FUTURE VISION BIE

Future Vision

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

A Small Contribution Would Support Us.

Dear Viewer,

Future Vision BIE is a free service and so that any Student/Research Personal **Can Access Free of Cost**.

If you would like to say **thanks**, you can make a **small contribution** to the author of this site.

Contribute whatever you feel this is worth to you. This gives **us support** & to bring **Latest Study Material** to you. After the Contribution Fill out this Form (https://forms.gle/tw3T3bUVpLXL8omX7). To Receive a **Paid E-Course for Free**, from our End within 7 Working Days.

Regards

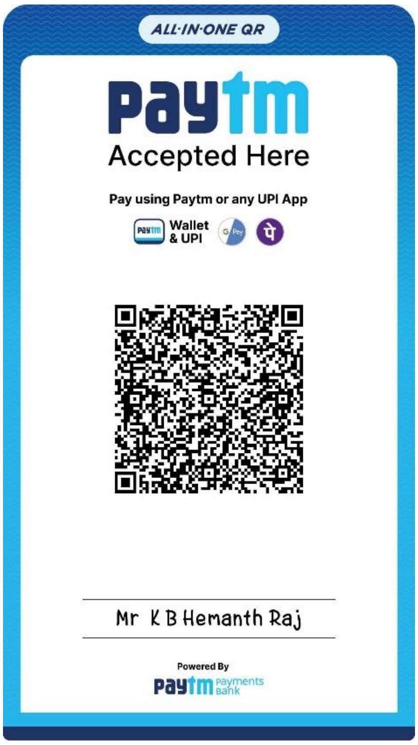
- K B Hemanth Raj (Admin)

Contribution Methods

UPI ID

1. futurevisionbie@oksbi
2. futurevisionbie@paytm

Scan & Pay



Account Transfer

Account Holder's Name: K B Hemanth Raj

Account Number: 39979402438

IFSC Code: SBIN0003982

MICR Code: 560002017

More Info: https://hemanthrajhemu.github.io/Contribution/

Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering & MORE...

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: https://bit.ly/FVBIESHARE

The Essential Guide to User Interface Design
An Introduction to GUI Design Principles and Techniques

Third Edition

Wilbert O. Galitz

BICENTENNIAL 1807 WILEY 2007 BICENTENNIAL

Wiley Publishing, Inc.

https://hemanthrajhemu.github.io

	The Web User Interface	28
	The Popularity of the Web	29
	Characteristics of a Web Interface	29
	The Merging of Graphical Business Systems and the Web	39
	Characteristics of an Intranet versus the Internet	39
	Extranets	40
	Web Page versus Application Design	40
	Principles of User Interface Design	44
	Principles for the Xerox STAR	44
	General Principles	45
	Part 1 Exercise	58
	What's Next?	58
Part 2	**The User Interface Design Process**	**59**
	Obstacles and Pitfalls in the Development Path	59
	Designing for People: The Seven Commandments	60
	Usability	64
	Usability Assessment in the Design Process	65
	Common Usability Problems	65
	Some Practical Measures of Usability	68
	Some Objective Measures of Usability	69
Step 1	**Know Your User or Client**	**71**
	Understanding How People Interact with Computers	71
	The Human Action Cycle	72
	Why People Have Trouble with Computers	73
	Responses to Poor Design	74
	People and Their Tasks	76
	Important Human Characteristics in Design	76
	Perception	76
	Memory	78
	Sensory Storage	79
	Visual Acuity	80
	Foveal and Peripheral Vision	81
	Information Processing	81
	Mental Models	82
	Movement Control	83
	Learning	83
	Skill	84
	Performance Load	84
	Individual Differences	85
	Human Considerations in the Design of Business Systems	87
	The User's Knowledge and Experience	87
	The User's Tasks and Needs	92
	The User's Psychological Characteristics	95
	The User's Physical Characteristics	96
	Human Interaction Speeds	100

https://hemanthrajhemu.github.io

	Performance versus Preference	101
	Methods for Gaining an Understanding of Users	102
	Step 1 Exercise	102
Step 2	**Understand the Business Function**	**103**
	Business Definition and Requirements Analysis	104
	Information Collection Techniques	104
	Defining the Domain	112
	Considering the Environment	112
	Possible Problems in Requirements Collection	113
	Determining Basic Business Functions	113
	Understanding the User's Work	114
	Developing Conceptual Models	115
	The User's New Mental Model	120
	Design Standards or Style Guides	120
	Value of Standards and Guidelines	121
	Customized Style Guides	124
	Design Support and Implementation	125
	System Training and Documentation Needs	125
	Training	126
	Documentation	126
	Step 2 Exercise	126
Step 3	**Understand the Principles of Good Interface and Screen Design**	**127**
	Human Considerations in Interface and Screen Design	128
	How to Discourage the User	128
	What Users Want	130
	What Users Do	130
	Interface Design Goals	131
	The Test for a Good Design	132
	Screen and Web Page Meaning and Purpose	132
	Organizing Elements Clearly and Meaningfully	133
	Consistency	133
	Starting Point	135
	Ordering of Data and Content	136
	Navigation and Flow	139
	Visually Pleasing Composition	141
	Distinctiveness	161
	Focus and Emphasis	162
	Conveying Depth of Levels or a Three-Dimensional Appearance	165
	Presenting Information Simply and Meaningfully	168
	Application and Page Size	178
	Application Screen Elements	184
	Organization and Structure Guidelines	220
	The Web — Web sites and Web Pages	230

https://hemanthrajhemu.github.io

	Intranet Design Guidelines	258
	Extranet Design Guidelines	259
	Small Screens	259
	Weblogs	260
	Statistical Graphics	261
	Types of Statistical Graphics	273
	Flow Charts	283
	Technological Considerations in Interface Design	284
	Graphical Systems	284
	Web Systems	287
	The User Technology Profile Circa 2006	292
	Examples of Screens	293
	Example 1	293
	Example 2	297
	Example 3	300
	Example 4	301
	Example 5	302
	Example 6	303
	Example 7	305
	Step 3 Exercise	306
Step 4	**Develop System Menus and Navigation Schemes**	**307**
	Structures of Menus	308
	Single Menus	308
	Sequential Linear Menus	309
	Simultaneous Menus	309
	Hierarchical or Sequential Menus	310
	Connected Menus	311
	Event-Trapping Menus	313
	Functions of Menus	313
	Navigation to a New Menu	314
	Execute an Action or Procedure	314
	Displaying Information	314
	Data or Parameter Input	314
	Content of Menus	314
	Menu Context	315
	Menu Title	315
	Choice Descriptions	315
	Completion Instructions	315
	Formatting of Menus	315
	Consistency	316
	Display	316
	Presentation	316
	Organization	317
	Complexity	320
	Item Arrangement	321
	Ordering	321

https://hemanthrajhemu.github.io

	Groupings	323
	Selection Support Menus	325
	Phrasing the Menu	328
	Menu Titles	329
	Menu Choice Descriptions	330
	Menu Instructions	332
	Intent Indicators	332
	Keyboard Shortcuts	333
	Selecting Menu Choices	337
	Initial Cursor Positioning	337
	Choice Selection	338
	Defaults	339
	Unavailable Choices	340
	Mark Toggles or Settings	340
	Toggled Menu Items	341
	Web Site Navigation	342
	Web Site Navigation Problems	343
	Web Site Navigation Goals	344
	Web Site Navigation Design	345
	Maintaining a Sense of Place	367
	Kinds of Graphical Menus	369
	Menu Bar	369
	Pull-Down Menu	371
	Cascading Menus	375
	Pop-Up Menus	377
	Tear-Off Menus	379
	Iconic Menus	380
	Pie Menus	380
	Graphical Menu Examples	382
	Example 1	382
Step 5	**Select the Proper Kinds of Windows**	**385**
	Window Characteristics	385
	The Attraction of Windows	386
	Constraints in Window System Design	388
	Components of a Window	390
	Frame	390
	Title Bar	391
	Title Bar Icon	391
	Window Sizing Buttons	392
	What's This? Button	393
	Menu Bar	393
	Status Bar	394
	Scroll Bars	394
	Split Box	394
	Toolbar	394
	Command Area	395

https://hemanthrajhemu.github.io

PART

Two

The User Interface Design Process

Part 2 presents an extensive series of guidelines for the interface design process. It is organized in the order of the development steps typically followed in creating a graphical system's or Web site's screens and pages. In total, 14 steps are presented, beginning with "Know Your User or Client" and ending with a discussion of testing. Other topics addressed include considerations in screen design, navigation, screen-based controls, writing messages and text, color, and graphics. This organization scheme enables all the interface design activities to be addressed easily, clearly, and sequentially.

Let's first look at several critical general aspects of the design process. "Obstacles and Pitfalls in the Development Path" points out the realities of designing for people, and some reasons why design may not live up to expectations. "Designing for People: The Seven Commandments" lists the guidelines that are the cornerstones of the entire design process. Then, the concept of usability, the primary objective on any development effort, is defined and discussed.

Obstacles and Pitfalls in the Development Path

Developing a computer system is never easy. The path is littered with obstacles and traps, many of them human in nature. Gould (1988) has made these general observations about design:

- Nobody ever gets it right the first time.
- Development is chock-full of surprises.

https://hemanthrajhemu.github.io

- Good design requires living in a sea of changes.
- Making contracts to ignore change will never eliminate the need for change.
- Even if you have made the best system humanly possible, people will still make mistakes when using it.
- Designers need good tools.
- You must have behavioral design goals like performance design goals.

The first five conditions listed will occur naturally because people are people, both as users and as developers. These kinds of behavior must be understood and accepted in design. User mistakes, while they will always occur, can be reduced. Guidelines in the various design steps address this problem. Behavioral design goals are reviewed in Step 2, "Understand the Business Function."

Pitfalls in the design process exist because of a flawed design process, including a failure to address critical design issues, an improper focus of attention, or development team organization failures. Common pitfalls are:

- No early analysis and understanding of the user's needs and expectations.
- A focus on using design features or components that are "neat" or "glitzy."
- Little or no creation of design element prototypes.
- No usability testing.
- No common design team vision of user interface design goals.
- Poor communication between members of the development team.

"Know Your User or Client" is addressed in Step 1. Prototypes and testing are addressed in Step 14, "Test, Test, and Retest."

Designing for People: The Seven Commandments

The complexity of a graphical or Web interface will always magnify any problems that do occur. While obstacles to design will always exist, pitfalls can be eliminated if the following design commandments remain foremost in the development process.

1. **Provide a multidisciplinary design team**. Provide a balanced design team, including specialists in:
   - Development, including system analysis and software design.
   - Interface Design.
   - Visual design.
   - Usability assessment.
   - Documentation.
   - Training.

   Effective design and development requires the application of very diverse talents. No one person possesses all the skills to perform all the necessary tasks; the best that can be hoped for is that one person may possess a couple of skills. A balanced design team with very different talents must be established. Needed are

https://hemanthrajhemu.github.io

specialists in development to define requirements and write the software, human factors specialists to define behavioral requirements and apply behavioral considerations, and people with good visual design skills. Also needed are people skilled in testing and usability assessment, documentation specialists, and training specialists.

All designers, however, should be strongly user-oriented. A study (Bailey, 1993) revealed that user-oriented designers are superior to computer-oriented designers when making user interface design decisions. User-oriented designers should be responsible for making the majority of user interface design decisions

Also, select team members who can effectively work and communicate with one another. To optimize communication, locate the team members in close proximity to one another.

2. **Solicit early and ongoing user involvement**. Involving the users in requirements determination and/or testing from the beginning provides a direct conduit to the knowledge they possess about jobs, tasks, system goals, and needs. Different types of users may exist:

- **End users**. Sometimes simply called users, these are the people who actually use the system to perform tasks and jobs. One caution, however: user involvement of this kind should be based on job or task knowledge, not status or position. The boss seldom knows what is really happening in the office. Throughout the remainder of this text the term “users” will be used to designate end users.
- **Customers**. These are the people within the using organization who pay for and usually specify the overall objectives and goals of the system.
- **Other interested parties**. These are people within the user organization who also have an interest in the development of the system.

Involvement of these different kinds of users also enables the developer to confront people’s resistance to change, a common human trait. People dislike change for a variety of reasons, among them fear of the unknown and lack of identification with the system. Involvement in design removes the unknown and gives the user a stake in the system or identification with it.

It has long been a belief among designers that involving users in the design phase of a system is beneficial in terms of system quality, efficiency, and effectiveness. Kujala (2003) performed a research literature review to clarify the relationship between user involvement during design and development and final system success. Reviewing more than three dozen studies, she reported both positive and negative findings. Bailey (2005a), in reviewing Kujala’s findings, reached the following conclusions regarding user involvement:

- A more accurate set of requirements will be obtained.
- In some situations there may be improved user acceptance of the system.
- There is little evidence that systems are either more effective or efficient when users are closely involved in making design decisions.
- During testing, users can be effectively used as participants. Emphasis should be on obtaining quantitative data.

https://hemanthrajhemu.github.io

It is helpful, then, for users to be involved at every stage in the interface design, development, and implementation cycle:

- **Early in the design process when requirements are being determined**. Users can help by providing design requirements and specifications, testing early design prototypes, and by allowing themselves to be observed performing their current tasks. Users can also provide feedback concerning their current system and the prototypes being tested.
- **Throughout prototyping to test designs and options**. Feedback and suggestions can made for each prototype tested in the development process.
- **During training**. Opinions can be gathered and any additional problems described.
- **After system delivery**. Opinions can be gathered and feedback concerning any additional problems encountered during actual system use can be provided.

3. **Gain a complete understanding of users and their tasks**. All users, including customers and other interested parties, today expect a level of design sophistication from all user interfaces, including Web sites. The product, system or Web site must be geared to people's needs and the system's goal, not those of the developers. A wide gap in technical abilities, objectives, and attitudes often exists between users and developers. A failure by developers to understand the differences will doom a product or system to failure.

   Usability goals in the form of measurable objectives must also be established. Set performance goals such as success rates and the time it takes to complete tasks. Set preference goals that address satisfaction and acceptance by users. Design success cannot be determined without quantitative values to compare system performance against.

4. **Create the appropriate design**. The total user experience must be created, including an appropriate allocation of function between the user and the system. Consider as many user interface issues as possible during the design process. A design methodology that has been found to be successful is called *parallel design.* Using this concept, proposed by Ovaska and Raiha (1995), multiple developers independently evaluate design requirements and issues and propose design solutions. Then, to find the best ideas, individual solutions are presented to, and discussed among, all developers. Two studies (Macbeth et al., 2000; McGrew, 2001) have found this process works exceptionally well. More design ideas are presented and considered, and developers responded to good ideas no matter who had proposed them. As Bailey (2002) suggests, the objective is to "saturate the design space." Interface designers should consider as many alternative designs as possible before selecting the best among them to begin the iterative design process.

   Begin utilizing design standards and guidelines at the start of the design process. All interface design decisions must be made as design proceeds, not after design is complete. This helps ensure that the best possible design decisions are made and that design consistency is achieved. This also avoids problems later on in the development process.

https://hemanthrajhemu.github.io

5. **Perform rapid prototyping and testing**. Prototyping and testing the product will quickly identify problems and allow solutions to be developed. The design process is complex and human behavior is still not well understood. While the design guidelines that follow go a long way toward achieving ease of use, all problems cannot possibly be predicted. Prototyping and testing must be continually performed during all stages of development to uncover all potential defects.

   If thorough testing is not performed before product release, the testing will occur in the user's office. Encountering a series of problems early in system use will create a negative first impression in the customer's mind, and this may harden quickly, creating attitudes that may be difficult to change. It is also much harder and more costly to fix a product after its release. In many instances, people may adapt to, or become dependent upon, a design, even if it is inefficient. This also makes future modifications much more difficult.

6. **Modify and iterate the design as much as necessary**. Design will be an *iterative* process. Design prototypes will be developed and tested, and changes will be made on the basis of the test results. The process will be repeated, fine-tuning occurring, until all usability goals are achieved. The substantial value of iterative design has been confirmed by several studies (Tan et al., 2001; Bailey and Wolfson, 2005; LeDoux et al., 2005). Each of these studies found that system modifications based upon the results of one test yielded performance improvements on a follow-up test. For example:

   - A 28% faster average task completion time (Tan et al., 2001).
   - A 37% reduction in usability problems (Tan et al., 2001).
   - Nine of ten task scenarios took less time (Bailey and Wolfson, 2005).
   - User satisfaction score increased from 63 to 73 (Bailey and Wolfson, 2005).
   - The average time to complete task scenarios was reduced from 68 to 51 seconds (25% improvement) (LeDoux et al., 2005).
   - The overall user satisfaction score improved from 49 to 82 (67% improvement) (LeDoux et al., 2005).

   Test, modify, and retest has been proven to work well. This process is repeated until all usability goals are achieved. Then the iterative process ends.

7. **Integrate the design of all the system components**. The software, the documentation, the help function, and training needs are all important elements of a graphical system or Web site and all should be developed concurrently. A system is being constructed, not simply software. Concurrent development of all pieces will point out possible problems much earlier in the design process, allowing them to be more effectively addressed. Time will also exist for design trade-offs to be thought out more carefully.

https://hemanthrajhemu.github.io

Usability

Much of the development process will focus on the concept of system *usability*. Usability is a quality attribute that assesses how easy a user interface is to use. The term *usability* also refers to methods for improving ease-of-use throughout the entire design process.

Bennett (1979) was the first to use the term usability to describe the effectiveness of human performance. In the following years a more formal definition was proposed by Shackel (1981) and modified by Bennett (1984). Shackel (1991) simply defined usability as "the capability to be used by humans easily and effectively, where,

easily = to a specified level of subjective assessment,

effectively = to a specified level of human performance."

In recent years more specific descriptions have been presented. Nielsen (2003) suggests usability possess these five quality components:

- **Learnability**: How easy is it for users to accomplish basic tasks the first time they encounter the design?
- **Efficiency**: Once users have learned the design, how quickly can they perform tasks?
- **Memorability**: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- **Errors**: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**: How pleasant is it to use the design?

The following dimensions of usability have been described by Quesenbery (2003).

- **Effective.** The completeness and accuracy with which users achieve their goals.
- **Efficient.** The speed (with accuracy) with which users can complete their tasks.
- **Engaging.** The degree to which the tone and style of the interface makes the product pleasing or satisfying to use.
- **Error tolerant.** How well the design prevents errors and helps with recovery from those that do occur.
- **Easy to learn.** How well the product supports both initial orientation and deepening understanding of its capabilities.

Usability is one of an interface's most important qualities. For systems or products whose use is discretionary, such as Web sites, a difficult to use interface can cause people to stop using it. For business applications, whose use is usually mandatory, the result is lowered worker productivity. Usability, however, cannot be looked at independently of another system quality, *utility*. Utility refers to a system or product's functionality. Does it do what people want it to do? An entity may have a high level of usability but does not accomplish anything of value for its user. Conversely, an entity may be capable of performing many valuable functions for its user but, because it is

https://hemanthrajhemu.github.io

not easy to use, the functions cannot be accomplished. Usability and utility are equally important in design.

MYTH **Usability is nothing but common sense.**

Usability also has a relationship with flexibility in design. In general, as the flexibility of a design increases, its usability decreases. Flexible designs can perform more functions than specialized designs, but they perform them less efficiently. The flexibility-usability trade-off exists because accommodating flexibility entails satisfying a greater number of design requirements. This leads to more design compromises and more complexity in the design. Flexibility generally pays most dividends when users cannot clearly anticipate their future needs. Then, flexible designs that enable people to address future contingencies are usually needed. That flexibility will usually result in a reduction in usability, however, and should always be considered.

Usability Assessment in the Design Process

Usability assessment should begin in the early stages of the product development cycle and should be continually applied throughout the process. The assessment should include the user's entire experience, and all the product's important components. Usability assessment methods are discussed in more detail in Step 14, "Test, Test, and Retest."

Common Usability Problems

Mandel (1994) lists the 10 most common usability problems in graphical systems as reported by IBM usability specialists. They are:

1. Ambiguous menus and icons.
2. Languages that permit only single-direction movement through a system.
3. Input and direct manipulation limits.
4. Highlighting and selection limitations.
5. Unclear step sequences.
6. More steps to manage the interface than to perform tasks.
7. Complex linkage between and within applications.
8. Inadequate feedback and confirmation.
9. Lack of system anticipation and intelligence.
10. Inadequate error messages, help, tutorials, and documentation.

The Web, with its dynamic capabilities and explosive entrance into our lives, has unleashed what seems like more than its own share of usability problems. Many are similar to those outlined previously. One study (Ceaparu et al., 2004) found users spend almost 40% of their computer time trying to get things to work or work better. Difficult installations, viruses, and connectivity troubleshooting challenge people. The

https://hemanthrajhemu.github.io

systems that slow them down the most are operating systems, e-mail, and Web browsing problems.

Web usability characteristics particularly wasteful of people's time, and often quite irritating, are:

Visual clutter. A lack of "white space," meaningless graphics, and unnecessary and wasteful decoration often turn pages into jungles of visual noise. Meaningful content lies hidden within the unending forest of vines and trees, forcing the user to waste countless minutes searching for what is relevant. Useless displayed elements are actually a form of visual noise.

Impaired information readability. Page readability is diminished by poor developer choices in typefaces, colors, and graphics. Use of innumerable typefaces and kaleidoscopic colors wrestle meaning from the screen. A person's attention is directed toward trying to understand why the differences exist, instead of being focused toward identifying and understanding the page's content. Backgrounds that are brightly colored or contain pictures or patterns greatly diminish the legibility of the overwritten text.

Incomprehensible components. Some design elements give the user no clue as to their function, leaving their purpose not at all obvious. Some icons and graphics, for example, are shrouded in mystery, containing no text to explain what they do. Some buttons don't look at all like command buttons, forcing the user to "minesweep" the screen with a mouse to locate the objects that can be used to do something. Command buttons or areas that give no visual indication that they are clickable often won't be clicked. Language is also often confusing, with the developer's terminology being used, not that of the user.

Annoying distractions. Elements constantly in motion, scrolling marquees or text, blinking text, or looping continually running animations compete with meaningful content for the user's eye's and attention—and destroy a page's readability. Automatically presented music or other sounds interrupt one's concentration, as do nonrequested pop-up widows, which must be removed, wasting more of the user's time. A person's senses are under constant attack, and the benefits afforded by one's peripheral vision are negated.

Confusing navigation. A site's structure often resembles a maze of twisting pages into which the user wanders and is quite soon lost. Poor, little, or no organization exists among pages. The size and depth of many Web sites can eventually lead to a "lost in space" feeling as perceived site structure evaporates as one navigates. Embarking on a side trip can lead to a radical change in context or a path with no signposts or landmarks. Navigation links lead to dead-ends from which there is no return, or boomerang you right back to the spot where you are standing without you being aware of it. Some navigation elements are invisible. (See mystery icons and minesweeping above.) Confusing navigation violates expectations and results in disturbing unexpected behavior.

Inefficient navigation. A person must transverse content-free pages to find what is meaningful. One whole screen is used to point to another. Large graphics waste screen space and add to the page count. The path through the navigation maze is

https://hemanthrajhemu.github.io

often long and tedious. Reams of useless data must be sifted through before a need can be fulfilled. Massive use of short pages with little content often creates the feeling that one is "link drunk."

Inefficient operations. Time is wasted doing many things. Page download times can be excessive. Pages that contain, for example, large graphics and maps, large chunky headings, or many colors, take longer to download than text. Excessive information fragmentation can require navigation of long chains of links to reach relevant material, also accelerating user disorientation.

