, have to be designed in a manner that the main purpose of the user interface – the intuitive and simple usability – can be fulfilled. Three goals with the design of graphical elements have been stated to be of great importance [2]:

- Recognition: How quickly can the meaning of a graphical element be recognized?
- Remembering: The actual meaning of an element.
- Discrimination: Distinguishing elements from each other.

The term which describes the function of GUIs best is 'look & feel'. All graphical user interfaces have a certain – nice, clean and aesthetic – look and users should intuitively feel how to operate them.

The following sections deal with the 'look' as well as with the 'feel' of GUIs and the importance of both characteristics with respect to the main purpose of GUIs: the ease of usability. Several possibilities are shown to give a user interface more grip and help the user to recognise, remember and discriminate the various graphical elements.

3 Look...

When looking at GUIs from different stages of developement, the most obvious change is their look. Graphical elements have been switched from flat 2-D widgets to more realistic looking so called 3-D window-frames, roll bars, buttons and icons with more details. The appearance of GUIs is now less abstract and more natural than with previous user interface. This makes it easier for the user to identify the elements and to assign the correct meaning.

Only details in the arrangement and general look of user interfaces have changed during the last few years. The way how to work with a user interface and how the user interface interacts with a user is still the same as with the first character based GUIs. Windows are rectangular, have a menu bar on the top and sliders to the right and at the bottom edge.

The main idea behind the look – not the function – of graphical interfaces is still the wish for a well organized and clean looking desktop without piles of documents and folders lying around in a mess. GUIs are designed to represent the perfect world, or at least the perfect desktop.

This strategy has severe drawbacks, as it runs against the goals for the design of GUIs: Recognition, remembering and discrimination. Graphical element is today's GUIs tent to look quite uniform and as they are usually arranged with respect to regular grids, they don't have much grip and no own characteristics.

4 ...& Feel

The look of a GUI is just one part. Its 'feel' – the grip of its graphical elements – is the other important issue. And that's where the problems with today's GUIs are situated. All elements of one kind look alike. Only very few characteristics are used to make these elements distinguishable.

With today's visual design, the 'look' of GUIs is much more important than the 'feel'. Due to the nice design and the clearly arranged elements, inexperienced users can easily learn to use the system but will never be able to use it as fast as a purely character based system. The usual approach is the definition and disposition of so called shortcuts which serve to invoke commands through the keyboard. An other approach is the enhancement of the GUI's 'feeling', which is interesting in situations where many options can be invoked so that the use of shurtcuts becomes impractical or simply not possible for every option.

The key for a better 'feeling' of graphical user interfaces is to design graphical elements which fulfill the 3 goals of recognition, remembering and discrimination and use all graphical capabilities of modern computer displays. The following subchapters describe two simple ways leading to GUI-components which have more grip and more characteristics than the components used at the time being.

4.1 Layout Principles

Three rules which are followed in the layout of all user interfaces are balance, gridding and proportion [5]. These rules have been introduced in order to generate user interfaces which do not distract the eye and give the interface a nice and well balanced look.

The application of these rules have the following meaning and consequences:

- Balance: The center of a widget, e.g. a window, is nicely framed and the eye is drawn to this area.

- Gridding: The sides of widgets of the the same – or at least similar – size are aligned on a grid. This concept serves on giving the interface a neatness and aesthetic appeal [4].

- Proportion: The size of rectangular areas that are laid out on a grid is treated by this rule. Certain ratios of the lengths of rectangle sides are more aesthetic than others [6], [7].

The compliance with these rules leads to nice and aesthetic user interfaces which please the eye. But a lot of possibilities to simplify work are lost. This framework rules against the human possibility to distinguish and remember non-aligned, non-balanced items. If all windows are well balanced, an important aspect of their characteristic is missing. It is much simpler for human beings to recognize items which are not in perfect order.

By aligning everything and omitting natural irregularities on a desktop, an important human capability gets lost. In fact, the gridding rule leads to an ignorance of important capabilities of high-resolution displays and icons can thus only be distinguished by their label[1].