Excessive or inefficient page scrolling. Long pages requiring scrolling frequently lead to the user's losing context as related information's spatial proximity increases and some information entirely disappears from view and, therefore, from memory. Out of sight is often out of mind. If navigation elements and important content are hidden below the page top, they may be missed entirely. To have to scroll to do something important or complete a task can be very annoying; especially if the scrolling is caused by what the user considers is an irrelevancy or noise.

Information overload. Poorly organized or large amounts of information tax one's memory and can be overwhelming. Heavy mental loads can result from making decisions concerning which links to follow and which to abandon, given the large number of choices available. Or from trying to determine what information is important, and what is not. Or from trying to maintain one's place in a huge forest of information trees. One easily becomes buried in decisions and information. Requiring even minimal amounts of learning to use a Web site adds to the mental load.

Design inconsistency. Design inconsistency has not disappeared with the Web. It has been magnified. The business system user may visit a handful of systems in one day, the Web user may visit dozens, or many more. It is expected that site differences will and must exist because each Web site owner strives for its own identity. For the user's sake, however, some consistency must exist to permit a seamless flow between sites. Consistency is needed in, for example, navigation element location on a page and the look of navigation buttons (raised). The industry is diligently working on this topic and some "common practices" are already in place. The learning principle of rote memorization, however, is still being required within many sites. For example, the industry practice of using different standard link colors for unvisited sites (blue) and visited sites (purple) is often violated. This forces users to remember different color meanings in different places, and this also causes confusion between links and underlined text. Design guidelines for graphical user interfaces have been available for many years. Too often they are ignored (or the designer is unaware of them). Examples of inappropriate uses abound in design. The use of check boxes instead of radio buttons for mutually exclusive options, for example. Or the use of drop-down list boxes instead of combination boxes when the task mostly requires keyboard form fill-in. The Web is a form of the graphical user interface, and GUI guidelines should be followed.

https://hemanthrajhemu.github.io

MAXIM **The usability of a system is improved when similar parts are expressed in similar ways.**

Outdated or undated information. One important value of a Web site is its "currentness." Outdated or undated information destroys a site's credibility in the minds of many users, and therefore its usefulness. A useless site is not very usable.

Stale design caused by emulation of printed documents and past systems. The Web is a new medium with expanded user interaction and information display possibilities. While much of what we have learned in the print world and past information systems interface design can be ported to the Web, all of what we know should not be blindly moved from one to the other. Web sites should be rethought and redesigned using the most appropriate and robust design techniques available.

Some of these usability problems are a result of the Web's "growing pains." For other problems developers themselves can only be blamed, for they too often have created a product to please themselves and "look cool," not to please their users. Symptoms of this approach include overuse of bleeding edge technology, a focus on sparkle, and jumping to implement the latest Internet technique or buzzword. These problems, of course, did not start with the Web. They have existed since designers began building user interfaces.

Some Practical Measures of Usability

Usability, or the lack thereof, can often be sensed by a simple observation of, or talking to, people using an interface. While these measures lack scientific rigor, they do provide an indication that there may be usability problems.

Are people asking a lot of questions or often reaching for a manual? Many questions or frequent glances at manuals are signs that things are not as clear and intuitive as they should be. When in doubt, the first reaction of many people is to ask someone for assistance. When no one is around, then we look in a manual.

Are frequent exasperation responses heard? "Oh damn!" or similar reactions are usually used to express annoyance or frustration. Their frequency, and loudness, may foretell a strong rejection of a product. The absence of exasperation, however, may not represent acceptance. Some people are not as expressive in their language, or are better able to smother their feelings.

Are there many irrelevant actions being performed? Are people doing things the hard way? Are there incidental actions required for, but not directly related to, doing a job? These include excessive mouse clicks or keyboard strokes to accomplish something, or going through many operations to find the right page in a manual or the right window or page in the display.

Are there many things to ignore? Are there many elements on the screen that the user must disregard? Are there many "doesn't pertain to me" items? If so,

https://hemanthrajhemu.github.io

remember, they still consume a portion of a person's visual or information-processing capacities, detracting from the capacities a person could devote to relevant things.

Do a number of people want to use the product? None of us goes out of our way to make our own lives more difficult. (Unfortunately, other people may, however.) We tend to gravitate to things easy to work with or do. If a lot of people want to use it, it probably has a higher usability score. Attitudes may be a very powerful factor in a system's or Web site's acceptance.

Some Objective Measures of Usability

Tyldesley (1988) and Shackel (1991) have both presented possible objective criteria for measuring usability. Tyldesley's criteria are shown in Table II.1 Shackel's criteria are as follows:

How *effective* is the interface? Can the required range of tasks be accomplished:

- At better than some required level of performance (for example, in terms of speed and errors)?
- By some required percentage of the specified target range of users?
- Within some required proportion of the range of usage environments?

How *learnable* is the interface? Can the interface be learned:

- Within some specified time from commissioning and start of user training?
- Based on some specified amount of training and user support?
- Within some specified relearning time each time for intermittent users?

How *flexible* is the interface? Is it flexible enough to:

- Allow some specified percentage variation in tasks and/or environments beyond those first specified?

What are the *attitudes* of the users? Are they:

- Within acceptable levels of human cost in terms of tiredness, discomfort, frustration, and personal effort?
- Such that satisfaction causes continued and enhanced usage of the system?

The selection of the most appropriate measurements will be dependent upon the type of system and/or application being tested. For Web sites some terminology may have to be modified to reflect page elements (e. g. Links, not commands).

Human performance goals in system use, like any other design goal, should be stated in quantitative and measurable ways. Without performance goals it will not be known if they are achieved, or how successful the system really is. Clear and concrete goals also provide objectives for usability testing and ensure that a faulty or unsatisfactory product will not be released.

Values for the various criteria should be specified in absolute terms. An absolute goal might be "Task A must be performed by a first-time user in 12 minutes with no

https://hemanthrajhemu.github.io

errors with 30 minutes of training and without referring to a manual." Goals may also be set in relative terms. For example, "Task B must be performed 50 percent faster than it was using the previous system."

The level of established goals will depend on the capabilities of the user, the capabilities of the system, and the objectives of the system. In addition to providing commitments to a certain level of quality, goals become the foundation for the system test plan.

Table II.1 Possible Usability Measurement Criteria

1. Time to complete a task.
2. Percentage of task completed.
3. Percentage of task completed per unit time (speed metric).
4. Ratio of successes to failures.
5. Time spent on errors.
6. Percentage of number of errors.
7. Percentage of number of competitors that do this better than the current product.
8. Number of commands used.
9. Frequency of help or documentation use.
10. Time spent using help or documentation.
11. Percentage of favorable to unfavorable user commands.
12. Number of repetitions of failed commands.
13. Number of runs of success and of failures.
14. Number of times the interface misleads the user.
15. Number of good and bad features recalled by users.
16. Number of available commands not invoked.
17. Number of regressive behaviors.
18. Number of users preferring your system.
19. Number of times users need to work around a problem.
20. Number of times the user is distracted from a work task.
21. Number of times the user loses control of a system.
22. Number of times the user expresses frustration or satisfaction.

From Tyldesley (1988).

https://hemanthrajhemu.github.io

STEP
1

Know Your User or Client

The journey into the world of interface design and the screen design process must begin with an understanding of the system user, the *most important* part of any computer system. The user needs a system that is built to serve. Understanding people and what they do is a difficult and often undervalued process but very critical because of the gap in knowledge, skills, and attitudes existing between system users and developers that build the systems. To create a truly usable system, the designer must always do the following:

- Understand how people interact with computers.
- Understand the human characteristics important in design.
- Identify the user's level of knowledge and experience.
- Identify the characteristics of the user's needs, tasks, and jobs.
- Identify the user's psychological characteristics.
- Identify the user's physical characteristics.
- Employ recommended methods for gaining understanding of users.

Understanding How People Interact with Computers

To begin, the human action cycle and how people interact with computers will be described. Then some characteristics of computer systems past and present that have

https://hemanthrajhemu.github.io

caused, and are causing, people problems will be discussed. Finally, the effect these problems have will be reviewed.

The Human Action Cycle

The human action cycle is a psychological model that describes how people interact with computer systems. It was presented by Norman (1988) and adapted by Stone et al. (2005), whose model is the basis for this discussion.

People tend to be goal oriented, wanting to achieve a specific objective when taking an action or performing a task. Actions and tasks can be cognitive or physical in nature. The action cycle consists of three stages: goal formation, execution of activities to achieve the goal, and evaluation of the results of the action. The cycle flows as follows:

A goal is formed. A cognitive activity, appropriate objectives, or an objective, are thought out and defined. The goal may be to type and print a letter, or to find a good price for a specific electronic device.

An execution plan is devised and implemented. This phase consists of three stages, the first two being cognitive in nature, the third being physical.

General methods to achieve the desired goals are decided upon. Typing a letter requires using a computer's word processing function. Finding a best price requires using the Internet to do some comparison shopping.

The action sequence is planned. Typing a letter requires opening the word processor, retrieving a blank document, and typing the letter. Best-price determination requires doing an Internet search to locate possible sources of the desired equipment, accessing the Web sites of these sources, and finding the equipment prices.

The actions are performed. The various available computer controls, such as the keyboard and mouse, are used to perform the planned tasks.

The results of the actions taken are evaluated. This is another cognitive phase that also consists of three stages.

The resulting output is perceived and understood. The letters and symbols typed appear on the screen. Prices of equipment are provided on the Web site being viewed.

The outcome is interpreted based upon expectations. Is the letter formatted properly and its content accurate and complete? Are the prices displayed for the proper equipment?

The results are compared to the formulated goals. Has the letter been printed correctly? Is the price a good price? A failure to achieve formulated goals may cause the action cycle's goals to be modified or performed again. Multiple iterations are possible.

Simple action cycles may last only a few seconds, and others may take hours to complete. In the execution phase numerous ways may exist to fulfill one's goal. Interface design should enable the human action cycles for tasks to be performed as quickly and accurately as possible.

https://hemanthrajhemu.github.io

Why People Have Trouble with Computers

Although system design and its behavioral implications have come under intense scrutiny in the last 15 to 20 years, this has not always been the case. Historically, the design of business computer systems has been the responsibility of programmers, systems analysts, and system designers, many of whom possess extensive technical knowledge but little behavioral training. In recent years the blossoming of the Web, with its extensive graphical capabilities, has found graphic artists being added to design teams. Like those who have come before them, most graphic artists also possess extensive technical knowledge in their profession but little training in usability. Design decisions, therefore, have rested mostly on the designer's intuition concerning the user's capabilities and the designer's wealth of specialized knowledge. Consequently, poorly designed interfaces have often gone unrecognized.

The intuition of designers or of anyone else, no matter how good or bad they may be at what they do, is error-prone. It is much too shallow a foundation on which to base design decisions. Specialized knowledge lulls one into a false sense of security. It enables one to interpret and deal with complex or ambiguous situations on the basis of context cues not visible to users, as well as a knowledge of the computer system that users do not possess. The result is a system that appears perfectly useful to its designers but one that the user is unable or unwilling to face up to and master.

What makes a system difficult to use in the eyes of its user? Following is a list of several contributing factors:

Too much flexibility. When the needs of a user are not well understood, the tendency is to build into a system as many functions as possible. More functions result in higher interface complexity. Higher complexity requires more learning and results in less efficient human performance. In general, as the flexibility of a system increases, its usability decreases.

Use of jargon. Systems often speak in a strange language. Words that are completely alien to the office or home environment or used in different contexts, such as *filespec*, *abend*, *segment*, and *boot*, proliferate. Learning to use a system often requires learning a new language.

Non-obvious design. Complex or novel design elements are not obvious or intuitive, but they must be mastered. Operations may have prerequisite conditions that must be satisfied before they can be accomplished, or outcomes may not always be immediate, obvious, or visible. The overall framework of the system may be invisible, the effect of which is that results cannot always be related to the actions that accomplish them.

Fine distinctions. Different actions may accomplish the same thing, depending upon when they are performed, or different things may result from the same action. Often these distinctions are minute and difficult to keep track of. Critical distinctions are not made at the appropriate time, or distinctions having no real consequence are made instead, as illustrated by the user who insisted that problems were caused by pressing the Enter key "in the wrong way."

Disparity in problem-solving strategies. People learn best by doing. They have trouble following directions and do not always read instructions before taking an

https://hemanthrajhemu.github.io

action. Human problem solving can best be characterized as "error-correcting" or "trial and error," whereby a tentative solution is formulated based upon the available evidence and then tried. This tentative solution often has a low chance of success, but the action's results are used to modify one's next attempt and so increase the chance of success. Most early computer systems, however, have enforced an "error-preventing" strategy, which assumes that a person will not take an action until a high degree of confidence exists in its success. The result is that when people head down wrong one-way paths, they often get entangled in situations that are difficult or impossible to get out of. The last resort action? Turn off the computer and restart.

Design inconsistency. The same action may have different names: for example, "save" and "keep," "write" and "list." The same command may cause different things to happen. The same result may be described differently: for example, "not legal" and "not valid." Or the same information may be ordered differently on different screens. The result is that system learning becomes an exercise in rote memorization. Meaningful or conceptual learning becomes very difficult.

Responses to Poor Design

Unfortunately, people remember the one thing that went wrong, not the many that go right, so problems are ascribed an abnormal level of importance. Errors are a symptom of problems. The magnitude of errors in a computer-based system has been found to be as high as 46 percent for commands, tasks, or transactions. Errors, and other problems that befuddle one, lead to a variety of psychological and physical user responses.

Psychological

Typical psychological responses to poor design are as follows:

Confusion. Detail overwhelms the perceived structure. Meaningful patterns are difficult to ascertain, and the conceptual model or underlying framework cannot be understood or established.

Annoyance. Roadblocks that prevent a task from being completed, or a need from being satisfied, promptly and efficiently lead to annoyance. Inconsistencies in design, slow computer reaction times, difficulties in quickly finding information, outdated information, and visual screen distractions are a few of the many things that may annoy users.

Frustration. An overabundance of annoyances, an inability to easily convey one's intentions to the computer, or an inability to finish a task or satisfy a need can cause frustration. Frustration is heightened if an unexpected computer response cannot be undone or if what really took place cannot be determined. Inflexible and unforgiving systems are a major source of frustration.

Panic or stress. When a system taxes a person's perceptual and cognitive abilities, panic and stress are often the results. Overcomplex systems and procedures, unexpected long delays during times of severe or unusual pressure, or long

https://hemanthrajhemu.github.io

response times when the user is operating under a deadline or dealing with an irate customer are examples of situations that can lead to panic and stress.

Boredom. When a person's perceptual and cognitive abilities are underused, the person often becomes apathetic and bored. Boredom results from, among other things, improper computer pacing (slow response times or long download times) or oversimplistic jobs or tasks. A bored individual is also likely to make more performance errors.

These psychological responses diminish user effectiveness because they are severe blocks to concentration. Thoughts irrelevant to the task at hand are forced to the user's attention, and necessary concentration is impossible. The result, in addition to higher error rates, is poor performance, anxiety, and dissatisfaction.

Physical

When people do something, they expect the benefits of what they are doing to outweigh the cost or effort to do it. When effort and the aforementioned psychological responses exceed the perceived benefits, the results are often the following physical reactions:

Abandonment of the system. The system is rejected and other information sources are relied upon. These sources must, of course, be available and the user must have the discretion to perform the rejection. In business systems this was a common reaction of managerial and professional personnel. With the Web, almost all users can exercise this option.

Partial use of the system. Only a portion of the system's capabilities are used, usually those operations that are easiest to perform or that provide the most benefits. Historically, this has been the most common user reaction to most computer systems. Many aspects of many systems often go unused.

Indirect use of the system. An intermediary is placed between the would-be user and the computer. Again, because this requires high status and discretion, it is another typical response of managers or others with authority.

Modification of the task. The task is changed to match the capabilities of the system. This is a prevalent reaction when the tools are rigid and the problem is unstructured, as in scientific problem solving.

Compensatory activity. Additional actions are performed to compensate for system inadequacies. A common example is the manual reformatting of information to match the structure required by the computer. This is a reaction common to workers whose discretion is limited, such as clerical personnel.

Misuse of the system. The rules are bent to shortcut operational difficulties. This requires significant knowledge of the system and may affect system integrity.

Direct programming. The system is reprogrammed by its user to meet specific needs. This is a typical response of the sophisticated worker.

These physical responses greatly diminish user efficiency and effectiveness. They force the user to rely upon other information sources, to fail to use a system's complete capabilities, or to perform time-consuming "work-around" actions.

https://hemanthrajhemu.github.io

People and Their Tasks

Today's business user is often overworked, fatigued, and continually interrupted. The home user may also experience these same conditions, and the pressures associated with children and family life as well. All computer users do tend to share the following: They tend not to read documentation, they do not understand well the problems the computer can aid in solving, and they know little about what information is available to meet their needs. Moreover, the users' technical skills have often been greatly overestimated by the system designer, who is frequently isolated psychologically and physically from the users' situations. Unlike the users, the designer is capable of resolving most system problems and ambiguities through application of experience and technical knowledge. Often the designer cannot really believe that anyone is incapable of using the system created.

The user, while being subjected to the everyday pressures of the office and home, frequently does not care about how technically sophisticated a system or Web site is. The user may even be computer illiterate, and possibly even antagonistic. He or she wants to spend time using a computer, not learning to use it. His or her objective is simply to get some work done, a task performed, or a need satisfied. Today, many users have also learned to expect a certain level of design sophistication. It is in this environment that a system will be placed.

Important Human Characteristics in Design

People are complex organisms with many attributes that have an important influence on interface design. Of particular importance in design are perception, memory, visual acuity, foveal and peripheral vision, sensory storage, information processing, learning, skill, and individual differences.

Perception

Perception is our awareness and understanding of the elements and objects of our environment through the physical sensation of our various senses, including sight, sound, smell, and so forth. Perception is influenced, in part, by *experience*. People classify stimuli based on models stored in our memories and in this way achieve understanding. In essence, people tend to match objects or sensations perceived to things already known. Comparing the accumulated knowledge of a child with that of an adult in interpreting the world is a vivid example of the role of experience in perception.

Other perceptual characteristics include the following:

Proximity. Our eyes and mind see objects as belonging together if they are near each other in space.

Similarity. Our eyes and mind see objects as belonging together if they share a common visual property, such as color, size, shape, brightness, or orientation.

Matching patterns. We respond similarly to the same shape in different sizes. The letters of the alphabet, for example, possess the same meaning, regardless of physical size.

https://hemanthrajhemu.github.io

Succinctness. We see an object as having some perfect or simple shape because perfection or simplicity is easier to remember.

Closure. Our perception is synthetic; it establishes meaningful wholes. If something does not quite close itself, such as a circle, square, triangle, or word, we see it as closed anyway.

Unity. Objects that form closed shapes are perceived as a group.

Continuity. Shortened lines may be automatically extended.

Balance. We desire stabilization or equilibrium in our viewing environment. Vertical, horizontal, and right angles are the most visually satisfying and easiest to look at.

Three-dimensional projection. When certain visual cues are present, we tend to see objects and patterns as three-dimensional. For example, an overlapped object, the smaller of similar objects, and shaded objects are perceived as farther away.

Top-down lighting bias. We interpret shaded or dark areas of an object as shadows caused by a light source from above. Objects with top-down lighting are seen as natural; those with other light orientations are seen as unnatural.

Expectancies. Perception is also influenced by expectancies; sometimes we perceive not what is there but what we expect to be there. Missing a spelling mistake in proofreading something we write is often an example of a perceptual expectancy error; we see not how a word *is* spelled, but how we *expect* to see it spelled.

Context. Context, environment, and surroundings also influence individual perception. For example, two drawn lines of the same length may look the same length or different lengths, depending on the angle of adjacent lines or what other people have said about the size of the lines.

Signals versus noise. Our sensing mechanisms are bombarded by many stimuli, some of which are important and some of which are not. Important stimuli are called signals; those that are not important or unwanted are called noise. This is known as figure-ground perception. Figure is the signal; ground is everything else forming the background. Signals, or figures, are more quickly comprehended if they are easily distinguishable from noise, or background, in our sensory environment. Noise interferes with the perception of signals to the extent that they are similar to one another. Noise can even mask a critical signal. For example, imagine a hidden word puzzle where meaningful words are buried in a large block matrix of alphabetic characters. The signals, alphabetic characters constituting meaningful words, are masked by the matrix of meaningless letters. The elements of a screen assume the quality of signal or noise, depending on the actions and thought processes of the user. Once a screen is first presented and has to be identified as being the correct one, the screen's title may be the signal, the other elements it contains simply being noise. When the screen is being used, the data it contains becomes the signal, and the title now reverts to noise. Other elements of the screen rise and fall in importance, assuming the roles of either signals or noise, depending on the user's needs of the moment. The goal in design is to allow screen elements to easily assume the quality of signal or noise, as the needs and tasks of the user change from moment to moment.

The goal in design, then, is to utilize our perceptual capabilities so that a screen can be structured in the most meaningful and obvious way.

https://hemanthrajhemu.github.io

Memory

Memory is not the most stable of human attributes, as anyone who has forgotten why they walked into a room, or forgotten a very important birthday, can attest. Today, memory is viewed as consisting of two components: long-term and short-term (or working) memory. This has not always been the case. In the 1950s most researchers believed there was only one memory system; the short-term component was not recognized or accepted. It was in this era that the classic memory study was published (Miller, 1956) indicating that memory limit is 7 ± 2 "chunks" of information. Shortly after this the concept of a short-term memory was identified, and in the 1970s the view of short-term memory was broadened and called "working memory."

Short-term memory, or working memory, receives information from either the senses or long-term memory, but usually cannot receive both at once because the senses are processed separately. Within short-term memory a limited amount of information processing takes place. Information stored within it is variously thought to last from 5 to 30 seconds, with the lower number being the most reasonable speculation. Based upon research over the years, estimates of working memory storage capacity has gradually been lowered from Miller's 7 ± 2 items to a size of 3 to 4 items today (Koyani et al., 2004). Keep in mind that as people age, their working memory capacity does diminish (Laguna and Babcock, 2000).

Knowledge, experience, and familiarity govern the size and complexity of the information that can be remembered. To illustrate, most native English-speaking people would find remembering English words much easier than remembering an equal number of words in Russian. For a Russian-speaking person the opposite would be true. Short-term memory is easily overloaded. It is highly susceptible to the interference of such distracting tasks as thinking, reciting, or listening, which are constantly erasing and overwriting it. Remembering a telephone number long enough to complete the dialing operation taxes the memory of many people.

In performance, research indicates that a greater working memory is positively related to increased reading comprehension, drawing inferences from text, reasoning skill, and learning technical information (Baddeley, 1992). Research indicates, as well, that when performing complex tasks, working memory can be increased through applying two senses, vision and audition, rather than one (Williams, 1998). Research also indicates that performance can be degraded when a person must attend to multiple information sources, and then must integrate the information before understanding occurs. Mayes et al. (2000), for example, found that reading speed is degraded when working memory is also being used.

Long-term memory contains the knowledge we possess. Information received in short-term memory is transferred to it and encoded within it, a process we call learning. It is a complex process requiring some effort on our part. The learning process is improved if the information being transferred from short-term memory has structure and is meaningful and familiar. Learning is also improved through repetition and deep analysis. Thinking hard about information improves recall of the same information (Lidwell et al., 2003).

Unlike short-term memory with its distinct limitations, long-term memory capacity is thought to be unlimited. An important memory consideration, with significant implications for interface design, is the difference in ability to recognize or recall words. The human active vocabulary (words that can be recalled) typically ranges

https://hemanthrajhemu.github.io

between 2,000 and 3,000 words. Passive vocabulary (words that can be recognized) typically numbers about 100,000. Recognition tasks provide memory cues that aid searching through long-term memory. Our power of recognition, therefore, is much greater than our power of recall, and this phenomenon should be utilized in design. To do this, one should present, whenever possible, lists of alternatives to remind people of the choices they have.

MAXIM **Minimize the need for a mighty memory.**

Other general ways to reduce user memory loads, reduce the need for mental integration, aid recall, and expand working memory, thus enhancing system usability, include the following:

- Presenting information in an organized, structured, familiar, and meaningful way.
- Giving the user control over the pace of information presentation.
- Placing all required information for task performance in close physical proximity. People can remember a few items for only 3 or 4 seconds.
- Placing important items at the beginning or end of listing, not in the middle. These items will be learned faster and recalled better.
- Placing information that must be compared in close proximity so memory does not have to be taxed.
- Not requiring people to perform other tasks using working memory if screen-reading speed is important.
- Making important items unique or distinctive in some manner because the likelihood of their being remembered will be increased. Highlighting key elements is one way to do this.

Sensory Storage

Sensory storage is the buffer where the automatic processing of information collected from our senses takes place. It is an unconscious process, large, attentive to the environment, quick to detect changes, and constantly being replaced by newly gathered stimuli. In a sense, it acts like radar, constantly scanning the environment for things that are important to pass on to higher memory.

Though seemingly overwhelmed at times by noise, it can occasionally detect, proverbially, a tree through a forest. One good example is what is sometimes called the "cocktail party effect." Have you ever been at a party when, across the room, through the din of voices, someone mentions your name and you hear it? In spite of the noise, your radar was functioning.

Repeated and excessive stimulation can fatigue the sensory storage mechanism, making it less attentive and unable to distinguish what is important (called *habituation*). Avoid unnecessarily stressing it. Design the interface so that all aspects and elements serve a definite purpose. Eliminating interface noise will ensure that important things are less likely to be missed.

https://hemanthrajhemu.github.io

Visual Acuity

The capacity of the eye to resolve details is called *visual acuity*. It is the phenomenon that results in an object becoming more distinct as we turn our eyes toward it and rapidly losing distinctness as we turn our eyes away — that is, as the visual angle from the point of fixation increases. It has been shown that relative visual acuity is approximately halved at a distance of 2.5 degrees from the point of eye fixation (Bouma, 1970). Therefore, a 5-degree diameter circle centered around an eye fixation character on a display has been recommended as the area near that character (Tullis, 1983) or the maximum length for a displayed word (Danchak, 1976).

If one assumes that the average viewing distance of a display screen is 19 inches (475 mm), the size of the area on the screen of optimum visual acuity is 1.67 inches (41.8 mm) in diameter. Assuming "average" character sizes and character and line spacing, the number of characters on a screen falling within this visual acuity circle is 88, with 15 characters being contained on the widest line, and seven rows being consumed, as illustrated in Figure 1.1.

The eye's sensitivity increases for those characters closest to the fixation point (the "0") and decreases for those characters at the extreme edges of the circle (a 50/50 chance exists for getting these characters correctly identified). This may be presumed to be a visual "chunk" of a screen and will have implications for screen grouping guidelines to be presented later. (Remember, it is the physical size of the circle, 5 degrees, that is critical, not the number of characters. A larger or smaller character size will decrease or increase the number of viewable characters.) These studies and recommendations were all generated with early generation displays, well before the advent of the personal computer possessing great flexibility in font sizes and styles. With today's better displays, the recommended viewing distance has been increased to about 24 inches. This increased distance, along with the many possible variations in font sizes, makes calculation of the exact "chunk" size and viewable number of characters difficult to calculate. The important principle to keep in mind is that fairly small visual chunks will exist on screens and these chunks should be considered in design.

The eye is also never perfectly steady as it sees; it trembles slightly. This tremor improves the detection of edges of objects being looked at, thus improving acuity. This tremor, however, can sometimes create problems. Patterns of closely spaced lines or dots are seen to shimmer. This movement can be distracting and disturbing. Patterns for fill-in areas of screens (bars, circles, and so on) must be carefully chosen to avoid this visual distraction.

```
    3213123
  54321212345
 6543211123456
765432101234567
 6543211123456
  54321212345
    3213123
```

Figure 1.1 Size of area of optimum visual acuity on a screen.

https://hemanthrajhemu.github.io

Foveal and Peripheral Vision

Foveal vision is used to focus directly on something; *peripheral vision* senses anything in the area surrounding the location we are looking at, but what is there cannot be clearly resolved because of the limitations in visual acuity just described. Foveal and peripheral vision maintain, at the same time, a cooperative and a competitive relationship. Peripheral vision can aid a visual search, but can also be distracting.

In its cooperative nature, peripheral vision is thought to provide clues to where the eye should go next in the visual search of a screen. Patterns, shapes, and alignments peripherally visible can guide the eye in a systematic way through a screen.

In its competitive nature, peripheral vision can compete with foveal vision for attention. What is sensed in the periphery is passed on to our information-processing system along with what is actively being viewed foveally. It is, in a sense, visual noise. Mori and Hayashi (1993) experimentally evaluated the effect of windows in both a foveal and peripheral relationship and found that performance on a foveal window deteriorates when there are peripheral windows, and the performance degradation is even greater if the information in the peripheral is dynamic or moving. Care should be exercised in design to utilize peripheral vision in its positive nature, avoiding its negative aspects.

Information Processing

The information that our senses collect that is deemed important enough to do something about then has to be processed in some meaningful way. Recent thinking (Lind, Johnson, and Sandblad, 1992) is that there are two levels of information processing going on within us. One level, the highest level, is identified with consciousness and working memory. It is limited, slow, and sequential, and is used for reading and understanding. You are utilizing this higher level now reading this book.

In addition to this higher level, there exists a lower level of information processing, and the limit of its capacity is unknown. This lower level processes familiar information rapidly, in parallel with the higher level, and without conscious effort. We look rather than see, perceive rather than read. Repetition and learning results in a shift of control from the higher level to the lower level.

Both levels function simultaneously, the higher level performing reasoning and problem solving, the lower level perceiving the physical form of information sensed. You've probably experienced this difference in working with screens. When a screen is displayed, you usually will want to verify that it is the one you want. If you're new to a system, or if a screen is new to you, you rely on its concrete elements to make that determination, its title, the controls and information it contains, and so forth. You consciously look at the screen and its components using this higher-level processing. As you become experienced and familiar with screens, however, you can identify a newly presented screen very quickly with just a momentary glance. Just its shape and structure adequately communicate to you that it is the correct screen for the context in which you are working. Your reasoning and problem solving continues unhindered; your lower-level information processing has assumed the screen identity task.

What assists this lower-level information processing? Visual distinctiveness of a screen is a strong contributor. If a screen is jammed with information and cluttered, it

https://hemanthrajhemu.github.io

loses its uniqueness and can only be identified through the more time-consuming, and thought-interrupting, reading process.

Higher-level processing can be impeded by a phenomenon called interference. Perception and cognition involve many different mental systems, each processing information independently of one another. The results of this processing are communicated to working memory where it is interpreted. Meaningful outputs from the higher-level memory are interpreted quickly. Interference occurs when perception and cognition are exposed to conflicting mental processes that must then be resolved. Some types of interference from Lidwell et al. (2003) include the following:

Stoop interference. One aspect of a stimulus triggers a mental process that conflicts with another aspect of the stimulus. For example, it takes longer to name the color of a printed word if the word is printed in a different color than the meaning of the word itself. A Stop sign displayed in green and a Go sign in red will also create a mental conflict.

Proactive interference. A person's existing memories interfere with learning. In learning a new language, for example, interference occurs when people apply their native language grammar to the new language. Similarly, applying the interaction procedures for a learned computer system will interfere with learning new interaction rules for a new system.

Retroactive interference. Learning interferes with existing memories. Learning a new telephone number can interfere with those numbers already memorized. Similarly, learning a new computer system can interfere with a system already memorized.

Minimize interference by avoiding designs that create conflicting mental processes as follows:

- Avoid coding combinations, including color and icons, which create interference.
- Understand and utilize a person's learned mental models in design (see following section).

Another characteristic of human information processing is that the time it takes to make a decision or decide on an option increases as the number of alternative choices increases. Called Hick's Law, it can be applied to simple decision-making tasks, not complex ones involving activities like reading, problem solving, and hierarchies of menus (Lidwell et al., 2003). Apply this law when designing systems where a set of options must be presented. When designing time-critical tasks, minimize errors and reduce human response times by doing the following:

- Minimize the number of options presented to the user.

Mental Models

As a result of our experiences and culture, we develop mental models of things and people we interact with. A mental model is simply an internal representation of a person's current understanding of something. Usually a person cannot describe this mental mode and most often is unaware it even exists. Mental models are gradually developed to understand something, explain things, make decisions, do something, or

https://hemanthrajhemu.github.io

interact with another person. Mental models also enable a person to predict the actions necessary to do things if the action has been forgotten or has not yet been encountered.

When confronting a new computer system, people will bring their own expectations and preconceptions based upon mental models they have formed working with other systems and doing things in their daily life. If the system conforms to the mental models a person has developed, the model is reinforced and the system's use feels more "intuitive." If not, interference occurs and difficulties in learning to use the system will be encountered. This is why in design it is critical that a user's mental models be identified and understood.

A person already familiar with one computer system will bring to another system a mental model containing specific visual and usage expectations. If the new system complies with already-established models, it will be much easier to learn and use. One key to forming a transferable mental model of a system is design consistency and design standards.

When designing, then,

- Design with people's mental models in mind.

There is one caveat, however. Never compromise goals and design simply to conform to existing mental models. In situations where the existing model does not fit well, the best solution is to present a clear and consistent new model for the user to learn.

Movement Control

Once data has been perceived and an appropriate action decided upon, a response must be made; in many cases the response is a movement. In computer systems, movements include such activities as pressing keyboard keys, moving the screen pointer by pushing a mouse or rotating a trackball, or clicking a mouse button. Particularly important in screen design is Fitts' Law (1954). This law states that the time to acquire a target is a function of the distance to and size of the target.

This simply means that the bigger the target is, or the closer the target is, the faster it will be reached. The implications in screen design are as follows:

- Provide large objects for important functions.
- Take advantage of the "pinning" actions of the sides, top, bottom, and corners of the screen.

Big buttons are better than small buttons. They provide a larger target for the user to access with the screen pointer. Create toolbar icons that "bleed" into the edges of a display, rather than those that leave a one-pixel, nonclickable edge along the display boundary. The edge of the screen will stop or "pin" the pointer's movement at a position over the toolbar, permitting much faster movement to the toolbar. A one-pixel edge will require more careful positioning of the pointer over the toolbar.

Learning

Learning, as mentioned earlier, is the process of encoding in long-term memory information that is contained in short-term memory. It is a complex process that requires

https://hemanthrajhemu.github.io

some effort on our part. Our ability to learn is important — it clearly differentiates people from machines. Given enough time, people can improve their performance in almost any task. Too often, however, designers use our learning ability as an excuse to justify complex design. Because people can be taught to walk a tightrope is no excuse for incorporating tightropes in a design when walkways are feasible.

A design developed to minimize human learning time can greatly accelerate human performance. People prefer to stick with what they know, and they prefer to jump in and get started. Unproductive time spent learning is something frequently avoided.

Regarding the learning process, evidence derived from studies of people learning a computer system parallels that found in studies of learning in other areas. People prefer to be active, to explore, and to use a trial-and-error approach. There is also evidence that people are very sensitive to even minor changes in the user interface, and that such changes may lead to problems in transferring from one system to another. Moreover, just the perception of having to learn huge amounts of information is enough to keep some people from even using a system. Learning can be enhanced if it

- Allows skills acquired in one situation to be used in another somewhat like it. Design consistency accomplishes this.
- Provides complete and prompt feedback.
- Is phased; that is, it requires a person to know only the information needed at that stage of the learning process.

A significant by-product of learning is that it enables users to anticipate the location of common screen or page elements before they are displayed (Bernard, 2002; Byrne et al., 1999). Experienced users often begin moving a mouse pointer to the area of an expected target before the target appears on the screen.

Skill

The goal of human performance is to perform skillfully. To do so requires linking inputs and responses into a sequence of actions. The essence of skill is performance of actions or movements in the correct time sequence with adequate precision. It is characterized by consistency and economy of effort. Economy of effort is achieved by establishing a work pace that represents optimum efficiency. It is accomplished by increasing mastery of the system through such things as progressive learning of shortcuts, increased speed, and easier access to information or data.

Skills are hierarchical in nature, and many basic skills may be integrated to form increasingly complex ones. Lower-order skills tend to become routine and may drop out of consciousness. System and screen design must permit development of increasingly skillful performance.

Performance Load

The greater the effort to perform a task, the less likely the task will be accomplished successfully, or even at all. The degree of effort that must be expended is commonly called performance load, the path of least resistance principle, or the principle of least effort (Lidwell et al., 2003). Performance load consists of two types: cognitive and kinematic.

https://hemanthrajhemu.github.io

Cognitive load is the amount of mental activity required to perform a task or achieve an objective. The implementation of the graphical user interface significantly reduced the cognitive load on computer users. One of its advantages is that it replaced commands that previously had to be recalled and formatted properly with commands in menus that simply had to be recognized.

Kinematic load is the degree of physical activity or effort necessary to perform a task or achieve an objective. The graphical user interface also reduced user effort by substituting a simple mouse click for a typed command consisting of several or more characters.

Cognitive load can be reduced by doing such things as

- Eliminating noise or unnecessary information from screens.
- Properly formatting and grouping information.
- Providing aids to allow the user to rely on powers of recognition, not recall.
- Automating tasks that require extensive memory.

Aids to reducing kinematic load include

- Minimizing the number of steps to accomplish tasks.
- Minimizing control actions and movements.
- Automating repetitive tasks.

A person's estimation of the degree of effort to use a particular computer system or application need not be based upon extensive experience with the system. A brief initial use of a system may result in a judgment being quickly reached that the estimated effort to be expended is not worth the expected results. Rejection then occurs. Rejection may also occur if a person simply thinks something will require great effort to use.

Individual Differences

In reality, there is no average user. A complicating but very advantageous human characteristic is that we all differ — in looks, feelings, motor abilities, intellectual abilities, learning abilities and speed, and so on. In a keyboard data entry task, for example, the best typists will probably be twice as fast as the poorest and make 10 times fewer errors.

For several computer-based tasks Nielsen (2006d) has compiled the difference in task completion speeds for users at the 25th percentile (75 percent of users are faster) and the 75th percentile (75 percent of users are slower). The results, based upon Nielsen's data as well as that of Egan (1988), are shown in Table 1.1.

For the simplest task, text editing, the faster users are 1.8 times speedier than the slower users. For programming, the most complicated task, the faster users are three times speedier than those slower. Programming only exceeded Web use in the faster-slower difference, achieving a ratio of 2.4.

Individual differences complicate design because the design must permit people with widely varying characteristics to satisfactorily and comfortably learn the task or job, or use the Web site. In the past this usually resulted in bringing designs down to the level of lowest abilities or selecting people with the minimum skills necessary to perform a job. But technology now offers the possibility of tailoring jobs to the specific

https://hemanthrajhemu.github.io

needs of people with varying and changing learning or skill levels. Multiple versions of a system can be easily created. Design must provide for the needs of all potential users.

Table 1.1: Task Completion Speed Ratios for 25th and 75th Percentile Users

COMPUTER TASK USE	RATIO OF 25TH PERCENTILE (SLOW) TO 75TH PERCENTILE (FAST) USERS
Text Editing (Egan)	1.8
Personal Computing (Nielsen)	1.9
Information Search (Non Web) (Egan)	2.2
Web Use (Nielsen)	2.4
Programming (Egan)	3.0

Table 1.2: Important User/Task Considerations

KNOWLEDGE/EXPERIENCE	
Computer Literacy	Highly technical or experienced, moderate computer experience, or none.
System Experience	High, moderate, or low knowledge of a particular system and its methods of interaction.
Application Experience	High, moderate, or low knowledge of similar systems.
Task Experience	Level of knowledge of job and job tasks.
Other Systems Use	Frequent or infrequent use of other systems in doing job.
Education	High school, college, or advanced degree.
Reading Level	Less than 5th grade, 5th–12th, more than 12th grade.
Typing Skill	Expert (135 WPM), skilled (90 WPM), good (55 WPM), average (40 WPM), or “hunt and peck” (10 WPM).
Native Language or Culture	English, another, or several.
JOB/TASK/NEED	
Type of System Use	Mandatory or discretionary use of the system.
Frequency of Use	Continual, frequent, occasional, or once-in-a-lifetime use of system.
Task or Need Importance	High, moderate, or low importance of the task being performed.
Task Structure	Repetitiveness or predictability of tasks being automated, high, moderate, or low.
Social Interactions	Verbal communication with another person required or not required.

https://hemanthrajhemu.github.io

Table 1.2 *(continued)*

Primary Training	Extensive or formal training, self-training through manuals, or no training.
Turnover Rate	High, moderate, or low turnover rate for jobholders.
Job Category	Executive, manager, professional, secretary, clerk.
Lifestyle	For Web e-commerce systems, includes hobbies, recreational pursuits, and economic status.
PSYCHOLOGICAL CHARACTERISTICS	
Attitude	Positive, neutral, or negative feeling toward job or system.
Motivation	Low, moderate, or high due to interest or fear.
Patience	Patience or impatience expected in accomplishing goal.
Expectations	Kinds and reasonableness.
Stress Level	High, some, or no stress generally resulting from task performance.
Cognitive Style	Verbal or spatial, analytic or intuitive, concrete or abstract.
PHYSICAL CHARACTERISTICS	
Age	Young, middle aged, or elderly.
Gender	Male or female.
Handedness	Left, right, or ambidextrous.
Disabilities	Blind, defective vision, deafness, motor handicap.

Derived from Mayhew (1992).

Human Considerations in the Design of Business Systems

The human characteristics described previously are general qualities we all possess. There are a host of other human aspects in which people may vary greatly. These are also important and must be identified in the design process. The kinds of user/task characteristics that must be established are summarized in Table 1.2 and more fully described in the following paragraphs. Many of these considerations are derived from Mayhew (1992).

The User's Knowledge and Experience

The knowledge possessed by a person, and the experiences undergone, shape the design of the interface in many ways. The following kinds of knowledge and experiences should be identified.

https://hemanthrajhemu.github.io

Computer Literacy

Are the users highly technical such as programmers or experienced data entry clerks? Do they have moderate computer experience or none at all? Will they be familiar with computer concepts and terms, the keyboard and its keys, and a mouse or other input mechanisms? If so, how familiar?

System Experience

Are users already familiar with the interaction requirements of the new system, somewhat familiar, or not familiar at all? Have users worked with similar systems? If so, what kind? What are the similarities? The differences? The same questions can be asked for Web systems.

At one time or another, various schemes have been proposed to classify the different and sometimes changing characteristics of people as they become more experienced using a system. Words to describe the new, relatively new, or infrequent user have included *naive, casual, inexperienced,* or *novice*. At the other end of the experience continuum lie terms such as *experienced, full-time, frequent, power,* or *expert*. In between these extremes is a wide range of *intermediate* or *intermittent* users. The words describing these categories are less important than the behavioral characteristics they imply. Experience to date has uncovered some basic differences in the feelings of ease of use based upon proficiency level. What is easy for the new user is not always perceived as easy for the "old hand," and vice versa. For simplicity in this discussion, the term *novice* will be used for the new user, the term *intermediate* for those in between, and the term *expert*, for the most proficient.

Novice users in business systems have been found to

- Depend on system features that assist recognition memory: menus, prompting information, and instructional and help screens.
- Need restricted vocabularies, simple tasks, small numbers of possibilities, and very informative feedback.
- View practice as an aid to moving up to expert status.

For years novice users have been told they are stupid and have been conditioned to accept the blame for their failure to understand and use a system or product. (Note the popularity of the manuals with "dummy" in the title.) These novice users have been forced to struggle through their "dumbness" to achieve mastery, because they had few other choices. If people cannot effectively use a system, then, who is to blame? It would seem the fault lies in the design of the system, for people cannot be redesigned.

Experts, on the other hand

- Rely upon free recall.
- Expect rapid performance.
- Need less informative feedback.
- Seek efficiency by bypassing novice memory aids, reducing keystrokes, chunking and summarizing information, and introducing new vocabularies.

The needs of the intermediate user fall somewhere in between these extremes.

https://hemanthrajhemu.github.io

In actuality, the user population of most systems is spread out along the continuum anchored by these two extremes. And, equally important, the behavior of any one user at different times may be closer to one extreme or the other. A person may be very proficient — an expert — in one aspect of a system and ignorant — a novice — in other aspects at the same time. Becoming an expert in use of the Web is particularly challenging. The faces of Web sites are continually changing, and individual sites are rarely used enough for expert competence to be established.

Microsoft has identified the problems that novice and intermediate users have in using their Windows systems. Novice users often have the following difficulties:

- Dragging and double-clicking using the mouse. Distinguishing between double-clicks and two separate clicks is particularly confusing.
- Managing windows. That overlapping windows represent a three-dimensional space is not always realized. Hidden windows are assumed to be gone and no longer exist.
- Managing files. The organization of files and folders nested more than two levels deep is difficult to understand. Structure is not as apparent as with physical files and folders.

Intermediate Windows users may understand the file hierarchy but often have difficulties with other aspects of file management such as moving or copying. These kinds of problems must be considered in design.

Exactly how experts and novices actually differ from one another in terms of knowledge, problem-solving behavior, and other human characteristics has been the subject of some research in recent years.

Experts in summarizing, have been found to possess the following traits:

- They possess an integrated conceptual model of a system.
- They possess knowledge that is ordered more abstractly and more procedurally.
- They organize information more meaningfully, orienting it toward their task.
- They structure information into more categories.
- They are better at making inferences and relating new knowledge to their objectives and goals.
- They pay less attention to low-level details.
- They pay less attention to surface features of a system.

Novices exhibit the following characteristics:

- They possess a fragmented conceptual model of a system.
- They organize information less meaningfully, orienting it toward surface features of the system.
- They structure information into fewer categories.
- They have difficulty in generating inferences and relating new knowledge to their objectives and goals.
- They pay more attention to low-level details.
- They pay more attention to surface features of the system.

https://hemanthrajhemu.github.io

A well-designed system, therefore, must support at the same time novice and expert behavior, as well as all levels of behavior in between. The challenge in design is to provide for the expert's needs without introducing complexity for those less experienced. In general the following graphical system aspects are seen as desirable expert shortcuts:

- Mouse double-clicks.
- Pop-up menus.
- Tear-off or detachable menus.
- Command lines.

Web Page Considerations

In Web page design, novice users have been found to need overviews, buttons to select actions, and guided tours; intermediate users want an orderly structure, obvious landmarks, reversibility, and safety as they explore; and experts like smooth navigation paths, compact but in-depth information, fast page downloads, extensive services to satisfy their varied needs, and the ability to change or rearrange the interface.

Application Experience

Have users worked with a similar application (for example, word processing, airline reservation, and so on)? Are they familiar with the basic application terms? Or does little or no application experience exist?

Task Experience

Are users experienced with the task being automated? If it is an insurance claim system, do users have experience with paying claims? If it is a banking system, do users have experience in similar banking applications? Or do users possess little or no knowledge of the tasks the system will be performing?

Other System Use

Will the user be using other systems while using the new system? If so, they will bring certain habits and expectancies to the new system. The more compatibility between systems, the lower the learning requirements for the new system and the higher the productivity using all systems.

MYTH Developers have been working with users for a long time. They always know everything users want and need.