A typical example for the conservative use of graphical capacities of today's displays are menu systems with menu bars and the menus themselves. All items, in either the menu bar or the menu itself, are listed in a most proportionate way. The same font is used for each item and the spacing between them is absolutely equal. The way of recognizing, remembering and discriminating menu entries is to actually read them.

There are many possibilities to ameliorate this state. The most important one is the layout. By giving each item a unique location, the remembering and the discrimination can be simplified a lot. By allowing items to have an orientation different than being aligned to the screens coordinate system, the remembering and the discrimination can be further enhanced. Of course, these parameters – location and orientation – should be used moderately. It makes no sense to strew them without any order and forcing users to search.

Instead of placing all menus in a menu bar on top of a window – or on top of the screen – they may be located separately at window corners or borders. But even the traditional menu bar can be ameliorated by breaking its alignment. Figure 1 shows these two possibilities which give each menu an identity of its own and makes them better distinguishable from the other ones. The user has thus more information than just the menu's lable.

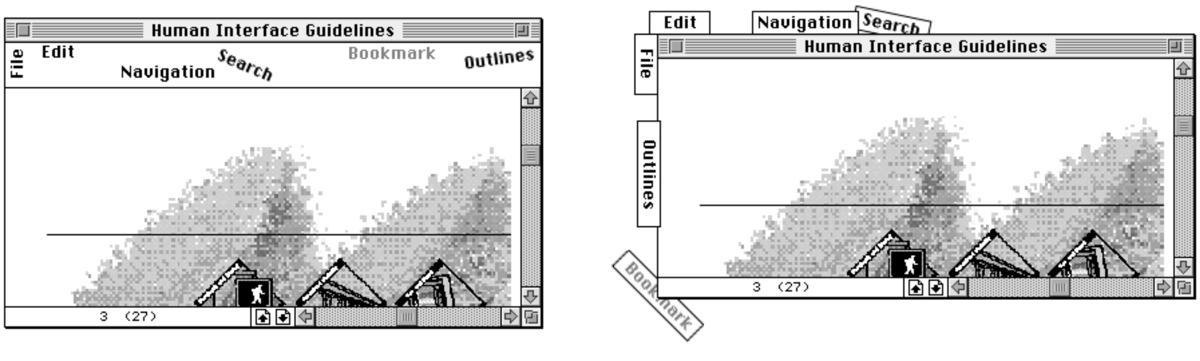


Figure 1: Two examples, how menus can be placed in order to enhance discrimination.

The same principle may be applied to the menu items themselves which are usually just listed up and structured in only one way: groups of items belonging to each other are separated through a vertical line or an empty menu item. Again, the user has actualy to read the menu labels in order to find the appropriate one as the menu structure provides no help in distinguishing and remembering them.

[1]In the context of this paper, a label is everything that does not directly belong to the element itself, i.e. text or a picture.

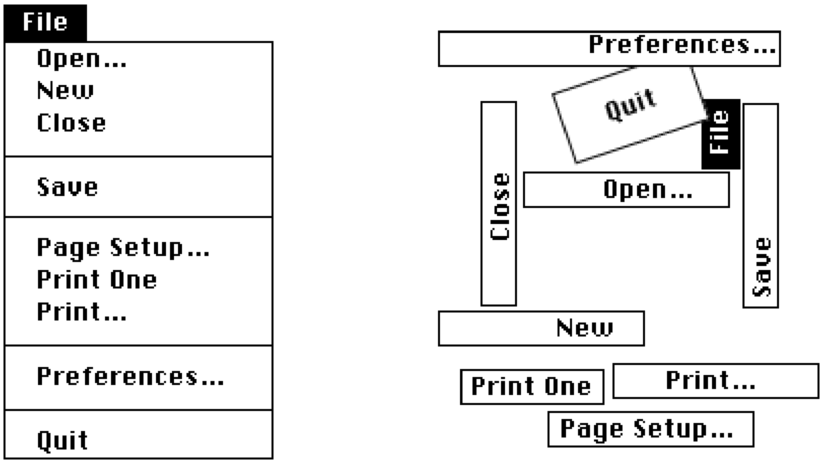


Figure 2: Breaking the regular alignment of menu items.