Education

What is the general educational level of users? Do they generally have high school degrees, college degrees, or advanced degrees? Are the degrees in specialized areas related to new system use?

https://hemanthrajhemu.github.io

Reading Level

For textual portions of the interface, the vocabulary and grammatical structure must be at a level that is easily understood by the users. Reading level can often be inferred from one's education level. A recent study found, however, that

- The average reading level in North America is at the 8th to 9th grade level.
- About one-fifth of all adults read at the 5th grade level or below.
- Adults tend to read at least one or two grades below the last school grade completed (D'Allesandro et al., 2001).

Typing Skill

Is the user a competent typist or of the hunt-and-peck variety? Is he or she familiar with the standard keyboard layout or other, newer layouts? A competent typist may prefer to interact with the system exclusively through the keyboard, whereas the unskilled typist may prefer the mouse.

Native Language and Culture

Do the users speak English, another language, or several other languages? Will the screens be in English or in another language? Other languages often impose different screen layout requirements. Are there cultural or ethnic differences between users? Will icons, metaphors, and any included humor or clichés be meaningful for all the user cultures?

Table 1.3 summarizes the native languages of those online around the world (global-reach.biz, September 2004). While native English speakers account for only about one-third of the user population, a significantly larger percent of worldwide users will have varying degrees of proficiency in reading, writing, and speaking English.

Table 1.3: Native Languages of Online Users

English	35.2%
Chinese	13.7%
Spanish	9.0%
Japanese	8.4%
German	6.9%
French	4.2%
Korean	3.9%
Italian	3.8%
Portuguese	3.1%
Dutch	1.7%
Other	10.1%

From global-reach.biz (September 2004).

https://hemanthrajhemu.github.io

Most of these kinds of user knowledge and experience are independent of one another, so many different user profiles are possible. It is also useful to look ahead, assessing whether future users will possess the same qualities.

The User's Tasks and Needs

The user's tasks and needs are also important in design. The following should be determined.

Mandatory or Discretionary Use

Users of the earliest computer systems were mandatory or nondiscretionary. That is, they required the computer to perform a task that, for all practical purposes, could be performed no other way. Characteristics of mandatory use can be summarized as follows:

- The computer is used as part of employment.
- Time and effort in learning to use the computer are willingly invested.
- High motivation is often used to overcome low usability characteristics.
- The user may possess a technical background.
- The job may consist of a single task or function.

The mandatory user must learn to live comfortably with a computer, for there is really no other choice. Examples of mandatory use today include a flight reservations clerk booking seats, an insurance company employee entering data into the computer so a policy can be issued, and a programmer writing and debugging a program. The toll exacted by a poorly designed system in mandatory use is measured primarily by productivity: for example, errors and poor customer satisfaction.

In recent years as technology and the Web have expanded into the office, the general business world, and the home, a second kind of user has been more widely exposed to the benefits and problems of technology. In the business office this other kind of user is much more self-directed than the mandatory user, not being told how to work but being evaluated on the results of his or her efforts. For him or her, it is not the means but the results that are most important. In short, this user has never been told how to work in the past and refuses to be told so now. This newer kind of user is the office executive, manager, or other professional, whose computer use is completely *discretionary*.

In the general business world and at home, discretionary users also include the people who are increasingly being asked to, or want to, interact with a computer in their everyday lives. Examples of this kind of interaction include library information systems, bank automated teller machines (ATMs), and the Internet. Common general characteristics of the discretionary user are as follows:

- Use of the computer or system is not absolutely necessary.
- Technical details are of no interest.
- Extra effort to use the system may not be invested.
- High motivation to use the system may not be exhibited.
- May be easily disenchanted.

https://hemanthrajhemu.github.io

- Voluntary use may have to be encouraged.
- Is from a heterogeneous culture.

For the business system discretionary user, the following may also be appropriate:

- Is a multifunction knowledge worker.
- The job can be performed without the system.
- May not have expected to use the system.
- Career path may not have prepared him or her for system use.

Quite simply, this discretionary user often judges a system on the basis of expected effort versus results to be gained. If the benefits are seen to exceed the effort, the system will be used. If the effort is expected to exceed the benefits, it will not be used. Just the *perception* of a great effort to achieve minimal results is often enough to completely discourage system use, leading to system rejection, a common discretionary reaction.

The discretionary user, or potential user, exhibits certain characteristics that vary. A study of users of ATMs identified five specific categories. Each group was about equal in size, encompassing about 20 percent of the general population. The groups, and their characteristics, are the following:

- People who understand technology and like it. They will use it under any and all circumstances.
- People who understand technology and like it, but will use it only if the benefits are clear.
- People who understand technology but do not like it. They will use it only if the benefits are overwhelming.
- People who do not understand anything technical. They might use it if it is very easy.
- People who will never use technology of any kind.

Again, clear and obvious benefits and ease of learning to use a system dominate these usage categories.

Frequency of Use

Is system use a continual, frequent, occasional, or once-in-a-lifetime experience? Frequency of use affects both learning and memory. People who spend a lot of time using a system are usually willing to spend more time learning how to use it in seeking efficiency of operation. They will also more easily remember how to do things. Occasional or infrequent users prefer ease of learning and remembering, often at the expense of operational efficiency.

Task or Need Importance

How important is the task or need for the user? People are usually willing to spend more time learning something if it makes the task being performed or need being fulfilled more efficient. For less important things, ease of learning and remembering are preferred, because extensive learning time and effort will not be tolerated.

https://hemanthrajhemu.github.io

Task Structure

How structured is the task being performed? Is it repetitive and predictable or not so? In general, the less structure, the more flexibility should exist in the interface. Highly structured tasks require highly structured interfaces.

Social Interactions

Will the user, in the normal course of task performance, be engaged in a conversation with another person, such as a customer, while using the system? If so, design should not interfere with the social interaction. Neither the user nor the person to whom the user is talking must be distracted in any way by computer interaction requirements. The design must accommodate the social interaction. User decision-making required by the interface should be minimized and clear eye-anchors built into the screen to facilitate eye movements by the user between the screen and the other person.

Primary Training

Will the system training be extensive and formal, will it be self-training from manuals, or will training be impossible? With less training, the requirement for system ease of use increases.

Turnover Rate

In a business system, is the turnover rate for the job high, moderate, or low? Jobs with high turnover rates would not be good candidates for systems requiring a great deal of training and learning. With low turnover rates, a greater training expense can be justified. With jobs possessing high turnover rates, it is always useful to determine why. Perhaps the new system can restructure monotonous jobs, creating more challenge and thereby reducing the turnover rate.

Job Category

In a business system, is the user an executive, manager, professional, secretary, or clerk? While job titles have no direct bearing on design per se, they do enable one to predict some job characteristics when little else is known about the user. For example, executives and managers are most often discretionary users, while clerks are most often mandatory ones. Secretaries usually have typing skills, and both secretaries and clerks usually have higher turnover rates than executives and managers.

Lifestyle

For Web e-commerce systems, user information to be collected includes hobbies, recreational pursuits, economic status, and other similar more personal information.

https://hemanthrajhemu.github.io

The User's Psychological Characteristics

A person's psychological characteristics also affect one's performance of tasks requiring motor, cognitive, or perceptual skills.

Attitude and Motivation

Is the user's attitude toward the system positive, neutral, or negative? Is motivation high, moderate, or low? While all these feelings are not caused by, and cannot be controlled by, the designer, a positive attitude and motivation allows the user to concentrate on the productivity qualities of the system. Poor feelings, however, can be addressed by designing a system to provide more power, challenge, and interest for the user, with the goal of increasing user satisfaction.

Web user attitudes and motivations have become a fertile research ground in the past several years. While much of the research is directed toward users as consumers, some of the findings provide interesting implications for design. For example:

- Consumer purchase behavior is driven by perceived security, privacy, quality of content and design, in that order (Ranganathan, C. and Ganapathy, S., 2002).
- Visual elements such as layout, use of color, and typography influence impression of site credibility (Fogg et al., 2002).
- Visual parameters such as font size, colors, and persistent navigation contribute to quality ratings of Web sites (Ivory and Hearst, 2002).
- Security and information quality and quantity are predictors of user-satisfaction in e-commerce (Lightner, 2003).
- Trust in the on-line purchase process is influenced by
  - Perceived creditability.
  - Ease of use.
  - Perceived degree of risk (Corritore, C.L., Krachcher, B., and Wiedenbeck, S., 2003).
- Including and highlighting design features that reduce negative attitudes about a site will increase usage (Jackson, 2003).
- Sensory impact influences younger users, whereas vendor reputation is a better predictor of satisfaction for older, more educated users (Lightner, 2003).

This glimpse into the attitudes and motivations of Web page users points out the value of good Web security, content, format, and usability. It also suggests that users of different ages will be influenced by different Web page attributes.

Patience

Is the user patient or impatient? Recent studies of the behavior of Web users indicate that they are becoming increasingly impatient. They are exhibiting less tolerance for Web-use learning requirements, slow response times, and inefficiencies in navigation and locating desired content. Recent research has found that people will wait longer for better content and experienced users won't wait as long as novices (Ryan and Valverde, 2003).

https://hemanthrajhemu.github.io

Stress Level

Will the user be subject to high levels of stress while using the system? Interacting with an angry boss, client, or customer, can greatly increase a person's stress level. High levels of stress can create confusion and cause one to forget things one normally would not. System navigation or screen content may have to be redesigned for extreme simplicity in situations that can become stressful.

Expectations

What are user's expectations about the system or Web site? Are they realistic? Is it important that the user's expectations be realized?

Cognitive Style

People differ in how they think about and solve problems. Some people are better at verbal thinking, working more effectively with words and equations. Others are better at spatial reasoning — manipulating symbols, pictures, and images. Some people are analytic thinkers, systematically analyzing the facets of a problem. Others are intuitive, relying on rules of thumb, hunches, and educated guesses. Some people are more concrete in their thinking, others more abstract. This is speculative, but the verbal, analytic, concrete thinker might prefer a textual style of interface. The spatial, intuitive, abstract thinker might feel more at home using a multimedia graphical interface.

The User's Physical Characteristics

The physical characteristics of people can also greatly affect their performance with a system.

Age

The invasiveness of the Web has greatly expanded the range of computer users. Computers are no longer the domains of the young and middle-aged only. Users now come in a wide range of ages from young to elderly. The Internet is quickly graying. AARP says that more than 40 million adults over 50 are online in the United States. The portion of people of a given age who use the Internet are (Weinschenk, 2006)

- 46–55 86%
- 56–65 75%
- 66+ 41%

Are the users children, young adults, middle-aged, senior citizens, or very elderly? Age can have a profound effect on computer, system, and Web usage. Older people may not have the manual dexterity to accurately operate many input devices. A double-click on a mouse, for example, is increasingly more difficult to perform as dexterity declines. With age, the eye's capability also deteriorates, affecting screen readability. Memory ability also diminishes. Recent research on the effects of age and system usability has found the following:

https://hemanthrajhemu.github.io

Younger adults (aged 18–36), in comparison to older adults (aged 64–81) (Mead et al., 1997; Piolat et al., 1998)

- Use computers and ATMs more often.
- Read faster.
- Possess greater reading comprehension and working memory capacity.
- Possess faster choice reaction times.
- Possess higher perceptual speed scores.
- Complete a search task at a higher success rate.
- Use significantly less moves (clicks) to complete a search task.
- Are more likely to read a screen a line at a time.

Older adults, in comparison to younger adults

- Are more educated.
- Possess higher vocabulary scores.
- Have more difficulty recalling previous moves and location of previously viewed information.
- Have more problems with tasks that require three or more moves (clicks).
- Are more likely to scroll a page at a time.
- Respond better to full pages rather than long continuous scrolled pages.

Age Classifications

While it is now well known that age has a profound effect on usability, the age research has blurred the lines when it comes to creating age categories for users. In searching for age-related deficiencies in performance, and when these deficiencies become evident, much inconsistency has existed in age groupings that have been studied. This has hindered the development of age-related guidelines. To address this problem Nichols et al. (2001) reviewed age classifications reported in a variety of studies. Combining these studies, they created the following age classifications:

- Young 19–35
- Middle-aged 40–59
- Older 58–82

Bailey (2002) has slightly modified these categories and included a fourth category now being used in the industry, "Oldest." These categories are described in Table 1.4. While differences may or may not exist between people falling in different age categories, age standardization will make comparisons between different studies possible.

Vision

Vision is a sense organ that begins to diminish in effectiveness at an early age, as anyone over 40 can attest. The eye begins its aging process in our early thirties; the amount of light able to pass through the retina begins to diminish. At 40 the process accelerates, and by age 50 most people need 50 percent more light to read by than they did when they were in their twenties. Failing to be able to read a menu in a dimly lit restaurant is often the first time we become aware of this problem.

https://hemanthrajhemu.github.io

Table 1.4: User Age Categories

Young	18–39
Middle-aged	40–59
Older	60–74
Oldest	75 and older

Also occurring is a reduced lens elasticity preventing focusing close to the eyes. The dreaded bifocal lens becomes a necessity. One's field of vision is also reduced, constricting the edges of what can be seen, and reduced retinal efficiency occurs hindering adapting to glare and changing light conditions. As a result of these changes, older adults read prose text in smaller type fonts more slowly than younger adults (Charness and Dijkstra, 1999).

Hearing

As people age, they require louder sounds to hear, a noticeable attribute in almost any everyday activity. Cohen (1994) determined the preferred levels for listening to speech at various age levels. These levels are summarized in Table 1.5.

Cognitive Processing

Brain processing also appears to slow with age. Working memory, attention capacity, and visual search appear to be degraded. Tasks where knowledge is important show the smallest age effect and tasks dependent upon speed show the largest effect (Sharit and Czaja, 1994).

Older users, a study found, also had more problems with Web searches that required three or more mouse clicks, and they searched less efficiently than younger users, requiring 81 percent more moves (Mead et al., 1997). Memory limitations seemed to be the cause of most of these problems. Older people also had a harder time adjusting to computer jargon and recovering from errors (Dulude, 2002).

Table 1.5: Hearing Comfort Levels by Age

AGE IN YEARS	SOUND LEVEL IN dB
15	54
25	57
35	61
45	65
55	69
65	74
75	79
85	85

https://hemanthrajhemu.github.io

Other age-related studies have compared people's performance with their time-of-day preferences (Intons-Peterson et al., 1998; Intons-Peterson et al., 1999). Older people were found to prefer to perform in the morning; younger people had no significant time of day preferences. In a memory test, younger users were able to perform well at all times in the day, older users, however, performed best during their preferred times.

The aforementioned research conclusions illustrate the kinds of differences age can play in making decisions.

Manual Dexterity

As people age, their manual dexterity diminishes. Typing and mouse movements become slower. Morris and Brown (1994) also found, in a task requiring speaking into a computer, that older users had an average speaking rate 14 percent slower than younger users.

Older People and Internet Use

Older people are now a significant force in Internet use. The Center for Digital Future at the USC Annenberg School says that the percentage of Internet use by older users is:

- Age 45–55 86%
- Age 56–65 75%
- Age 66 + 41%

Specific design guidelines for this large class of older users are discussed in Step 10.

Gender

A user's sex may have an impact on both motor and cognitive performance. Women are not as strong as men, so moving heavy displays or controls may be more difficult. Women also have smaller hands than men, so controls designed for the hand size of one may not be used as effectively by the other. Significantly more men are color-blind than women, so women may perform better on tasks and screens using color-coding. Tan et al. (2003) found that males significantly outperform females in navigational tasks.

Handedness

A user's handedness, left or right, can affect ease of use of an input mechanism, depending on whether it has been optimized for one or the other hand. Research shows that for adults

- 87 percent are right-handed.
- 13 percent are left-handed or can use both hands without a strong preference for either one.
- No gender or age differences exist.
- There is a strong cultural bias; in China and Japan only 1 percent of people are left-handed.

MAXIM **Ease of use promotes use.**

https://hemanthrajhemu.github.io

Disabilities

Disabilities such as blindness, defective vision, color-blindness, deafness, and motor handicaps can affect performance on a system not designed with these disabilities in mind. People with special needs must be considered in design. This is especially true for systems such as the Web that permit unlimited user access. Guidelines for people with disabilities are discussed in Step 10.

Human Interaction Speeds

Many researchers have studied the speed at which people can perform using various communication methods.. The following, as summarized by Bailey (2000), have been found to be typical interaction speeds for various tasks. These speeds are also summarized in Table 1.6.

Table 1.6: Average Human Interaction Speeds

READING	
Prose text:	250–300 words per minute.
Proofreading text on paper:	200 words per minute.
Proofreading text on a monitor:	180 words per minute.
Listening:	150–160 words per minute.
Speaking to a computer:	105 words per minute.
After recognition corrections:	25 words per minute.
KEYING: TYPEWRITER	
Fast typist:	150 words per minute and higher.
Average typist:	60–70 words per minute.
COMPUTER	
Transcription:	33 words per minute.
Composition:	19 words per minute.
TWO FINGER TYPISTS	
Memorized text:	37 words per minute.
Copying text:	27 words per minute.
HAND PRINTING	
Memorized text:	31 words per minute.
Copying text:	22 words per minute.

https://hemanthrajhemu.github.io

Reading. The average adult, reading English prose in the United States, has a reading speed in the order of 250 to 300 words per minute. Proofreading text on paper has been found to occur at about 200 words per minute, on a computer monitor, about 180 words per minute (Ziefle, 1998). Nontraditional reading methods have also been explored in research laboratories. One technique that has dramatically increased reading speeds is called Rapid Serial Visual Presentation, or RSVP. In this technique single words are presented one at a time in the center of a screen. New words continually replace old words at a rate set by the reader. Bailey (1999a) tested this technique with a sample of people whose paper document reading speed was 342 words per minute (with a speed range of 143 to 540 words per minute). Single words were presented on a screen in sets at a speed sequentially varying ranging from 600 to 1,600 words per minute. After each set a comprehension test was administered. For measured comprehension scores of 75 percent or higher, the average reading speed was 1,212 words per minute. This is about 3.5 times faster than reading in the traditional way. Bailey concludes that computer technology can help improve reading speeds, but nontraditional techniques must be used.

Listening. Words can be comfortably heard and understood at a rate of 150 to 160 words per minute. This is generally the recommended rate for audio books and video narration (Williams, 1998). Omoigui et al. (1999) did find, however, that when normal speech is speeded up using compression, a speed of 210 words per minute results in no loss of comprehension.

Speaking. Dictating to a computer occurs at a rate of about 105 words per minute (Karat et al., 1999; Lewis, 1999). Speech recognizer misrecognitions often occur, however, and when word correction times are factored in, the speed drops significantly to an average of 25 words per minute. Karat et al. (1999) also found that the speaking rate of new users was 14 words per minute during transcription and eight words per minute during composition.

Keying. Fast typewriter typists can key at rates of 150 words per minute and higher. Average typing speed is considered to be about 60 to 70 words per minute. Computer keying has been found to be much slower, however. Speed for simple transcription found by Karat et al. (1999) was only 33 words per minute and for composition only 19 words per minute. In this study, the fastest typists typed at only 40 words per minute, the slowest at 23 words per minute. Brown (1988) reports that two-finger typists can key memorized text at 37 words per minute and copied text at 27 words per minute. Something about the computer, its software, and the keyboard does seem to significantly degrade the keying process. (And two-finger typists are not really that bad off after all.)

Hand printing. People hand-print memorized text at about 31 words per minute. Text is copied at about 22 words per minute (Brown, 1988).

Performance versus Preference

Occasionally, when asked, people may prefer an interface design feature that actually yields poorer performance than another feature. Numerous instances of performance/preference differences have been reported in the literature (Andre and

https://hemanthrajhemu.github.io

Wickens, 1995). Examples include pointing with a mouse or cursor, alternative menu interaction techniques, use of color, two-dimensional versus three-dimensional displays, and prototype fidelity.

Preferences are influenced by many things including familiarity, aesthetics, novelty, and perceived effort in feature use. Rarely are people aware of the many human mechanisms responsible for the speed and accuracy of human-computer interaction. Ideally, in design, always augment preferences with performance measures and try to achieve an optimized solution. Where optimization is impossible, however, implement the feature that provides the best performance, and, very importantly, explain to the user why this is being done. In stating preferences, the user may not always be right.

Methods for Gaining an Understanding of Users

Gould (1988) suggests using the following kinds of techniques to gain an understanding of users, their tasks and needs, the organization where they work, and the environment where the system may be used:

- Visit user locations, particularly if they are unfamiliar to you, to gain an understanding of the user's work environment.
- Talk with users about their problems, difficulties, wishes, and what works well now.
- Establish direct contact; avoid relying on intermediaries.
- Observe users working or performing a task to see what they do, their difficulties, and their problems.
- Videotape users working or performing a task to illustrate and study problems and difficulties.
- Learn about the work organization where the system may be installed.
- Have users think aloud as they do something to uncover details that may not otherwise be solicited.
- Try the job yourself. It may expose difficulties that are not known or expressed by users.
- Prepare surveys and questionnaires to obtain a larger sample of user opinions.
- Establish testable behavioral target goals to give management a measure for what progress has been made and what is still required.

These techniques are discussed in more detail in Step 2.

In conclusion, this chapter addressed *the* most important principle in interface and screen design. Simply stated, it is this: *Know your user, client, or customer.*

Step 1 Exercise

An exercise for Step 1 can be found on this book's companion Web site, www.wiley .com/college/galitz.

https://hemanthrajhemu.github.io

STEP

2

Understand the Business Function

A thorough understanding of the user has been obtained, and the focus now shifts to the business function being addressed. Requirements must be determined and user activities being performed must be described through task analysis. From these, a conceptual model of the system will be formulated. Design standards must also be created (if not already available), usability goals established, and training and documentation needs determined.

A detailed discussion of all of these topics is beyond the scope of this book. The reader in need of more detail is referred to books exclusively addressing systems analysis, task analysis, usability, training, and documentation. The general steps to be performed are the following:

- Perform a business definition and requirements analysis.
- Determine basic business functions.
- Describe current activities through task analysis.
- Develop a conceptual model of the system.
- Establish design standards or style guides.
- Establish system usability design goals.
- Define training and documentation needs.

https://hemanthrajhemu.github.io

Business Definition and Requirements Analysis

The objective of this phase is to establish the need for a system. A requirement is an objective that must be met. A product description is developed and refined, based on input from users, marketing, or other interested parties.

Information Collection Techniques

There are many techniques for capturing information for determining requirements. Keil and Carmel (1995), Popowicz (1995), and Fuccella et al. (1999) described many of the following methods. They have also provided insights into their advantages and disadvantages. The techniques listed are classified as direct and indirect. Direct methods consist of face-to-face meetings with, or actual viewing of, users to solicit requirements. Indirect methods impose an intermediary, someone or something, between the users and the developers.

Before beginning the analysis, the developer should be aware of the policies and work culture of the organization being studied. He or she should also be familiar with any current system or process the new system is intended to supplement or replace.

Direct Methods

The significant advantage of the direct methods is the opportunity they provide to hear the user's comments in person and firsthand. Person-to-person encounters permit multiple channels of communication (body language, voice inflections, and so on) and provide the opportunity to immediately follow up on vague or incomplete data. Here are some recommended direct methods for getting input from users.

Individual Face-to-Face Interview

A one-on-one visit is held with the user. Information can be collected in a friendly and fast way. It may be structured or more open-ended. Generally, structured interviews are easier for the interviewer. Open-ended interviews are harder to conduct but provide a greater opportunity to detect and follow up on relevant issues. Data analysis is more difficult, however, with an open-ended interview.

The interview must have focus and topics to be covered must be carefully planned so data is collected in a common framework, and to ensure that all-important aspects are thoroughly covered. A formal questionnaire should not be used, however. Useful topics to ask the user to describe in an interview include the following:

- The activities performed in completing a task or achieving a goal or objective.
- The methods used to perform an activity.
- What interactions exist with other people or systems.

It is also very useful to uncover any of the following:

- Potential measures of system usability.
- Unmentioned exceptions to standard policies or procedures.
- Relevant knowledge the user must possess to perform the activity.

https://hemanthrajhemu.github.io

If designing a Web site, the following kinds of interview questions are appropriate for asking potential users:

- Present a site outline or proposal and then solicit comments on the thoroughness of content coverage, and suggestions for additional content.
- Ask users to describe situations in which the proposed Web site might be useful.
- Ask users to describe what they like and dislike about the Web sites of potential competitors.
- Ask users to describe how particular Web site tasks should be accomplished.

Ask users to describe the ideal version of a product. Breivik and Supphellen (2002) found that asking participants to generate rank-ordered lists of product attributes yielded widely varying results. If, instead, people are asked to describe the ideal version of the product, fewer attributes will be presented but they are ones that are important — attributes that people use to evaluate a product.

Advantages of a personal interview are that you can give the user your full attention, can easily include follow-up questions to gain additional information, will have more time to discuss topics in detail, and will derive a deeper understanding of your users, their experiences, attitudes, beliefs, and desires. If you conduct an interview at the work site, you can see the user's technology and environment. Disadvantages of interviews are that they can be costly and time-consuming to conduct, and someone skilled in interviewing techniques should perform them. The interviewer must establish a positive relationship with the user, ask questions in a neutral manner, be a good listener, and know when and how to probe for more information. Time must also be allowed for free conversation in interviews. Recording the session for playback to the entire design team provides all involved with some insights into user needs.

Telephone Interview or Survey

This interview is conducted using the telephone. It must have structure and be well planned. Arranging the interview in advance allows the user to prepare for it. Telephone interviews are less expensive and less invasive than personal interviews. They can be used much more frequently and are extremely effective for very specific information. Telephone interviews have some disadvantages. It is impossible to gather contextual information such as a description of the working environment, replies may be easily influenced by the interviewer's comments, and body language cues are missing. Also, it may be difficult to contact the right person for the telephone interview.

MAXIM **Know thy users, for they are not you.**

Traditional Focus Group

A small group of users (8 to 12) and a moderator are brought together to discuss the requirements. While the discussion is loosely structured, the range of topics must be determined beforehand. A typical session lasts about two hours. The purpose of a focus group is to probe users' experiences, attitudes, beliefs, and desires, and to obtain their reactions to ideas or prototypes. Focus groups are not usually useful for establishing how users really work or what kinds of usability problems they really have.

https://hemanthrajhemu.github.io

What users think or say they will do in focus groups and what they actually do in usability tests often differs (Eysenbach and Kohler, 2002). Focus group discussion can be influenced by group dynamics, for good or bad. Recording of the session, either video or audio, permits later detailed analysis of participants' comments. Again, the recording can also be played for the entire design team, providing insights into user needs for all developers. Setting up a focus group involves the following:

- Establish the objectives of the session.
- Select participants representing typical users or potential users.
- Write a script for the moderator to follow.
- Find a skilled moderator to facilitate discussion, to ensure that the discussion remains focused on relevant topics, and to ensure that everyone participates.
- Allow the moderator flexibility in using the script.
- Take good notes, using the session recording for backup and clarification.

Facilitated Team Workshop

A facilitated team workshop is similar in structure and content to a traditional focus group but is slightly less formal. A common technique used in system requirements determination for many years, it is now being replaced (at least in name) by focus groups. Team workshops have had the potential to provide much useful information. Like focus groups, they do require a great deal of time to organize and run.

Observational Field Study

To see and learn what users actually do, they are watched and followed in their own environment, office, or home in a range of contexts for a period of time. Observation provides good insight into tasks being performed, the working environment and conditions, the social environment, and working practices. It is more objective, natural, and realistic. Observation, however, can be time-consuming and expensive. A limitation is the inability of the evaluator to obtain a full record of the user's activities in one session. When taking notes, the evaluator must quickly decide what to record, and important aspects may be missed. Also, direct observation is considered to be intrusive and can change a user's performance and behavior.

Video recording of the observation sessions provide a permanent record and permit a later detailed task analysis. Because a large amount of data is collected, its analysis can be even more time-consuming. Playing the recording for the entire design team again provides all involved with some insights into user tasks.

Requirements Prototyping

A demonstration model, or very early prototype, is presented to users for their comments concerning functionality and to clarify requirements. Prototypes are discussed in more detail in Step 14.

User-Interface Prototyping

A demonstration model, or early prototype, is presented to users to uncover user-interface issues and problems. Again, prototypes are discussed in more detail in Step 14.

https://hemanthrajhemu.github.io

Usability Laboratory Testing

A special laboratory is constructed and users are brought in to perform actual newly designed tasks. They are observed, and the results are measured and evaluated to establish the usability of the product at that point in time. Usability tests uncover what people actually do, not what they think they do, which is a common problem with verbal descriptions. The same scenarios can be presented to multiple users, providing comparative data from several users. Problems uncovered may result in modification of the requirements. Usability labs can generate much useful information but are expensive to create and operate.

Laboratory tests can also be held in an office, conference room, or a hotel meeting room. To collect test data, laboratories are now available in portable units that can be easily shipped and set up at remote facilities. Portable labs now possess most of the components incorporated within specially constructed labs and are less expensive to conduct. Usability lab testing is discussed in more detail in Step 14.

Card Sorting for Web Sites

This is a technique used to establish hierarchical groupings and the information architecture for Web sites. It is normally used only after gathering substantial potential site content information using other analysis techniques. Potential content topics are placed on individual index cards and users are asked to sort the cards into groupings that are meaningful to them. Card sorting assists in building the site's structure, map, and page content. Briefly, the process is as follows:

- Identify about 50 content topics from previous analyses and inscribe them on index cards. Limit topics to no more than 100.
- Provide blank index cards for names of additional topics participants may want to add, and colored blank cards for groupings that participants will be asked to create.
- Number the cards on the back.
- Arrange for a facility with a large enough table for spreading out cards.
- Select participants representing a range of users. Use one or two people at a time and 15 to 20 in total.
- Explain the process to the participants, saying that you are trying to determine what categories of information will be useful, what groupings make sense, and what the groupings should be called.
- Ask the participants to sort the cards and talk out loud while doing so. Advise the participants that additional content cards may be named and added as they think necessary during the sorting process.
- Observe and take notes as the participants talk about what they are doing. Pay particular attention to the sorting rationale.
- Upon finishing the sorting, if a participant has too many groupings, ask that they be arranged hierarchically.
- Ask participants to provide a name for each grouping on the colored blank cards, using words that the user would expect to see that would lead them to that particular grouping.

https://hemanthrajhemu.github.io

- Make a record of the groupings using the numbers on the back of each card.
- Reshuffle the cards for the next session.

When the session is finished, you can analyze the results and look for commonalities among the different sorting sessions.

The recommended number of participants, 15 to 20, is based upon a research study by Tullis and Wood (2004). They had 168 people perform a card-sorting task. Then, they analyzed statistically the results of smaller groups of users. They found the following correlations between the sorts for the entire group and the sorts for the smaller groups: 15 people, .090; 30 people, 0.95; 60 people, 0.98. After 15 people, the law of diminishing returns set in. Successive larger numbers increased the correlations very little.

The sorting can also be accomplished on the Web. The National Institute of Standards and Technology (NIST, 2001) has developed a card-sorting tool. The designer sets up the cards and names the categories. The user then sorts by dragging and dropping. Sorting can also be done with other Web-based tools.

Indirect Methods

An indirect method of requirements determination is one that places an intermediary between the developer and the user. This intermediary may be electronic or another person. Using an intermediary can certainly provide useful information. Working through an intermediary, however, takes away the multichannel communication advantages of face-to-face user-developer contact. Some electronic intermediaries do provide some advantages, as will be described shortly. Imposition of a human intermediary can create additional problems. First, there may be a filtering or distortion of the message, either intentional or unintentional. Next, the intermediary may not possess a complete or current understanding of the user's needs, passing on an incomplete or incorrect message. Finally, the intermediary may be a mechanism that discourages direct user-developer contact for political reasons. Indirect methods include the following:

MIS Intermediary

A company representative who defines the user's goals and needs to designers and developers fulfills this intermediary role. This representative may come from the Management Information Services department, or he or she may be from the using department. While much useful information can be provided, all too often this person does not have the breadth of knowledge needed to satisfy all design requirements.

Paper Survey or Questionnaire

A paper questionnaire or survey is administered to a sample of users to obtain their needs. Questionnaires have the potential to be used for a large target audience located almost anywhere, and are much cheaper than customer visits. However, they generally have a low return rate, often generating responses only from those "very happy" or "very unhappy." They may take a long time to collect and may be difficult to analyze. Questionnaires are useful for determining users' attitudes, experiences and desires,

https://hemanthrajhemu.github.io

but not for determining actual tasks and behaviors. Questionnaires should be composed mostly of closed questions (yes/no, multiple choice, short answer, and so on). Open-ended questions require much more analysis. Questionnaires should be relatively short and created by someone experienced in their design.

Electronic Survey or Questionnaire

A questionnaire or survey is administered to a sample of users via e-mail or the Web. Characteristics, advantages, and disadvantages are similar to paper surveys and questionnaires. They are, however, significantly less expensive than mailed surveys. The speed of their return can also be much faster than those distributed in a paper format. In creating an electronic survey

- Determine the survey objectives.
- Determine where you will find the people to complete the survey.
- Create a mix of multiple choice and open-ended questions requiring short answers addressing the survey objectives.
- Keep it short — about 10 items or less is preferable.
- Keep it simple, requiring no more than 5 to 10 minutes to complete.
- Run a pilot test to ensure that the questions
  - Are clear and unambiguous.
  - Will gather the needed information.

Also consider a follow-up, more detailed survey, or surveys, called iterative surveys. Ask people who complete and return the initial survey if they are willing to answer more detailed questions. If so, create and send the more detailed survey. Among other things, the detailed survey content can address questions the initial survey raises. A useful follow-up survey goal is to ask the participant to prioritize their needs and to rank expected user tasks according to their importance. A third follow-up survey can also be designed to gather additional information about the most important requirements and tasks. Iterative surveys, of course, take a longer time to complete. Don't forget to thank participants for their help and time.

Electronic Focus Group

An electronic focus group is similar to a traditional focus group except that the discussion is accomplished electronically using specialized software on a workstation, e-mail, or a Web site. As with the direct methods, the opportunity to immediately follow up on vague or incomplete data exists. All comments, ideas, and suggestions are available in hard-copy form for easier analysis. Specialized software can provide ratings or rankings of items presented in lists, a task requiring much more effort in a traditional focus group.

Other advantages of electronic focus groups over traditional focus groups are that the discussion is less influenced by group dynamics; has a smaller chance of being dominated by one or a few participants; can be anonymous, leading to more honest comments and less caution in proposing new ideas; can generate more ideas in a shorter time because all participants can communicate at once; and can lead to longer sessions because the participant is in a more comfortable "home environment" and not

https://hemanthrajhemu.github.io

confined to a conference room. Among the disadvantages are that the depth and richness of verbal discussions does not exist and the communication enhancement aspects of seeing participant's body language are missing.

Marketing and Sales

Company representatives who regularly meet customers obtain suggestions or needs, current and potential. This information is collected inexpensively, because the representative is going to visit the company anyway. Business representatives do have knowledge of the nature of customers, the business, and the needs that have to be met. Some dangers: The information may be collected from the wrong people, the representative may unintentionally bias questions, there may be many company "filters" between the representative's contact and the end user, and quantities may sometimes be exaggerated. ("Lots of people are complaining about . . ." may mean only one or two.) The developers should know the interests and bias of the representatives collecting the information.

Support Line

Information is collected by the unit (Customer Support, Technical Support, Help Desk, and so on) that helps customers with day-to-day problems. This is fairly inexpensive and the target user audience is correct. The focus of this method is usually only on problems, however.

E-Mail, Bulletin Boards, or Guest Book

Problems, questions, and suggestions by users posted to a bulletin board, a guest book, or through e-mail are gathered and evaluated. Again, the focus of this method is usually only on problems. The responsibility is on the user to generate the recommendations, but this population often includes unhappy users. This is a fairly inexpensive method.

User Group

Improvements suggested by customer groups who convene periodically to discuss system and software usage are evaluated. User groups have the potential to provide a lot of good information, if organized properly. They require careful planning, however.

Competitor Analysis

Reviews of competitor's products, or Web sites, can also be used to gather ideas, uncover design requirements, and identify tasks. The designers can perform this evaluation or, even better, users can be asked to perform the evaluation.

Trade Show

Customers at a trade show can be exposed to a mock-up or prototype and asked for comments. This method is dependent on the knowledge level of the customers and may provide only a superficial view of most prominent features.

https://hemanthrajhemu.github.io

Other Media Analysis

Analyze how other media, print or broadcast, present the process, information, or subject matter of interest. Findings can be used to gather ideas, uncover design requirements, and identify better ways to accomplish or show something.

System Testing

New requirements and feedback stemming from ongoing system testing can be accumulated, evaluated, and implemented as necessary.

Requirements Collection Guidelines

- Establish four to six different developer-user links.
- Provide most reliance on direct links.

Keil and Carmel (1995) evaluated the suitability and effectiveness of various requirements-gathering methods by collecting data on 28 projects in 17 different companies. Fourteen of the projects were rated as relatively successful, 14 as relatively unsuccessful. Each requirements collection method was defined as a developer-user link. Their findings and conclusions are as follows:

Establish 4 to 6 Different Developer-User Links

The more successful projects had utilized a greater number of developer-user links than the less successful projects. The mean number of links for the successful projects: 5.4; the less successful: 3.2. This difference was statistically significant. Few projects used more than 60 percent of all possible links. Keil and Carmel recommend, based upon their data, that, at minimum, four different developer-user links must be established in the requirements-gathering process. They also concluded that the law of diminishing returns begins to set in after six links.

Effectiveness ratings of the most commonly used links in their study were also obtained. (Not all the previously-described techniques were considered in their study.) On a 1 to 5 scale (1 = ineffective, 5 = very effective) the following methods had the highest ratings:

Custom projects (software developed for internal use and usually not for sale):

Facilitated Teams	5.0
User-Interface Prototype	4.0
Requirements Prototype	3.6
Interviews	3.5

Package projects (software developed for external use and usually for sale):

Support Line	4.3
Interviews	3.8
User-Interface Prototype	3.3
User Group	3.3

https://hemanthrajhemu.github.io

Provide the Most Reliance on Direct Links

The problems associated with the less successful projects resulted, at least in part, from too much reliance on indirect links, or using intermediaries. Ten of the 14 less successful projects had used none or only one direct link. The methods with the highest effectiveness ratings listed previously were mostly direct links.

Keil and Carmel caution that number of links is only a partial measure of user participation. How well the link is employed in practice is also very important.

Defining the Domain

The domain is the area of expertise and specialist knowledge for which a system is being developed. The domain provides the system's underlying concepts. For example, an airline system requires reservations, flight scheduling, crew scheduling, and so forth. A court services system might require functions such as pre-trial monitoring, probation, parole, and drug testing. Specialized knowledge to perform the tasks and accomplish goals must be understood and defined.

MAXIM What people say they do and what they actually do are often different — because people are not always aware of how they work.

Requirements information can be partially derived from reviewing documentation from the old manual process or the current computer system. Domain experts must also be consulted using some of the previously described techniques. Keep in mind that domain experts may be unable to explain what they do, and what knowledge they possess, because it is so ingrained they are not consciously aware of its existence. In the course of interviews with domain experts, unsolicited information about users, their tasks, and the work environment may also be collected. This can be added to the body of requirements knowledge being accumulated in other ways and from other people.

Considering the Environment

Where the work or task is performed is also important in establishing requirements. Important environmental considerations are physical, safety, social, organizational, and user support (Stone et al., 2005).

The *physical* environment and conditions address issues like lighting, temperature, noise, and cleanliness. Cold, noise, and dust can impact design decisions concerning type of controls to use and the potential for using voice technology. Workspace layout is important as well. Are users in individual offices or tightly cramped together?

The *safety* environment addresses any existing hazards or health and safety issues. Is any special clothing needed? Will stress levels be high? Are there any pollution or other poor environmental conditions?

The *social* environment addresses the relationships between people and how they interact or do not interact. Do people share tasks and/or computers? Do people help each other or distract one another? Do people cooperate with one another or work alone? Is there a social hierarchy inhibiting communication?

https://hemanthrajhemu.github.io

The *organizational* environment addresses how a system will be integrated within the existing networks of people and technology. Does management support the work? What is the mission or purpose of the work? What are the working hours, potential for interruptions, work practices, cultural factors, and so on? Is performance monitored or is work paced? Is the user given autonomy and discretion?

The *user support* environment involves the availability of documentation and training materials, and colleagues to provide help when necessary. It also includes the availability of assistive devices to the user when necessary.

Each of these environmental issues will affect decisions and choices made in the design of the interface.

Possible Problems in Requirements Collection

Like other aspects of the design process, problems may occur in the requirements determination phase. Stone et al. (2005) says the following situations can hinder collection of the proper information:

- **Not enough user, customer, and other interested party involvement in the process.** The result may be incomplete requirements.
- **Lack of requirements management or coordination.** Requirement modifications are not recorded, tracked properly, or carried out. Requirements may be inaccurate or incomplete.
- **Communication problems among all participants.** It is possible that not all participants understand the exact requirements.
- **Capturing the relevant information may be difficult.** The relevant knowledge may exist in many places and locations, including books, operating manuals, and in people's heads.
- **People who do understand the problem may be constrained.** Heavy workloads and a lack of time may make people reluctant to, or unable to, participate.
- **Organizational and political factors and agendas may influence the process.** The resulting views may not tally with the users views. Resistance to new ideas and change may be reflected in comments made.
- **Disparities in knowledge may exist.** Some people may know what they want only in general terms, whereas others may be forceful and detailed about their wants. Getting a balanced view may be difficult.
- **Changing economic and business environments and personnel roles.** As a result, want and needs change as well.

In many cases there may be no way to prevent these kinds of problems from occurring. Awareness of these issues, however, may enable planning for these contingencies to be built into the requirements-gathering process.

Determining Basic Business Functions

A detailed description of what the product will do is prepared. Major system functions are listed and described, including critical system inputs and outputs. A flowchart of

https://hemanthrajhemu.github.io

major functions is developed. The process the developer will use is summarized as follows:

- Gain a complete understanding of the user's mental model based upon
  - The user's needs and the user's profile.
  - A user task analysis.
- Develop a conceptual model of the system based upon the user's mental model. This includes
  - Defining objects.
  - Developing metaphors.

The user interface activities described in Steps 1 and 3 are usually performed concurrently with these steps.

Understanding the User's Work

The next phase in interface design is to thoroughly describe the aims and goals of people who will be using the system. Also to be determined is what the system will be used for so that the necessary system functionality can be provided to permit people to achieve their goals. The technique used to gain an understanding of what the computer system must do is called *task analysis*. Another object of task analysis is to gain a picture of the user's *mental model*.

Mental Models

A mental model is an internal representation of a person's current conceptualization and understanding of something: themselves, other people, the environment, and the thing with which they interact. Mental models are gradually developed through experience, training, and instruction. They enable a person to understand, explain, and do something. Mental models enable a person to predict the actions necessary to do things if the actions have been forgotten or have not yet been encountered.

Performing a Task Analysis

User activities, the way in which people perform tasks, are precisely described in a task analysis. Task analysis involves breaking down the user's activities to the individual task level. The goal is to obtain an understanding of why and how people currently do the things that will be automated. Knowing why establishes the major work goals; knowing how provides details of actions performed to accomplish these goals. Task analysis also provides information concerning workflows; the interrelationships between people, objects, and actions; and the user's conceptual frameworks. The output of a task analysis is a complete description of all user tasks and interactions.

Work activities are studied and/or described by users using the techniques just reviewed: direct observation, interviews, questionnaires, or obtaining measurements of actual current system usage. Measurements, for example, may be obtained for the frequency with which tasks are performed or the number of errors that are made.

https://hemanthrajhemu.github.io

One result of a task analysis is a description of the user's current tasks, called a *scenario*. Scenarios are narrative descriptions of what people do in the course of completing a task. They can be long or short, abstract or detailed. They should, however, be detailed enough so that designers can infer, and reason about, the implications the activities and their flow have on interface design. Unlike prototypes, scenarios can be created before the system is. Their accuracy can be verified by testing them with users. They can be used throughout the conceptual design process to guide and evaluate iteratively the completeness of the requirements. Scenarios can also be refined throughout the conceptual design process, becoming more detailed as requirement gathering proceeds. Straub (2004a) in a review of several descriptive studies, suggests scenarios play a critical unifying role at several levels in the early stages of conceptual design:

- They bring a level of coherence to requirements by providing a "real world" task level description of the motivation and events that trigger tasks and workflow as users navigate to task completion.
- They provide a design-neutral bridge between developers working on different modules of the interface, thereby maintaining a holistic view of the design process.
- They provide a meaningful and accessible common ground for communicating and conveying the minds and needs of the users to the system models that the developers create.

Scenarios should be well documented and maintained. Changes in task requirements can then be easily incorporated as design iteration occurs. Another result is a list of objects the users see as important to what they do. The objects can be sorted into the following categories:

- Concrete objects — things that can be touched.
- People who are the object of sentences — normally organization employees (customers, for example).
- Forms or journals — things that keep track of information.
- People who are the subject of sentences — normally the users of a system.
- Abstract objects — anything not included above.

Developing Conceptual Models

The output of the task analysis is the creation, by the designer, of a conceptual model for the user interface. A conceptual model is the general conceptual framework through which the system's functions are presented. Such a model describes how the interface will present objects, the relationships between objects, the properties of objects, and the actions that will be performed. A conceptual model is based on the user's mental model. Because the term mental model refers to a person's current level of knowledge about something, people will always have them. Because mental models are influenced by a person's experiences, and people have different experiences, no two user mental models are likely to be exactly the same. Each person looks at the interface from a slightly different perspective.

https://hemanthrajhemu.github.io

The goal of the designer is to facilitate for the user the development of a useful mental model of the system. This is accomplished by presenting to the user a meaningful conceptual model of the system. When the user then encounters the system, his or her existing mental model will, hopefully, mesh well with the system's conceptual model. As a person works with a system, he or she then develops a mental model of the system. The system mental model the user derives is based upon the system's behavior, including factors such as the system inputs, actions, outputs (including screens and messages), and its feedback and guidance characteristics, all of which are components of the conceptual model. Documentation and training also play a formative role. Mental models will be developed regardless of the particular design of a system, and then they will be modified with experience. What must be avoided in design is creating for the user a conceptual model that leads to the creation of a false mental model of the system, or that inhibits the user from creating a meaningful or efficient mental model.

Guidelines for Designing Conceptual Models

- Reflect the user's mental model, not the designer's.
- Draw physical analogies or present metaphors.
- Comply with expectancies, habits, routines, and stereotypes.
- Provide action-response compatibility.
- Make invisible parts and processes of a system visible.
- Provide proper and correct feedback.
- Avoid anything unnecessary or irrelevant.
- Provide design consistency.
- Provide documentation and a help system that will reinforce the conceptual model.
- Promote the development of both novice and expert mental models.