If a menu is not just regarded as a list of items, but as a set with a structure, items may be arranged in a manner which enhances remembrance and distinguishing. A menu item layout as shown in Figure 2 where each menu item has its own position – and in special cases even an orientation of its own – may look unusual or even ugly, but surely has a lot more characteristics and can be remembered more easily.

A second issue apart from the location is the integration of menu items and the related buttons. The usual approach is to separate the buttons from the menus and to place and arrange them in tool bars which are located near a window border or in freely placeable tool boxes. In either case, the relation between the menu items and the buttons is only understandable if the meaning of the buttons is clear. Unfortunately this is seldom the case and therefore buttons have to be labeled – which makes them useless in a certain way. If buttons are arranged and placed near a menu and if the menu items of an opened menu lie close to the respective button and serve directly as label, the meaning of a button becomes much clearer. Figure 3 shows this integration of buttons and menus.

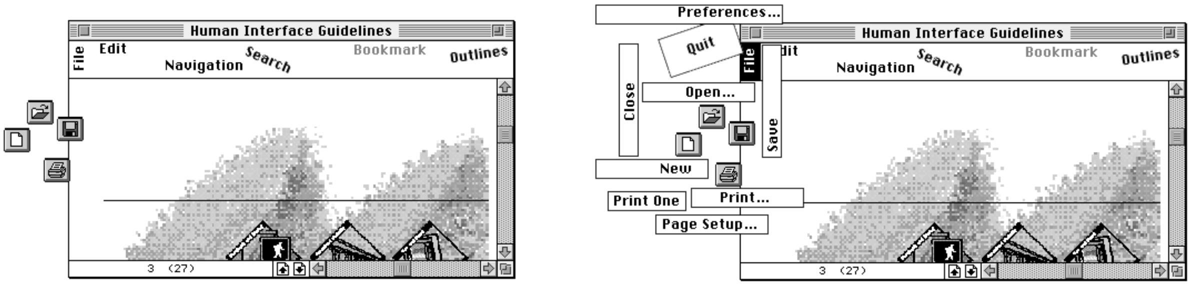
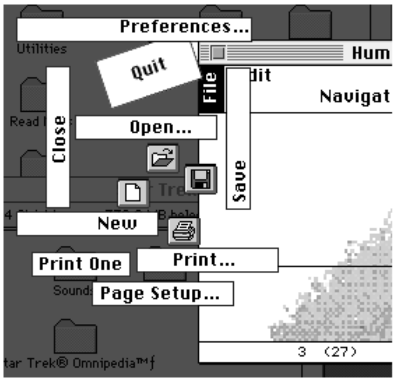


Figure 3: Buttons and menu items can easily be integrated.

A disadvantage of the proposed freely placed menu items and buttons is the lack of a comprising frame or window to emphasize them. This problem can be solved easily by either highlighting the buttons or by framing them (see Figure 4).

4.2 Visual Coding – the Grip of Graphical Elements

The second important possibility to help reaching the three design goals recognition, remembering and discrimination is through visual coding. In the context of interface



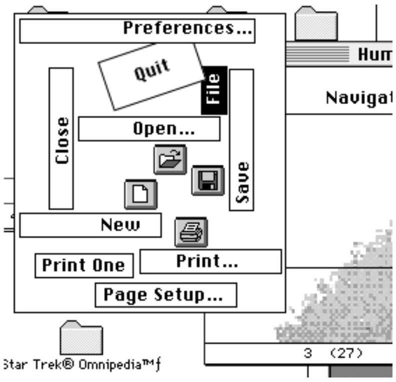


Figure 4: Two possibilities to emphasize menu items and the related buttons.

design, coding means the creation of visual distinctions among several different user interface elements. The most important coding techniques are: color, shape, structure, texture, intensity and font.