Unfortunately, little research is available to assist the software designer in creating conceptual models. Development of a user's mental model can be aided, however, by following these general guidelines for conceptual model development:

Reflect the user's mental model, not the designer's. A user will have different expectations and levels of knowledge than the designer. So, the mental models of the user and designer will be different. The user is concerned with the task to be performed, and the business objectives that must be fulfilled. The designer's model is focused on the design of the interface, the kinds of objects, the interaction methods, and the visual representations on the screen. Objects must be defined, along with their relationships, behaviors, and properties. Interaction methods must also be defined, such as input mechanisms, interaction techniques, and the contents of menus. Visual screen representations must also be created, including functionality and appearance.

Draw physical analogies or present metaphors. Replicate what is familiar and well known. Duplicate actions that are already well learned. The success of graphical

https://hemanthrajhemu.github.io

systems can be attributed, in part, to their employing the desktop metaphor. A metaphor, to be effective, must be widely applicable within an interface. Metaphors that are only partially or occasionally applicable should not be used. In the event that a metaphor cannot be explicitly employed in a new interface, structure the new interface in terms of familiar aspects from the manual world.

Comply with expectancies, habits, routines, and stereotypes. Create a system that builds on knowledge, habits, routines, and expectancies that already exist. Use familiar associations, avoiding the new and unfamiliar. With color, for example, accepted meanings for red, yellow, and green are already well established. Use words and symbols in their customary ways. Replicate the language of the user, and create icons reflecting already known images.

Provide action-response compatibility. All system responses should be compatible with the actions that elicit them. Names of commands, for example, should reflect the actions that will occur. The organization of keys in documentation or help screens should reflect the ordering that actually exists on the keyboard.

Make invisible parts of the system visible. Systems are composed of parts and processes, many of which are invisible to the user. In creating a mental model, a person must make a hypothesis about what is invisible and how it relates to what is visible. New users of a system often make erroneous or incomplete assumptions about what is invisible and develop a faulty mental model. As more experience is gained, their mental models evolve to become more accurate and complete. Making invisible parts of a system visible will speed up the process of developing correct mental models. An example of a process being made visible can be illustrated by moving a document between files. In a command language interface, the document must be moved through a series of typed commands. The file is moved invisibly, and the user assumes correctly, unless an error message is received. In a graphical direct-manipulation system, the entire process is visible, with the user literally picking up the file in one folder by clicking on it, and dragging it to another folder.

Provide Proper and Correct Feedback. Be generous in providing feedback. Keep a person informed of what is happening, and what has happened, at all times, including the following:

- *Provide a continuous indication of status.* Mental models are difficult to develop if things happen, or are completed, unknown to the user. During long processing sequences, for example, interim status messages such as "loading . . .," "opening . . .," or "searching . . ." reassure the user and enable him or her to understand internal processes and more accurately predict how long something will take. Such messages also permit the pinpointing of problems if they occur.
- *Provide visible results of actions.* For example, highlight selected objects, display new locations of moved objects, and show files that are closed.
- *Display actions in progress*. For example, show a window that is being changed in size actually changing, not simply the window in its changed form. This will strengthen cause-and-effect relationships in the mental model.

https://hemanthrajhemu.github.io

- *Present as much context information as possible*. To promote contextual understanding, present as much background or historical information as possible. For example, on a menu screen or in navigation, maintain a listing of the choices selected to get to the current point. On a query or search screen, show the query or search criteria when displaying the results.
- *Provide clear, constructive, and correct error messages*. Incomplete or misleading error messages may cause false assumptions that violate and weaken the user's mental model. Error messages should always be structured to reinforce the mental model. For example, error messages addressing an incomplete action should specify exactly what is missing, not simply advise a person that something is incomplete.

Avoid the unnecessary or irrelevant. Never display irrelevant information on the screen. People may try to interpret it and integrate it into their mental models, thereby creating a false one. Irrelevant information might be unneeded data fields or information, screen controls, system status codes, or error message numbers. If potentially misleading information cannot be avoided, point this out to the user. Also, do not overuse display techniques or use them in meaningless ways. Too much color, for example, may distract people and cause them to make erroneous assumptions as they try to interpret the colors. The result will be a faulty and unclear mental model.

Provide design consistency. Design consistency reduces the number of concepts to be learned. Inconsistency requires the mastery of multiple models. If an occasional inconsistency cannot be avoided, explain it to the user. For example, if an error is caused by a user action that is inconsistent with other similar actions, explain in the error message that this condition exists. This will prevent the user from falsely assuming that the model he or she has been operating under is incorrect.

Provide documentation and a help system that will reinforce the conceptual model. Consistencies and metaphors should be explicitly described in the user documentation. This will assist a person in learning the system. Do not rely on people to uncover consistencies and metaphors themselves. The help system should offer advice aimed at improving mental models.

Promote the development of both novice and expert mental models. Novices and experts are likely to bring to bear different mental models when using a system. It will be easier for novices to form an initial system mental model if they are protected from the full complexity of a system. Employ levels of functionality that can be revealed through progressive disclosure.

Defining Objects

- Determine all objects that have to be manipulated to get work done. Describe
  - The objects used in tasks.
  - Object behavior and characteristics that differentiate each kind of object.
  - The relationship of objects to each other and the people using them.
  - The actions performed.

https://hemanthrajhemu.github.io

- The objects to which actions apply.
- Information or attributes that each object in the task must preserve, display, or allow to be edited.

- Identify the objects and actions that appear most often in the workflow.
- Make the several most important objects very obvious and easy to manipulate.

All objects that have to be manipulated to get work done must be clearly described. Their behavioral characteristics must be established and the attributes that differentiate each kind of object must be identified. Relationships of objects to each other and the people using them must be determined. The actions people take on objects must also be described. Information or attributes that each object in the task must preserve, display, or allow to be edited must be defined.

The most important objects must be made very obvious and easy to manipulate. Weinschenk (1995) suggests that if the most important objects are not obvious in the workflow, go through the workflow document highlighting all nouns and verbs associated with nouns. Frequently appearing nouns are possible major objects. Frequently appearing verbs are actions pointing to possible major objects.

Developing Metaphors

- Choose the analogy that works best for each object and its actions.
- Use real-world metaphors.
- Use simple metaphors.
- Use common metaphors.
- Multiple metaphors may coexist.
- Use major metaphors, even if you can't exactly replicate them visually.
- Test the selected metaphors.

A metaphor is a concept where one's body of knowledge about one thing is used to understand something else. Metaphors act as building blocks of a system, aiding understanding of how a system works and is organized. Select a metaphor or analogy for the defined objects. Choose the analogy that works best for the objects and their actions. Real-world metaphors are most often the best choice. Replicate what is familiar and well known. Duplicate actions that are already well learned. If a faster or better way exists to do something, however, use it. Use simple metaphors, as they are almost always the most powerful. Use common metaphors; uniqueness adds complexity. Multiple metaphors may coexist. Use major metaphors even if you can't exactly replicate them visually on the screen. Exactly mimicking the real world does not always aid understanding. It can lead a person to expect behavioral limitations that do not actually exist. A representation will be satisfactory. Finally, test the selected metaphors. Do they match one's expectations and experiences? Are they easily understood or quickly learned? Change them, if testing deems it necessary.

A common metaphor in a graphical system is the desktop and its components, items such as folders and a trash can. The Web utilizes a library metaphor for the activities of browsing and searching. Browsing in a library occurs when you wander around book

https://hemanthrajhemu.github.io

stacks looking for something interesting to read. When searching, you devise an active plan to find some specific information. For example, first, check the topic in the card catalog. Next, ask the librarian, and so forth.

MAXIM Given a choice between functionally equivalent designs, the simplest design should be selected.

A word of caution in creating metaphors, however. Today's technology permits doing a lot of things, many not even thinkable in the old manual world (or even the old computer world). Do not be constrained from developing a more powerful interface because a current metaphor just happens to exist. If you do limit yourself, you may find yourself in the position of the farm tractor designers of the early last century. In developing a new tractor, the metaphor was the horse and plow. Reins controlled the horse, so reins were installed on the tractor for controlling it as well. Needless to say, it was not successful. We do not want to read about your decision sometime later this century.

The User's New Mental Model

When the system is implemented, and a person interacts with the new system and its interface, an attempt will be made by the person to understand the system based upon the existing mental model brought to the interaction. If the designer has correctly reflected the user's mental model in design, the user's mental model is reinforced and a feeling that the interface is intuitive will likely develop. Continued interaction with the system may influence and modify the user's concept of the system, and his or her mental model may be modified as well. Refinement of this mental model, a normal process, is aided by well-defined distinctions between objects and by being consistent across all aspects of the interface.

What happens, however, if the new system does not accurately reflect the user's existing mental model? The results include breakdowns in understanding, confusion, errors, loss of trust, and frustration. Another result is an inability to perform the task or job.

Historically, when system designers have known in advance there was a gap between their conceptual model and the mental model the user would bring to the new system, designers have tried to bridge this gap through extensive documentation and training. The problems with this approach are as follows: People are unproductive while being trained, people rarely read the documentation and training materials, and even if the training material is read, the material is presented out of context. This creates difficulties for the users in understanding the material's relevance to their needs and goals.

Design Standards or Style Guides

This text provides an extensive collection of design principles and design rules. Additional sources of design guidance can be found in design standards and style

https://hemanthrajhemu.github.io

guides. A design standard or style guide documents an agreed-upon way of doing something. In interface design it describes the appearance and behavior of the interface and provides some guidance on the proper use of system components. It also defines the interface principles, rules, guidelines, and conventions that must be followed in detailed design. It will be based on the characteristics of the system's hardware and software, the principles of good interface and screen design to be presented in Step 3 and other steps, the needs of system users, and any unique company or organization requirements that may exist. In some instances, federal laws may require the application of specific design principles.

Value of Standards and Guidelines

Developing and applying design standards or guidelines achieves design consistency. This is valuable to users because the standards and guidelines

- Allow faster performance.
- Reduce errors.
- Reduce training time.
- Foster better system utilization.
- Improve satisfaction.
- Improve system acceptance.
- Reduce development and support costs.

They are valuable to system developers because they

- Increase visibility of the human-computer interface.
- Simplify design.
- Provide more programming and design aids, reducing programming time.
- Reduce redundant effort.
- Reduce training time.
- Provide a benchmark for quality control testing.

Business System Interface Standards, Guidelines, and Style Guides

Human-computer interface guidelines were not developed in isolation but had their beginnings with paper-based media. Throughout its history, guidelines for effective presentation of information on paper slowly evolved based upon research and designer experience. The structure of forms, newspapers, and text were molded to permit easy user comprehension and usage. The advent of the character-based visual display terminal (VDT) provided a new electronic presentation medium but the characteristics of people did not change. Neither did the characteristics of information. It was quickly realized that many guidelines used in the paper world could easily migrate to the computer world. So, knowledge gained in the paper worlds was transferred to the electronic world.

https://hemanthrajhemu.github.io

Some businesses and organizations developed and implemented human-computer interface and screen design guidelines as far back as the early 1970s (for example, see Galitz and DiMatteo, 1974). The first text to present a compilation of interface and screen design guidelines was that of Galitz (1981). In 1986 the United States Air Force released a set of design guidelines for its user interface designers (Smith and Mosier, 1986). Today, some people have questioned the applicability of older guidelines because many were developed using mainframe systems. To see if this was true, Nielsen (2005a) evaluated the relevance of the Smith and Mosier guidelines to today's Web world. Using a sampling technique, he concluded that 90 percent of the 1986 guidelines are still valid, though several are less important because they relate to design elements that are rarely used today. This is hardly surprising because the human and information rate of change is glacial compared to technology.

It was not until the late 1980s that the computer industry in general and other end-user organizations fully awakened to the need for guidelines. Then, a flurry of guideline documents began to appear. Some were for internal company or organization use only; others were published for general consumption by companies such as IBM (1987), Sun Microsystems (1990), Apple Computer (1992b), and Microsoft (1992). These guidelines have been updated over the last decade, and today many of these interface guidelines are published on the Web as well. A current listing of important available commercial style guides on the Web is shown in Table 2.1. In addition, IBM has available its Common User Access (CUA) guidelines developed in the late 1980s (IBM, 1992). It is only available in print form.

Unfortunately, past research on guideline utilization in business systems has hardly been encouraging. Standards conformance problems identified include difficulties in finding information being sought, difficulties in interpreting information, and numerous rule violations. An early study by Thovtrup and Nielsen (1991), for example, reported that designers were only able to achieve a 71 percent compliance with a two-page standard in a laboratory setting. In an evaluation of three real systems, they found that the mandatory rules of the company's screen design standard were violated 32 to 55 percent of the time.

Table 2.1: Commercial Style Guides

Apple Human Interface Guidelines for the Macintosh `http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines`
IBM Ease of Use Web site `www-03.ibm.com/easy/page/558`
Microsoft Windows XP User Interface Guidelines `http://www.microsoft .com`
Sun Microsystems Java Look and Feel Design Guidelines `http://java.sun.com/products/jlf/ed2/book/HIGTitle.html`

https://hemanthrajhemu.github.io

Thovtrup and Nielsen, in analyzing why the rules in the screen design standard were broken, found a very positive designer attitude toward the standard, both in terms of its value and content. Rules were not adhered to, however, for the following reasons:

- An alternative design solution was better than that mandated by the standard.
- Available development tools did not allow compliance with the standard.
- Compliance with the standard was planned, but time was not available to implement it.
- The rule that was broken was not known or was overlooked.

User Interface Standards

A user interface standard is an official set of internationally agreed-upon design approaches and principles for interface design. An organization involved in developing such standards is the International Organization for Standardization (ISO). ISO is a worldwide federation of national standards bodies composed of various technical committees that are actively involved in creating worldwide standards. Several ISO standards now exist for human-computer interaction and usability. They are as follows:

- **ISO 9241: Ergonomic requirements for office work with visual display terminals.**

  Contains guidelines for the ergonomic aspects of hardware and software in the design and use of display terminals. Topics addressed include display requirements, the use of icons and menus, and the work environment.
- **ISO 14915: Software ergonomics for multimedia user interfaces.**

  Presents guidelines for the design of multimedia user interfaces.
- **ISO 13407: Human-centered design processes for interactive systems.**

  Provides guidance on user-centered design activities through the development process for computer-based interactive systems. Its focus is on management of the design process.
- **ISO/CD 20282: Ease of operation of everyday products.**

  A four-part standard describing how to design products to ensure that they are usable by consumers. It focuses on ensuring that the needs of a wide range of users are met, including those with disabilities. Part 1, *Context of use and user characteristics*, and Part 2, *Test method* have been created to a draft level.

More information concerning these standards is available at the Web sites of ISO (`www.ISO.org`) and the American National Standards Institute (ANSI) (`http://webstore.ansi.org`).

Web Guidelines and Style Guides

Web interface design issues have also unleashed a plethora of Web-specific design guidelines and style guides, many of which are found on the Web itself. These guidelines

https://hemanthrajhemu.github.io

can be seen on the sites of the various computer companies and interface consulting firms, in newsletters, and even on personal Web sites. Although many of the traditional interface guidelines are applicable in a Web environment, the Web imposes a host of additional considerations.

The haste to publish Web design guidelines has been fueled by the explosive growth of the Web and a corresponding explosive growth in the number of developers creating sites for public access. In the brief existence of the Web, there has not been an opportunity for conventions and style guides to be properly developed and then accepted by the development community. This situation is made worse by the fact that many Web developers have had limited knowledge of traditional interface issues and concerns, and many are unfamiliar with the traditional interface design guidelines. Web guideline documents have attempted to fill this void.

Because a Web user can freely move among a seemingly endless supply of sites, no site will be seen in isolation. Commonality is of even greater importance than in business systems, where movement occurs between small numbers of applications. Today many unique Web standards and guidelines are evolving through research as well as by trial and error. Things are being tried to see what works best. De facto standards are being established when an overwhelming majority of big sites focus on one way to do something. An example is a menu bar that now frequently appears down the left side of the page. Standards and conventions will continue to evolve with experience and as the results of usability research become available. Koyani et al. (2004) have compiled a research-based listing of Web design guidelines. Worldwide standards are also being looked at by organizations such as the World Wide Web Consortium (2001).

Customized Style Guides

A customized style guide can also be created for an organization or system to be developed. Relevant materials from various standards, style guides, and other usability sources can be pulled together to create a document reflecting an organization's specific needs. In creating such a document

- Include checklists to present principles and guidelines.
- Provide a rationale for why the particular guidelines should be used.
- Provide a rationale describing the conditions under which various design alternatives are appropriate.
- Include concrete examples of correct design.
- Design the guideline document following recognized principles for good document design.
- Provide good access mechanisms such as a thorough index, a table of contents, glossaries, and checklists.

Checklists and rationale. Provide checklists for presenting key principles and guidelines. Checklists permit ease in scanning, ease in referring to key points, and make a document more readable by breaking up long sequences of text. Also provide a rationale for why the particular guidelines should be used. Understanding the reasoning will increase guideline acceptance. This is especially important if

https://hemanthrajhemu.github.io

the guideline is a deviation from a previous design practice. Also, when two or more design alternatives exist, provide a rationale describing the conditions under which the alternatives are appropriate. It may not be easy for designers to infer when various alternatives are appropriate. You have probably noticed that this book uses a checklist format to present key guidelines and thoughts, and guideline rationale is described in the text.

Concrete examples. To be effective, a guideline must include many concrete examples of correct design. Imitation is often a way people learn.

Document design and access. Always design the document, be it paper or electronic, by following recognized principles for good document design. This greatly enhances readability. Provide good access mechanisms such as a thorough index, a table of contents, glossaries, and checklists. An unattractive or hard-to-use document will not be inviting and consequently will not be used.

Design Support and Implementation

- Use all available reference sources in creating the guidelines.
- Use development and implementation tools that support the guidelines.
- Begin applying the guidelines immediately.

Available Reference Sources. Use all the available reference design sources in creating guidelines. References include this text, other books on user interface design, project-specific guidelines, and the style guides for interface design and Web design created by companies such as Apple, IBM, Microsoft, and Sun Microsystems. Other reference sources that meet development needs should also be utilized.

Tools. Use tools that support implementation of the guidelines that are established. Development tools make the design process much easier. If the design tools cannot support the guideline, it cannot be adhered to.

Applying the Guidelines. Two questions often asked are, "Is it too late to develop and implement standards?" and, "What will be the impact on systems and screens now being used?" To address these questions, researchers reformatted several alphanumeric inquiry screens to improve their comprehensibility and readability. When these reformatted screens were presented to expert system users, decision-making time remained the same but errors were reduced. For novice system users, the reformatted screens brought large improvements in learning speed and accuracy. Therefore, it appears that changes that enhance screens will benefit both novice and expert users already familiar with the current screens. It is never too late to begin to change.

System Training and Documentation Needs

Training and documentation are also an integral part of any development effort.

https://hemanthrajhemu.github.io

Training

System training will be based on user needs, system conceptual design, system learning goals, and system performance goals. Training may include such tools as formal or video training, manuals, online tutorials, reference manuals, quick reference guides, and online help. (Various types of training methods are discussed in more detail in Step 9.) Training needs must be established and training components developed as the design process unfolds. This will ensure that the proper kinds of training are defined, properly integrated with the design, and developed correctly. This will also assure that the design is not imposing an unreasonable learning and training requirement on the user. Any potential problems can also be identified and addressed earlier in the design process, reducing later problems and modification costs.

MYTH **That problem can be handled with documentation and training.**

Documentation

System documentation is a reference point, a form of communication, and a more concrete design — words that can be seen and understood. It will also be based on user needs, system conceptual design, and system performance goals. Creating documentation during the development progress will uncover issues and reveal omissions that might not otherwise be detected until later in the design process. As with training, any potential problems can be identified and addressed earlier in the design process, again reducing later problems and modification costs.

Step 2 Exercise

An exercise for Step 2 can be found on this book's companion Web site, www.wiley.com/college/galitz.

https://hemanthrajhemu.github.io

STEP

3

Understand the Principles of Good Interface and Screen Design

A well-designed interface and screen

- Reflects the capabilities, needs, and tasks of its users.
- Is developed within the physical constraints imposed by the hardware on which it is displayed.
- Utilizes the capabilities of its controlling software effectively.
- Achieves the business objectives of the system for which it is designed.

To accomplish these goals, the designer must first understand the principles of good interface and screen design. What follows is an extensive compilation of general screen design guidelines for the user interface. It includes a detailed series of guidelines dealing with user considerations, including

- A test for a good design.
- Organizing screen elements.
- Screen navigation and flow.
- Visually pleasing composition.
- Typography.
- Keying procedures.
- Data output.
- Web sites and Web pages.

https://hemanthrajhemu.github.io

- Statistical graphics.
- Technological considerations.

The step concludes with considerations imposed by a system's hardware and software.

Human Considerations in Interface and Screen Design

Use of a screen or Web page, and a system, is affected by many factors. These include how much information is presented, how the information is organized, what language is used to communicate to the user, how distinctly the components are displayed, what aesthetics are used, and how consistent a screen or page is with other screens or pages. First, let's look at what aspects of poor design can be distracting to the user, what a user is looking for in good design, and the kinds of things screen users do interacting with a system or Web site. Then, we'll address the principles of good design.

How to Discourage the User

Barnett (1993, 2005) in his excellent books on paper form design has compiled a list of factors that, when poorly designed, hinder the use of paper forms. These factors certainly apply to electronic forms, screens, and Web pages as well, and include

- Unclear captions and badly worded questions. These cause hesitation, and rereading, to determine what is needed or must be provided. They may also be interpreted incorrectly and cause errors.
- Improper type and graphic emphasis. Important elements are hidden. Emphasis is drawn away from what is important to that which is not important.
- Misleading headings. These also create confusion and inhibit one's ability to see existing relationships.
- Information requests perceived to be irrelevant or unnecessary. The value of what one is doing is questioned, as is the value of the system.
- Information requests that require one to backtrack and rethink a previous answer, or look ahead to determine possible context. Inefficiency results, and mistakes increase.
- Cluttered, cramped layout. Poor layout creates a bad initial impact and leads to more errors. It may easily cause system rejection.
- Poor quality of presentation, legibility, appearance, and arrangement. Again, this degrades performance, slowing the user down and causing more errors.

https://hemanthrajhemu.github.io

Howlett (1995), based upon her experiences at Microsoft, suggests the most common problems in visual interface design are

- Visual inconsistency in screen detail presentation and with the operating system.
- Lack of restraint in the use of design features and elements.
- Overuse of three-dimensional presentations.
- Overuse of too many bright colors.
- Poorly designed icons.
- Bad typography.
- Metaphors that are either overbearing, too cute, or too literal, thereby restricting design options.

These kinds of problems, she concludes, lead to screens that can be chaotic, confusing, disorganized, distracting, or just plain ugly.

Web pages also present to the user an expanded variety of distractions because of the extensive incorporation of graphics. These distractions include

- Numerous visual and auditory interruptions.
- Extensive visual clutter.
- Poor information readability.
- Poor information legibility.
- Incomprehensible screen components.
- Confusing and inefficient navigation.
- Inefficient operations and extensive waste of user time.
- Excessive or inefficient page scrolling.
- Information overload.
- Design inconsistency.
- Outdated information.
- Stale design caused by emulation of printed documents and past systems.

Nielsen (2005e) has for more than ten years compiled lists of the top design mistakes in Web design. This list, based upon submissions of his Alert Box readers, includes the following:

1. Legibility problems, including small fonts and poor contrast between text and backgrounds.
2. Nonstandard presentation of links.
3. Annoying use of Flash.
4. Content that has not been written for the Web.
5. Poor search facilities.
6. Browser incompatibilities.
7. Cumbersome forms to fill out.

https://hemanthrajhemu.github.io

8. No contact or other company information included.
9. Frozen layouts with fixed page widths resulting in cutting off the side of a page.
10. Inadequate photo enlargement.