The usual questions regarding visual coding are the number of different categories a particular technique can encode and how many code values in different coding techniques can be used without producing too many missinterpretations. These questions have been investigated with experiments, i.e. in [3]. The common goal is to use as many different types of graphical elements at the same time as possible. Even the use of legends is suggested for clarification [5] which, of course, leads once again to more abstract information on the screen.

The use of a large number of different types of widgets runs usually against the wish to have a clear and simple user interface which helps to enhance recognition, remembering and discrimination. In a typical application, e.g. word processing or spreadsheets, the number of different types of graphical elements is usually no crucial point. It is much more likely to have to distinguish between elements of the same type. As today's rules tend to code them as similar as possible, their only difference usually is just a label.

A typical situation where quick decisions between 2 and maybe 3 to 4 options have to be made are dialog boxes. They contain at least a confirmation- (OK) and a negation-button (Cancel). As these buttons are of the same type, they have the same coding and the user gets no visual help in distinguishing them. This means that he has to look at the buttons carefully before making a decision. This is correct according to today's designing rules as the main isea is to make them distinguishable from all other graphical elements on the screen. Anyhow, as only the few controlls in the dialog box are of interest and all investigations about the number of simultaneously usable element types are useless in such a simple case of a dialog box. Only the few elements in the dialog have to be distinguishable.

It makes more sense to give each button an own and very specific characteristic and even use redundant codes extensively in order to give the buttons more grip. The more information the user gets at first sight, the better he can remember a certain button, the better he can distinguish it from other buttons and the faster he can make an error-free decision, which button to select.

An effective way to give a button its own grip, is to make each button look as different as possible from the other ones. This can be achieved by using as much redundancy in coding the buttons as possible. In the case of having only 4 buttons, all coding techniques may be used simultaneously to give each button its own grip (see Figure 5).

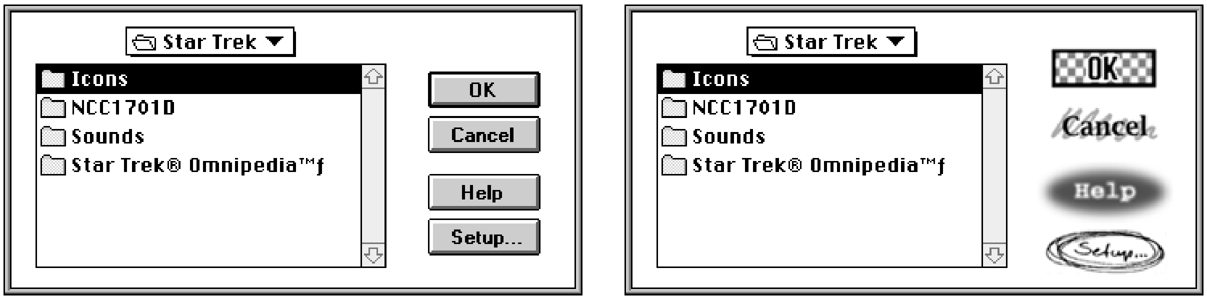


Figure 5: A dialog box with traditional controls and with the new designed controls.

Of course this visual coding technique may be combined with the layout principle proposed in section 4.1. In this case it is important to choose always a similar location and orientation for the same control, i.e. to place the Cancel-button near the right border of each dialog. Figure 6 shows an example of a dialog designed this way.

5 Summary and Conclusion

Basically today's user interfaces are still the same as the first prototypes of 20 years ago. The capabilities of modern displays have not been used to improve their functionality. Still, the computer and its rational and well organized behaviour drives the design of GUIs and lead to aesthetically looking desktops, which are always in perfect order.