Poor design is not a new phenomenon. It has existed since people began interacting with media used for presenting and collecting information. Some of the distractions have been around a long time; others are fairly new, the by-product of technological advances. Many of these problems could be avoided if designers adhered to the design guidelines now widely available.

Interface distractions cause a person to think about things they shouldn't have to think about, and divert one's attention from performing a task or satisfying a need. The responses to poor design described in Step 1 are often the result of these kinds of problems. All distractions and discouragements must be eliminated in design.

What Users Want

What are people looking for in the design of screens? One organization asked a group of users and got the following responses:

- An orderly, clean, clutter-free appearance.
- An obvious indication of what is being shown and what should be done with it.
- Expected information located where it should be.
- A clear indication of what relates to what, including options, headings, captions, data, and so forth.
- Plain, simple English.
- A simple way of finding out what is in a system and how to get it out.
- A clear indication of when an action can make a permanent change in the data or system.

The desired direction is toward simplicity, clarity, and understandability — qualities lacking in many of today's screens and Web pages.

What Users Do

When interacting with a computer, a person

1. **Identifies a task to be performed or a need to be fulfilled.** The task may be very structured, including activities such as: Enter this data from this form into the system, answer a specific question regarding the status of an order, or collect the necessary information from a customer to make a reservation. Alternatively, the task may have some structure but also include more free-form activities, including answering questions such as: What is the best local rehabilitation program in which to enroll my client, or what are my customer's exact needs and then which of our products features are best suited for him or her. Finally, the need may be very general or even vague. Where should I take an exotic vacation near a beautiful beach? Where can I get the best price on a new PC?

https://hemanthrajhemu.github.io

2. **Decides how the task will be completed or the need fulfilled.** For a structured or semi-structured task a set of transaction screens will be available. The proper transaction is identified and the relevant screen series retrieved. To satisfy a general or vague need will require browsing or searching through screens that might possibly have relevance.
3. **Manipulates the computer's controls.** To perform the task or satisfy the need, the keyboard, mouse, and other similar devices are used to select choices from lists, choose commands to be performed, key data into text boxes, and so forth.

MAXIM **People will spend many, many hours *staring* at your screens.**

4. **Gathers the necessary data or content while filtering out meaningless data or content.** Using structured and semi-structured transaction screens, the user collects information from its source: a form, a coworker, or a customer. This information is identified on the screen, or placed on the screen, through control manipulation. To satisfy a general or vague need may require following Web site links down many paths. Path activities may also require other kinds of control manipulation as well.
5. **Forms judgments resulting in decisions relevant to the task or need.** Structured transactions will require minimal decision-making. Has all the data been collected and is the data valid? Has the transaction been successfully accepted by the system? If not accepted, why not? Semi-structured transactions, in addition, may require decisions such as: Which set of screens, from all available, should I use to complete this process? How much information is needed to complete the sale of this particular product, make a reservation in this hotel, or complete the enrollment process for a specific program? To satisfy a general or vague need will require decisions like: Where should I look to get my answer? Which link should I follow? Is this all the information I need? How do I order it?

Interface Design Goals

To make an interface easy and pleasant to use, then, the goal in design is to

- Reduce visual work.
- Reduce intellectual work.
- Reduce memory work.
- Reduce motor work.
- Minimize or eliminate any burdens or instructions imposed by technology.

The result will always be improved user productivity and increased satisfaction. Let's begin the review of the principles of good design by applying the following simple test to all screens or Web pages.

https://hemanthrajhemu.github.io

The Test for a Good Design

- Can all screen or Web page elements be identified by cues other than by reading the words that make them up?

A simple test for good design does exist. A screen that passes this test will have surmounted the first obstacle to effectiveness. The test is this: Can all screen elements (field captions, data, title, headings, text and information, types of controls, navigation elements, and so on) be identified without reading the words that identify or comprise them? That is, can a component of a screen be identified through cues independent of its content? If this is so, a person's attention can quickly be drawn to the part of the screen that is relevant at that moment. People look at a screen or page for a particular reason, perhaps to locate a piece of information such as a customer name, to identify the name of the screen, or to find an instructional or error message. The *signal* at that moment is that element of interest on the screen. The *noise* is everything else on the screen. Cues independent of context that differentiate the components of the screen will reduce visual search times and minimize confusion.

Try this test on the front page of the morning newspaper. Where is the headline? A story heading? The weather report? How did you find them? The headline was identified probably by its large and bold type size; story headings, again by a type size visually different than other page components; the weather report, probably by its location (bottom right? top left?). Imagine finding the headline on the front page of the newspaper if the same type size and style were used for all components and their positions changed from day to day. If this is true of a screen or Web page, scanning will be a lengthy and cumbersome process, and the screen or Web page will not be appealing to use.

Unfortunately, many of today's screens and Web pages cannot pass this simple test and are unnecessarily difficult to use. Almost all the tools available to the creator of the newspaper's front page are now available to the screen designer. Technology has added some additional weapons. An effective solution can be achieved. It simply involves the thoughtful use of display techniques, consistent location of elements, the proper use of "white space" and groupings, and an understanding of what makes visually pleasing composition. The best interfaces make everything on the screen *obvious*.

Screen and Web Page Meaning and Purpose

- Each element
- Every control
- All text
- The screen organization
- All emphasis
- Each color
- Every graphic

https://hemanthrajhemu.github.io

- All screen animation
- Each message
- All forms of feedback
- Must
  — Have meaning to users.
  — Serve a purpose in performing tasks.

All interface elements must have meaning to users and serve a purpose in performing tasks or fulfilling needs. If an element does not have meaning for the user, do not include it in the interface because it is *noise*.

Signals and Noise

Noise is *useless* information. Signals are *useful* information. Noise reduces clarity of a screen or Web page because useful information is diluted by useless noise. Noise is distracting, competes for the user's attention, and contributes to information overload. That which is important will be more difficult to find.

The objective in design is to minimize noise and maximize signals. Present information efficiently, simply, clearly, concisely, and appropriately. Remove all unnecessary elements. At any moment in an interaction, however, (as previously described in "The Test for a Good Design") a signal can become a noise, and a noise a signal in the mind of the screen viewer. In this case what is noise and what is a signal should be immediately obvious in some easily discernable manner.

Organizing Elements Clearly and Meaningfully

Visual clarity is achieved when the display elements are organized and presented in meaningful, understandable, and recognizable ways. A clear and clean organization makes it easier to recognize a screen's essential elements and to ignore its secondary information when appropriate. Clarity is influenced by a multitude of factors: consistency in design, a visually pleasing composition, a logical and sequential ordering, the presentation of the proper amount of information, groupings, and alignment of screen items. What must be avoided is visual clutter created by indistinct elements, random placement, and confusing patterns.

Consistency

- Provide real-world consistency. Reflect a person's experiences, expectations, work conventions, and cultural conventions.
- Provide internal consistency. Observe the same conventions and rules for all aspects of an interface screen, and all application or Web site screens, including
  — Operational and navigational procedures.
  — Visual identity or theme.

https://hemanthrajhemu.github.io

— Component
  - Organization.
  - Presentation.
  - Usage.
  - Locations.
- Follow the same conventions and rules across all related interfaces.
- Deviate only when there is a clear benefit for the user.

People strive for consistency in their attitudes, thoughts, and beliefs. Similarly, consistency is a very important component of design. Ozok and Salvendy (2000; 2004) found that users commit fewer errors when the visual and linguistic aspects of information sites are consistent. Other studies have found that consistency leads to a reduction in task completion times, an increase in user satisfaction, and a reduction in learning time (Koyani et al., 2004). Quite simply, consistency greatly aids learning. It establishes expectations, permits a person to employ conceptual learning and transfer training, and enables the user to easily anticipate the location of screen elements of interest.

Inconsistency forces one to memorize, and remember, a variety of different ways to do something or interpret what is presented on the screen. Inconsistency makes it difficult for a coherent structure to emerge. It can also be distracting, causing a person to wonder why things are different. Inconsistency also creates a screen variation that makes it difficult to notice another variation that may be important for a person's task or need.

In Web site design consistency greatly enhances visual scanning, a frequent user activity. It also fosters a sense of place, reassuring a person that he or she is rooted in a certain location. This provides stability and reduces navigation confusion.

MYTH **Users can get used to anything!**

So, provide both functional and aesthetic consistency in design. Be consistent with the real world in which a person already exists. This world will already have been well learned and possess an established mental model. Generalization to the system interface will most easily occur. Provide internal consistency so that learning may be focused on the task or job, not on irrelevancies. As far as consistent location of screen elements, people do tend to have good memories for the locations of things. Take advantage of this phenomenon. The graphical system products and style guides have established consistent locations for most screen elements, as well as numerous other conventions. In Web site design location and presentation, consistency is slowly evolving. Follow all current conventions and new conventions as they are established. Also be consistent across user interfaces for all the reasons already mentioned.

If an inconsistency will *benefit* the user, such as calling attention to something extremely critical, consider deviating from consistency. Be wary of too many deviations, though, as the impact of each inconsistency will diminish screen usability.

https://hemanthrajhemu.github.io

Starting Point

- Provide an obvious starting point in the screen's upper-left corner.
- Focus user attention on the most important parts of a screen or page.

How a person scans an array of information is dependent upon the composition of the display. Learning also influences it.

Textual Displays

Eyeball fixation studies indicate that in looking at displays of textual information, usually one's eyes move first to the upper-left center of the display, and then quickly move through the display in a clockwise direction. Streveler and Wasserman (1984) found that visual targets located in the upper-left quadrant of a screen were found fastest and those located in the lower-right quadrant took longest to find. This general top-to-bottom, left-to-right orientation through text is generally assumed to be a habit formed from reading text.

Graphical and Web Displays

Graphical displays modify a person's scanning behavior because of the additional visual cues that weight and composition provide. Hornoff and Halverson (2003) found that people do take advantage of visual details such as white space or components that stand out conspicuously from other components. Studies have also found that

- People tend to look at text first, not images.
- Larger type dominates over smaller type.
- Changing information is looked at before non-changing information.

Eye tracking studies indicate that in information-based sites people tend to look first at text, not images, by a margin of nearly two to one. Faraday (2001) found that when text and images are of the same size, text is more likely to be the entry point into a Web page. To become an entry point, images must be much larger than text. This study also found that larger size text dominates smaller size text as an entry point, and that normal size text — even if displayed in bold or as a hyperlink, rarely acts as an entry point.

Learning also impacts scanning behavior. A person who is familiar with information presented in a consistent way is more likely to look at areas that often change rather than areas that stay the same (Lidwell et al., 2003).

In conclusion, provide an obvious starting point in the upper-left corner of the screen. This is near the location where visual scanning begins and will permit a left-to-right, top-to-bottom reading of information or text as is common in Western cultures. For information with no obvious or prescribed flow, focus the users' attention on the most important components of the displayed information array.

https://hemanthrajhemu.github.io

An eye-tracking study by Nielsen (2006c) found that people often read Web pages in an F-shaped pattern: two horizontal stripes followed by a vertical stripe. This eye movement is fast and appears to differ from previous research on paper documents and other kinds of screens. Nielsen says this reading/scanning pattern is comprised of these three components:

- People first look horizontally, usually across the upper part of the content area. (The F's top bar.)
- Next, people move down the page and then look across in a second horizontal movement that typically covers a shorter space than the previous movement. (The F's lower bar.)
- Finally, people scan the contents' left side in a vertical movement. Scan speeds are quite variable, sometimes slow and systematic, other times quite fast. (The F's stem.)

Movement patterns are not always an F shape, sometimes resembling an upside down L or an E. Generally, however, Nielsen says, the looking pattern roughly resembles an F, and the distance between the top and bottom bar can vary. These distance differences are probably caused by the structure of the Web page and how it's attention-getting components are organized.

Ordering of Data and Content

- Divide information into units that are logical, meaningful, and sensible.
- Organize by the degree of interrelationship between data or information.
- Provide an ordering of screen units of information and elements that is prioritized according to the user's expectations and needs.
- Possible ordering schemes include
  — Conventional.
  — Sequence of use.
  — Frequency of use.
  — Function.
  — Importance.
  — General to specific.
- Form groups that cover all possibilities.
- Ensure that information that must be compared is visible at the same time.
- Ensure that only information relative to the users' tasks or needs is presented on the screen.

In application and Web page design an organizational scheme's goal is to keep to a minimum the number of information variables the user must retain in short term memory. A logical, meaningful, and sensible arrangement of screen data and content will lower memory requirements. In Web page design information importance is the strongest determiner of content ordering.

https://hemanthrajhemu.github.io

In ordering applications, units of information and screen elements should be prioritized according to the user's needs and expectations. People develop expectations on how to accomplish certain tasks and find different types of information. A meaningful organization permits faster learning. In Web site design it is also easier to develop a clear navigation system if the site is meaningfully organized. Clear organization also makes it easier for Web users to find what they need, and to predict where a navigation link will take them.

Common information ordering schemes include the following:

Conventional. Through convention and custom, some ordering schemes have evolved for certain elements. Examples are by days of the week, by months of the year, by one's name and address, or along a timeline. These elements should always be ordered in the customary way.

Sequence of use. Sequence of use grouping involves arranging information items in the order in which they are commonly received or transmitted, or in natural groups. An address, for example, is normally given by street, city, state, and zip code. Another example of natural grouping is the league standings of football teams, appearing in order of best to worst records.

Frequency of use. Frequency of use is a design technique based on the principle that information items used most frequently should be grouped at the beginning, the second most frequently used items grouped next, and so forth.

Function or category. Information items are grouped according to their purpose or by some common parameter. All items pertaining to insurance coverage, for example, may be placed in one location. Transportation vehicles may be grouped within the categories of planes, trains, and automobiles. Such grouping also allows convenient group identification using headings for the user. Subcategories with subheadings may also be established.

Importance. Importance grouping is based on the information's importance to the user's task or need. Important items are placed first or in the most prominent position. Items may be organized from best to worst or largest to smallest.

General to specific. If some data is more general than others, the general elements should precede the specific elements. This will usually occur when there is a hierarchical relationship among data elements. This is a common Web site organization scheme.

Ordering normally reflects a combination of these techniques. Information may be organized functionally but, within each function, individual items may be arranged by sequence or importance. Numerous permutations are possible. The ordering scheme established must encompass all the information.

Ensure that information that must be compared is always visible to the user at the same time. A common problem in design is forcing the user to remember things located on different screens, or within a screen but scrolled out of sight. The corollary is to ensure that only information relative to the tasks or needs at hand is presented on a screen. Irrelevant information is noise.

https://hemanthrajhemu.github.io

Ordering Web Pages

- Establish levels of importance.
- Place critical information near the top of the Web site.
- Place important items at the top of a page.
- Organize information clearly.
- Place important items consistently.
- Facilitate scanning.
- Structure for easy comparison.

In addition to the aforementioned considerations, the design and ordering of a Web page must reflect the estimated importance of page elements to the viewer. These guidelines, compiled, described, and referenced by Koyani et al. (2004) reflect Web page ordering research.

Establish levels of importance. Establish a high to low level of importance for items of information to appear on the Web site. The nature of the information should suggest ways to divide and organize information most useful to viewers. A technique to aid in determining importance and organizing Web sites is that of card sorting described in Step 2. Follow this high-to-low approach throughout the Web site.

Place critical information near the top of the Web site. Critical information should be placed as near to the top of the Web site as possible. The more steps needed for people to find what they are looking for, the greater the probability an incorrect choice will be made. Critical information should always be located within two or three steps of the homepage.

Place important items at the top of a page. All important information should be placed high on the page so it can be quickly found. This includes critical content and navigation options. The most important items should be positioned top-center where viewers look first. Less used information should be placed lower on the page. Text at the bottom of a page is rarely seen. On navigation pages, most major choices should be visible with no or minimum scrolling.

Placing critical or important information first also has these benefits:

- It is more likely to be remembered.
- It establishes context aiding the interpretation of subsequent items.
- It enhances efficient searching and scanning of information.

Organize information clearly. Provide a clear and logical organizational structure that reflects the user's needs and the Web site's goals at several levels: Web site, page, and paragraph. Clear organization enhances understanding of the site's component relationships and aids locating of desired information.

Place important items consistently. Provide consistency in item location throughout all Web pages. This is especially important for clickable navigation components.

https://hemanthrajhemu.github.io

Facilitate page scanning. So that desired information can be easily found, structure each content page for efficient visual scanning. Studies report that about 80 percent of people scan newly presented pages. Only about 16 percent read word-for-word. People spend about 12 percent of their time trying to find what they are looking for (Koyani et al., 2004).

Structure for easy comparison. Pages should be structured for easy comparison of important related components. Place in close physical proximity to one another items that will be used or analyzed together. This avoids taxing the user's memory.

Navigation and Flow

- Provide an ordering of screen information and elements that
  — Is rhythmic, guiding a person's eye through the display.
  — Encourages natural movement sequences.
  — Minimizes pointer and eye movement distances.
- Locate the most important and most frequently used elements or controls at the top left.
- Maintain a top-to-bottom, left-to-right flow.
- Assist in navigation through a screen by
  — Aligning elements.
  — Grouping elements.
  — Using line borders.
- Through focus and emphasis, sequentially, direct attention to items that are
  1. Critical.
  2. Important.
  3. Secondary.
  4. Peripheral.
- Tab through windows in logical order of displayed information.
- Locate command buttons at end of the tabbing order sequence.
- When groups of related information must be broken and displayed on separate screens, provide breaks at logical or natural points in the information flow.

Navigation through a screen or page should be obvious and easy to accomplish. Navigation can be made obvious by grouping and aligning screen controls, and judiciously using line borders to guide the eye. Sequentially, direct a person's attention to elements in terms of their importance. Using the various display techniques, focus attention on the most important parts of a screen. Always tab through a screen in the logical order of the information displayed, and locate command buttons at the end of the tab order sequence. Guidelines for accomplishing all of these general objectives will be found in subsequent pages.

The direction of movement between screen items should be obvious, consistent, and rhythmic. The eye, or pointer, should not be forced or caused to wander long distances about the display seeking the next item. The eye can be guided through the screen

https://hemanthrajhemu.github.io

with lines formed through use of white space and display elements. More complex movements may require the aid of display contrasts. Sequence of use can be made more obvious through the incorporation of borders around groupings of related information or screen controls. Borders provide visual cues concerning the arrangement of screen elements, because the eye will tend to stay within a border to complete a task. Aligning elements will also minimize screen scanning and navigation movements. In establishing eye movement through a screen, also consider that the eye tends to move sequentially, for example:

- From dark areas to light areas.
- From big objects to little objects.
- From unusual shapes to common shapes.
- From highly saturated colors to unsaturated colors.

These techniques can be used initially to focus a person's attention to one area of the screen and then direct it elsewhere.

For screens containing data, locate the most important or frequently used screen controls to the top left of the screen where initial attention is usually directed. This will also reduce the overall number of eye and manual control movements needed to work with a screen.

Then, maintain a top-to-bottom, left-to-right flow through the screen. This is contrary to the older text-based screen cursor movement direction that precedes left to right, then top to bottom. This top-to-bottom orientation is recommended for information *entry* for the following reasons:

- Eye movements between items will be shorter.
- Control movements between items will be shorter.
- Groupings are more obvious perceptually.
- When one's eye moves away from the screen and then back, it returns to about the same place it left, even if it is seeking the next item in a sequence (a visual anchor point remains).

Unfortunately, most product style guides recommend a left-to-right orientation. This orientation is based upon the presumption that since people read left-to-right, a screen must be organized in this way. Many screens, however, do not present text but listings of small pieces of information that must be scanned. All the research on human scanning finds a top-to-bottom presentation of information is best.

Why do we persist in this left-to-right orientation for nontextual screens? A common screen metaphor applied in today's systems is that of the paper form. We often see a paper form exactly replicated on a screen. Unfortunately, the left-to-right orientation of the typical form is poorly suited to the needs and characteristics of its user. Its complexity is generally higher than it should be, and its sequentiality is often not as obvious as it should be and certainly not at all efficient.

The left-to-right orientation of paper forms was not dictated by human needs but by mechanical considerations. The metaphor for earliest display screens six decades ago (although this term was not used then) was the typewriter. The left-to-right orientation of the typewriter was developed to permit one to type text on paper, a significant enhancement over handwriting as a medium of human communication. At some

https://hemanthrajhemu.github.io

point early in the typewriter's life, however, its ability to be used to complete paper forms also became evident. So, we started designing forms to be completed by typewriter. They had to be filled out left-to-right because the design of the typewriter made any other completion method very difficult for a person to do.

Our earliest display screens reflected this left-to-right entry orientation and have done so for many years. Today in our display-based world, the typewriter's mechanical limitations no longer exist. Let's shed the artificial constraints we have imposed upon ourselves and get rid of the left-to-right orientation for nontextual screens. A top-to-bottom orientation has many more advantages for the screen user.

Top-to-bottom orientation is also recommended for presenting displays of read-only information that must be scanned. This will be described shortly.

Visually Pleasing Composition

- Provide a visually or aesthetically pleasing composition possessing the following qualities:
  - — Balance
  - — Symmetry
  - — Regularity
  - — Predictability
  - — Sequentiality
  - — Economy
  - — Unity
  - — Proportion
  - — Simplicity
  - — Groupings

Eyeball fixation studies also indicate that during the initial scanning of a display in a clockwise direction, people are influenced by the symmetrical balance and weight of the titles, graphics, and text of the display. The human perceptual mechanism seeks order and meaning, trying to impose structure when confronted with uncertainty. Whether a screen has meaningful and evident form or is cluttered and unclear is immediately discerned. A cluttered or unclear screen requires that some effort be expended in learning and understanding what is presented. The screen user who must deal with the display is forced to spend time to learn and understand. The user who has an option concerning whether the screen will or will not be used may reject it at this point if the perceived effort in understanding the screen is greater than the perceived gain in using it.

MAXIM Aesthetic designs are perceived as easier to use than less-aesthetic designs.

An entity's design composition communicates to a person nonverbally, but quite powerfully. It is an unconscious process that attracts, motivates, directs, or distracts. Meaningfulness and evident form are significantly enhanced by a display that is pleasing to one's eye. Visually pleasing composition draws attention subliminally, conveying

https://hemanthrajhemu.github.io

a positive message clearly and quickly. A lack of visually pleasing composition is disorienting, obscures the intent and meaning, slows one down and confuses one.

The notion of what is artistic has evolved throughout history. Graphic design experts have, through perceptual research, derived a number of principles for what constitutes a visually pleasing appearance. These include balance, symmetry, regularity, predictability, economy, unity, sequentiality, proportion, simplicity, and groupings. Keep in mind that this discussion of visually pleasing composition does not focus on the words on the screen, but on the perception of structure created by such qualities as spacing, shapes, intensities, and colors, and the relationship of screen elements to one another. It is as if the screen were viewed through "squinted eyes," causing the words themselves to become a blur.

Balance

- Create screen balance by providing an equal weight of screen elements, left and right, top and bottom.

Balance, illustrated in Figure 3.1, is stabilization or equilibrium, a midway center of suspension. The design elements have an equal weight, left to right, top to bottom. The opposite of balance is instability: the design elements seemingly ready to topple over. Our discomfort with instability, or imbalance, is reflected every time we straighten a picture hanging askew on the wall. Balance is most often informal or asymmetrical, with elements of different colors, sizes and shapes being positioned to strike the proper relationships.