The wish for nice and clean desktops is understandable but human beings are not constructed for a perfect world. We need certain irregularities and fine details to perform at our best. This paper has presented two ways to improve the recognition, the remembering and the discrimination of graphical elements in user interfaces.

The proposed methods, to reorganize the layout and to use redundant information in visual coding is only a first step in messing GUIs up. It is absolutely imaginable to take further steps in this direction and e.g. to use non-rectangular windows which allow to visualize program state or performable actions.

6 Acknowledgments

I would like to thank my colleague Christoph Streit for all the helpful discussions and Andrea Dubuis and Diethelm Utzinger for editing this text and providing a lot of useful

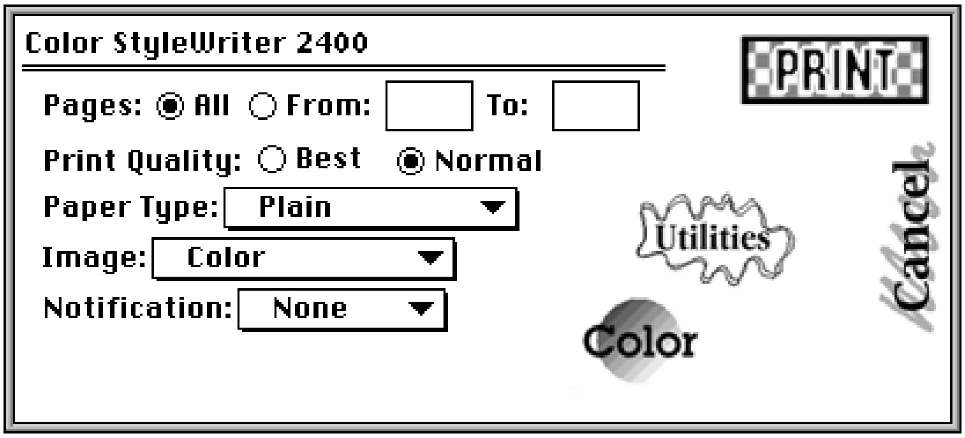


Figure 6: Combining the new layout principle and the proposed visual coding technique.

remarks. Thanks also to all the nameless test persons.

References

[1] Apple Computer: *Human Interface Guidelines: The Apple Desktop Interface*, Addison-Wesley, 1989.

[2] Bewley, W., Roberts, T., Schroit, D., Verplank, W.: *Human Factors Testing in the Design of Xerox's 8010 'Star' Office Workstation*, Proceedings CHI 83 Human Factors in Computing Systems, 1983.

[3] Christ, R.: *Review and Analysis of Color Coding for Visual Display*, Human Factors 17(6), 1975.

[4] Feiner, S.: *A Grid-Based Approach to Automating Display Layout*, Proceedings of Graphics Interface 88, Edmonton, Canada 1988.

[5] Foley, J.D., van Dam, A., Feiner, S., Hughes, J.S.: *Computer Graphics: Principles and Practice*, 2nd Ed., Addison-Wesley, 1990.

[6] Marcus, A.: *Computer-Assisted Chart Making from the Graphic Designers Perspective*, SIGGRAPH 80, 1980.

[7] Marcus, A.: *Corporate Identity for Iconic Interface Design: The Graphic Design Perspective*, IEEE Computer Graphics and Applications, Vol. 4, Nr. 12, 1984.

[8] Marcus, A.: *Graphic Design for Electronic Documents and User Interfaces*, Addison-Wesley, 1992.

[9] Nielsen, J.: *Hypertext and Hypermedia*, Academic Press, 1990.

[10] Open Software Foundation: *OSF/Motif Window Manager and OSF (Motif Toolkit Preliminary Functional Specification)*, OSF, 1990.

[11] Salton, G.: *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, 1989.

[12] Shneiderman, B.: *Designing the User Interface, Strategies for Effective Human-Computer Interaction*, 2nd Ed., Addison-Wesley, 1993.

[13] Sun Microsystems : *OPEN LOOK Graphical User Interface Functional Specification*, 1989.