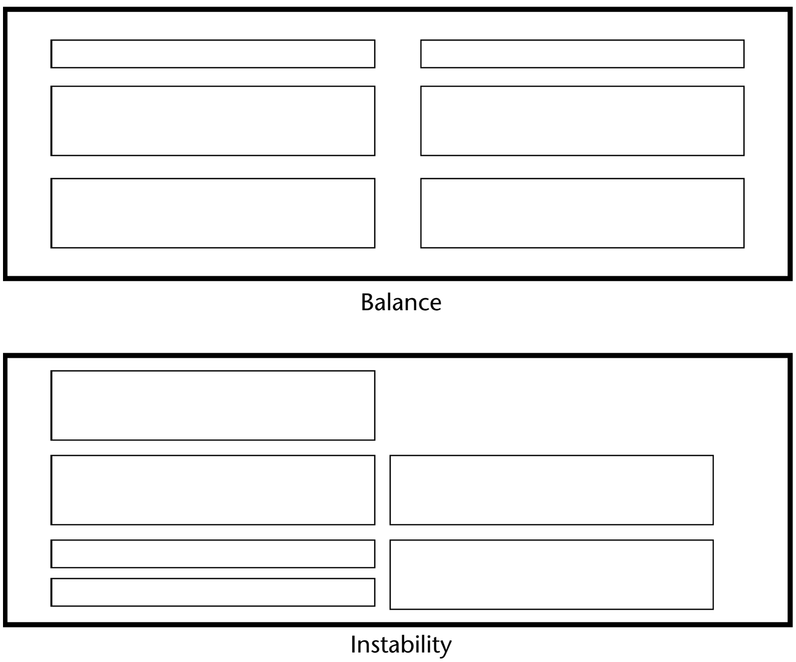


Figure 3.1: Balance (versus instability).

https://hemanthrajhemu.github.io

Dark colors, unusual shapes, and larger objects are "heavier," whereas light colors, regular shapes, and small objects are "lighter." Balance on a screen is accomplished through centering the display itself, maintaining an equal weighting of components on each side of the horizontal and vertical axis, and centering titles and illustrations.

In Web page design, vertical, or left-to-right balance is usually the most important concept. Web pages are often scrollable, thereby shifting the horizontal, or top-to-bottom, balance point as the screen is scrolled. Horizontal balance is therefore more difficult to maintain.

Symmetry

- Create symmetry by replicating elements left and right of the screen centerline.

Symmetry, illustrated in Figure 3.2, is axial duplication: A unit on one side of the centerline is exactly replicated on the other side. This exact replication creates formal balance, but the difference is that balance can be achieved without symmetry. Symmetry's opposite is asymmetry. Our eye tends to perceive something as more compressed or compact when it is symmetric. Asymmetric arrays are perceived as larger.

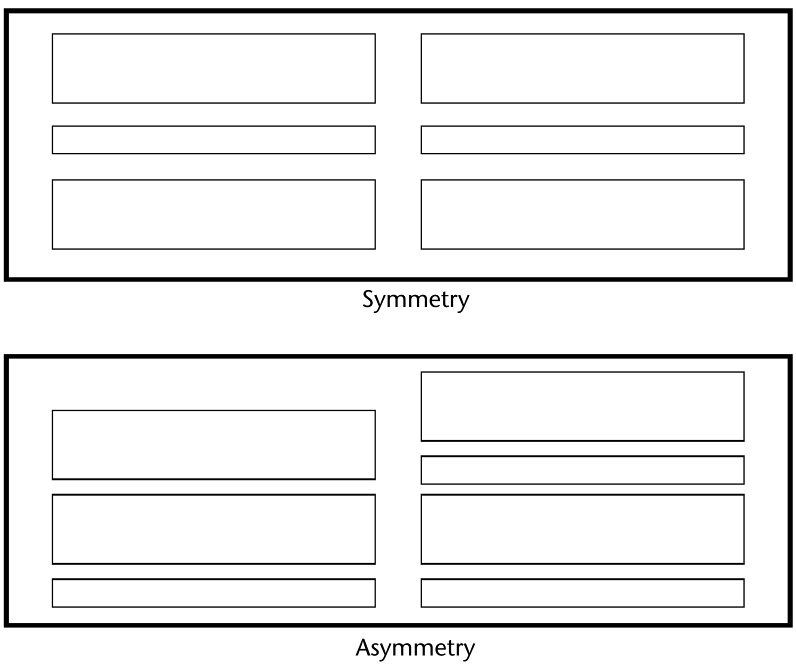


Figure 3.2: Symmetry (versus asymmetry).

https://hemanthrajhemu.github.io

Regularity

- Create regularity by establishing standard and consistently spaced horizontal and vertical alignment points.
- Also, use similar element sizes, shapes, colors, and spacing.

Regularity, illustrated in Figure 3.3, is a uniformity of elements based on some principle or plan. Regularity in screen design is achieved by establishing standard and consistently spaced column and row starting points for screen elements. It is also achieved by using elements similar in size, shape, color, and spacing. The opposite of regularity, irregularity, exists when no such plan or principle is apparent. A critical element on a screen will stand out better, however, if it is not regularized.

Predictability

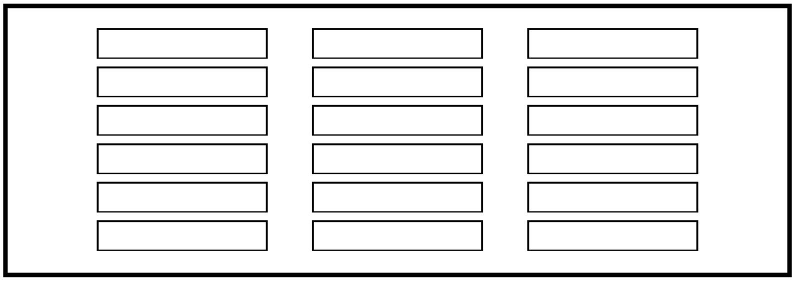
- Create predictability by being consistent and following conventional orders or arrangements.

Predictability, illustrated in Figure 3.4, suggests a highly conventional order or plan. Viewing one screen enables one to predict how another will look. Viewing part of a screen enables one to predict how the rest of the screen will look. The opposite of predictability — spontaneity — suggests no plan and thus an inability to predict the structure of the remainder of a screen or the structure of other screens. In screen design predictability is also enhanced through design consistency.

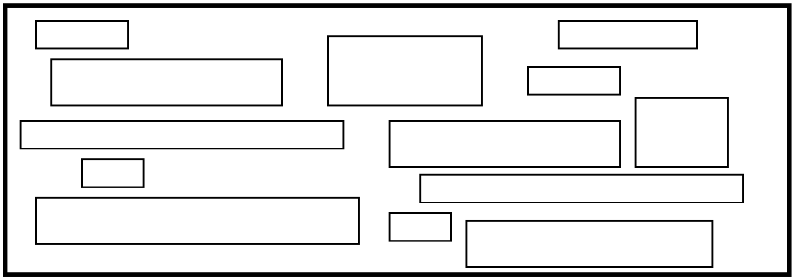
Sequentiality

- Provide sequentiality by arranging elements to guide the eye through the screen in an obvious, logical, rhythmic, and efficient manner.
- The eye tends to be attracted to
  — A brighter element before one less bright.
  — Isolated elements before elements in a group.
  — Graphics before text.
  — Color before black and white.
  — Highly saturated colors before those less saturated.
  — Dark areas before light areas.
  — A big element before a small one.
  — An unusual shape before a usual one.
  — Big objects before little objects.

https://hemanthrajhemu.github.io

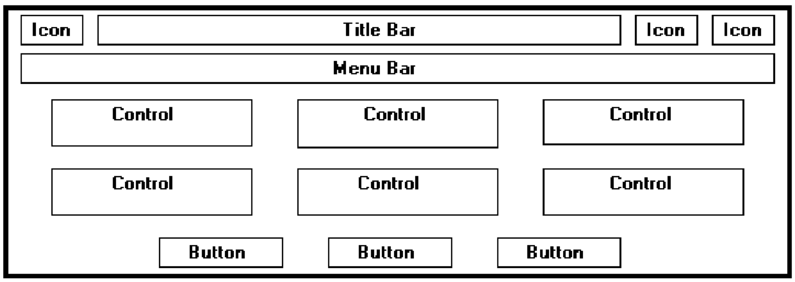


Regularity

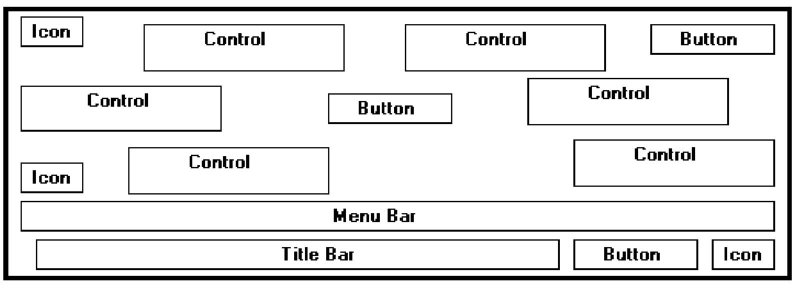


Irregularity

Figure 3.3: Regularity (versus irregularity).



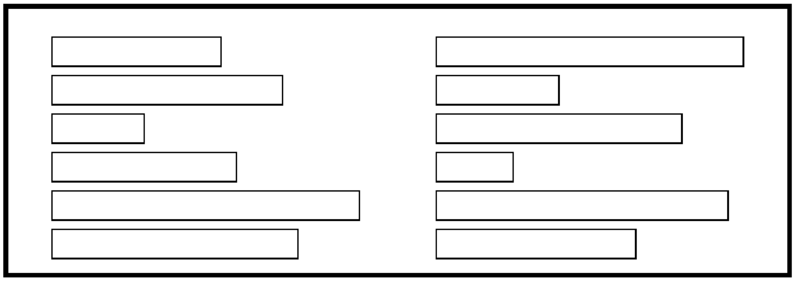
Predictability



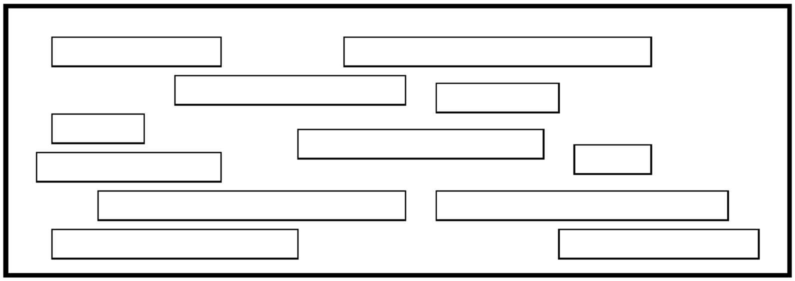
Spontaneity

Figure 3.4: Predictability (versus spontaneity).

https://hemanthrajhemu.github.io



Sequentiality



Randomness

Figure 3.5: Sequentiality (versus randomness).

Sequentiality, illustrated in Figure 3.5, is a plan of presentation to guide the eye through the screen in a logical, rhythmic order, with the most important information significantly placed. Sequentiality can be achieved by alignment, spacing, and grouping as illustrated. The opposite of sequentiality is randomness, whereby an arrangement and flow cannot be detected.

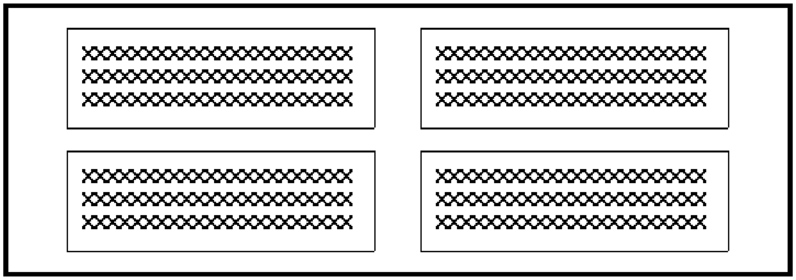
The eye tends to move first to the elements listed above, and then from one to the other. For example, it moves from highly saturated colors to unsaturated colors, from dark to light areas, from big to little objects, and from unusual to usual shapes.

Economy

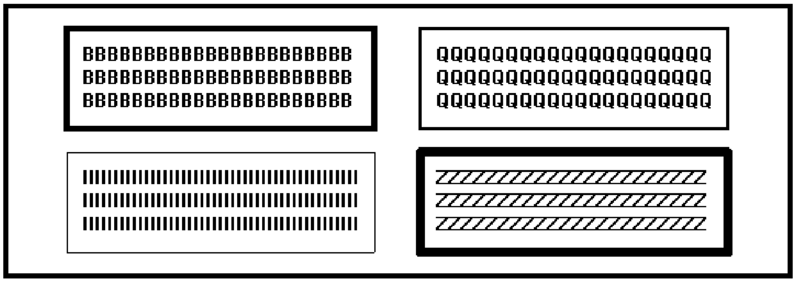
- Provide economy by using as few styles, display techniques, and colors as possible.

Economy, illustrated in Figure 3.6, is the frugal and judicious use of display elements to get the message across as simply as possible. The opposite is intricacy, the use of many elements just because they exist. The effect of intricacy is ornamentation, which often detracts from clarity. Economy in screen design means mobilizing just enough display elements and techniques to communicate the desired message, and no more. Historically, the use of color in screens has often violated this principle, with screens sometimes taking on the appearance of Christmas trees.

https://hemanthrajhemu.github.io



Economy



Intricacy

Figure 3.6: Economy (versus intricacy).

Unity

- Create unity by
  - — Using similar sizes, shapes, or colors for related information.
  - — Leaving less space between elements of a screen than the space left at the margins.

Unity, illustrated in Figure 3.7, is coherence, a totality of elements that is visually all one piece. With unity, the elements seem to belong together, to dovetail so completely that they are seen as one thing. The opposite of unity is fragmentation, each piece retaining its own character. In screen design, similar sizes, shapes, and colors promote unity, as does *white space* — borders at the display boundary. Unity should exist between related screens, and Web site screens, as well.

Proportion

- Create windows and groupings of data or text with aesthetically pleasing proportions.

Down through the ages, people and cultures have had preferred *proportional* relationships. What constitutes beauty in one culture is not necessarily considered the same by another culture, but some proportional shapes have stood the test of time and are found in abundance today.

https://hemanthrajhemu.github.io

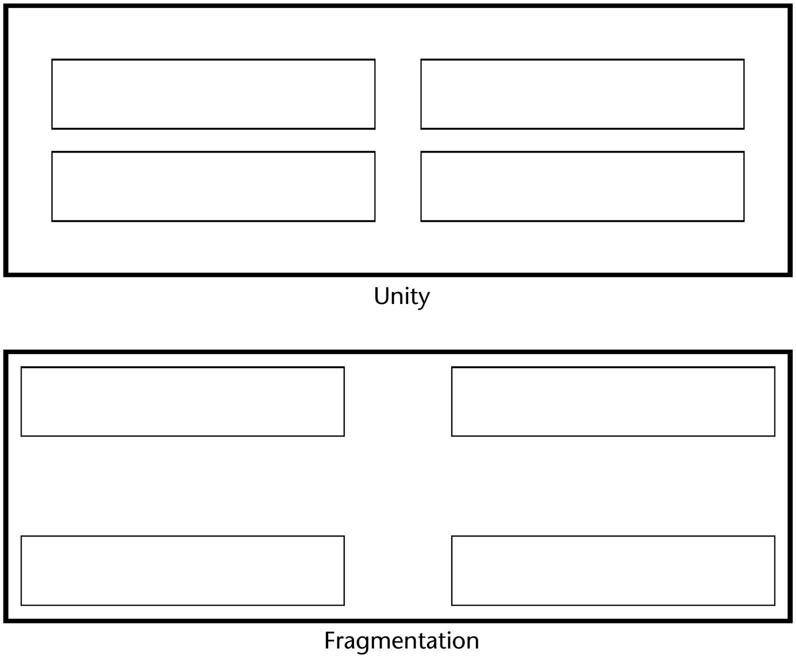


Figure 3.7: Unity (versus fragmentation).

Marcus (1992) describes the following shapes, illustrated in Figure 3.8, as aesthetically pleasing.

Square (1:1). The simplest of proportions, it has an attention-getting quality and suggests stability and permanence. When rotated it becomes a dynamic diamond, expressing movement and tension.

Square root of two (1:1.414). A divisible rectangle yielding two pleasing proportional shapes. When divided equally in two along its length, the two smaller shapes that result are also both square roots of two rectangles. This property only occurs with this proportion and is often used in book design. An open book has the same outside proportion as the individual pages within it. The square root of two has been adopted as a standard paper size in many countries of the world (the United States excluded).

Golden rectangle (1:1.618). An old (fifth century B.C.) proportion is the golden rectangle. Early Greek architecture used this proportion, and a mathematical relationship exists between this number and growth patterns in plant and animal life. This "divine division of a line" results when a line is divided such that the smaller part is to the greater part as the greater part is to the whole. The golden rectangle also has another unique property. A square created from part of the rectangle leaves a remaining area with sides also in the golden rectangle proportion.

Square root of three (1:1.732). Used less frequently than the other proportions, its narrowness gives it a distinctive shape.

Double square (1:2). In Japan, the tatami mat used for floor covering usually comes in this proportion. Rectangles more elongated than this one have shapes whose distinctiveness is more difficult to sense.

https://hemanthrajhemu.github.io

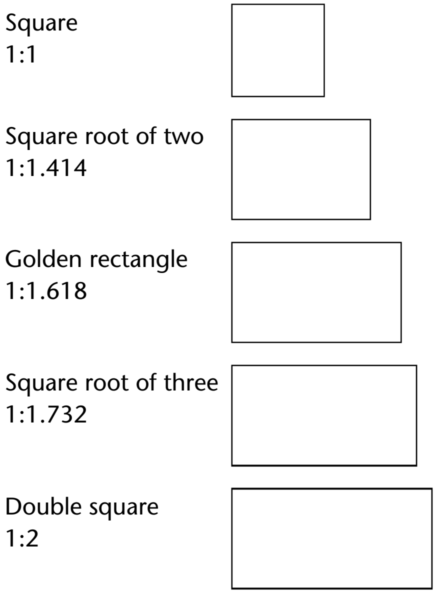


Figure 3.8: Pleasing proportions.

While these pleasing shapes have passed the test of time, not everything we encounter conforms to these principles. The American letter paper size has a ratio of 1:1.29, a typical American television screen has a ratio of 1:1.33, and CRT screens typically have ratios in the range of about 1:1.33 to 1:1.50.

In screen design, aesthetically pleasing proportions should be considered for major components of the screen, including window sizes, Web page sizes, graphics, and groupings of data or text.

Simplicity (Complexity)

- Optimize the number of elements on a screen, within limits of clarity.
- Minimize the alignment points, especially horizontal or columnar.
  — Provide standard grids of horizontal and vertical lines to position elements.

Simplicity, illustrated in Figure 3.9, is directness and singleness of form, a combination of elements that results in ease of comprehending the meaning of a pattern. The opposite pole on the continuum is complexity. The scale created may also be considered a scale of complexity, with extreme complexity at one end and minimal complexity (simplicity) at the other. While the graphics designer usually considers this concept as *simplicity*, we will address it as *complexity* since it has been addressed by this term in the design literature, where an objective measure of it has been derived.

This measure of complexity was derived by Tullis (1983) based on the work of Bonsiepe (1968), who proposed a method of measuring the complexity of typographically designed pages through the application of information theory (Shannon and Weaver, 1949). To illustrate, this measure involves the following steps:

https://hemanthrajhemu.github.io

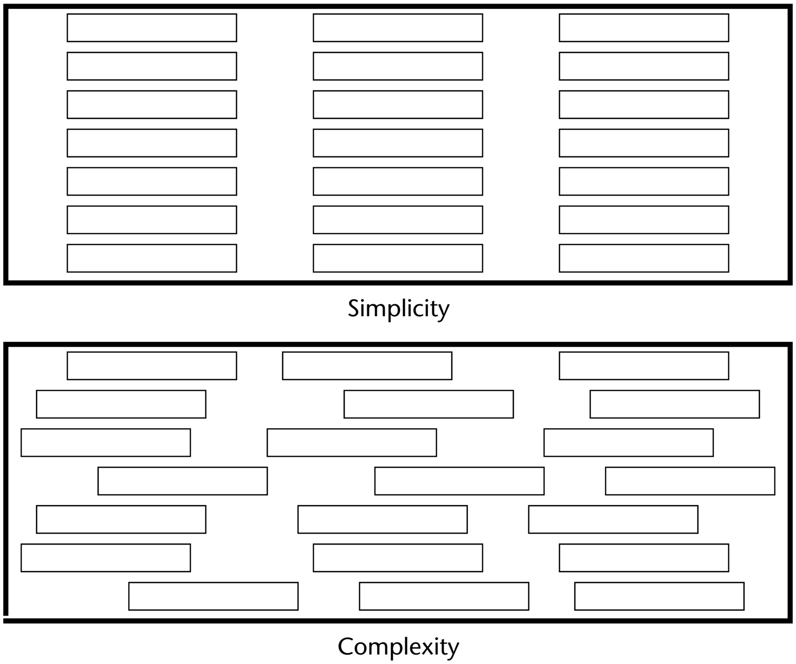


Figure 3.9: Simplicity (versus complexity).

1. Draw a rectangle around each element on a screen, including captions, controls, headings, data, title, and so on.
2. Count the number of elements and horizontal alignment points (the number of columns in which a field, inscribed by a rectangle, starts).
3. Count the number of elements and vertical alignment points (the number of rows in which an element, inscribed by a rectangle, starts).

This has been done for the text-based screens illustrated in Figures 3.10 and 3.11. These screens are examples from the earlier study by Tullis (1981) described in the introduction. They are an original read-only inquiry screen (Figure 3.10) from the screens whose mean search time was 8.3 seconds, and a redesigned screen (Figure 3.11) from the screens whose mean search time was 5.0 seconds.

A complexity calculation using information theory for each screen is as follows:

- Figure 3.10 (original):
  - 22 fields with 6 horizontal (column) alignment points = 41 bits.
  - 22 fields with 20 vertical (row) alignment points = 93 bits.
  - Overall complexity = 134 bits.
- Figure 3.11 (redesigned):
  - 18 fields with 7 horizontal (column) alignment points = 43 bits.
  - 18 fields with 8 vertical (row) alignment points = 53 bits.
  - Overall complexity = 96 bits.

https://hemanthrajhemu.github.io

The redesigned screen is, thus, about 28 percent simpler, or less complex, than the original screen.

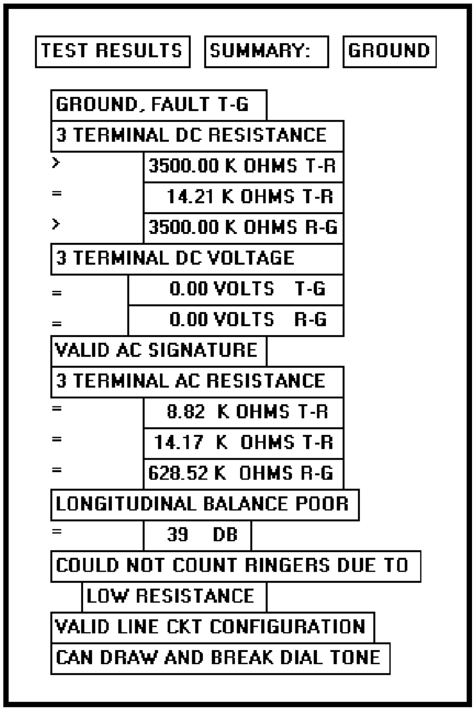


Figure 3.10: Original screen, from Tullis (1981), with title, captions, and data inscribed by rectangles.

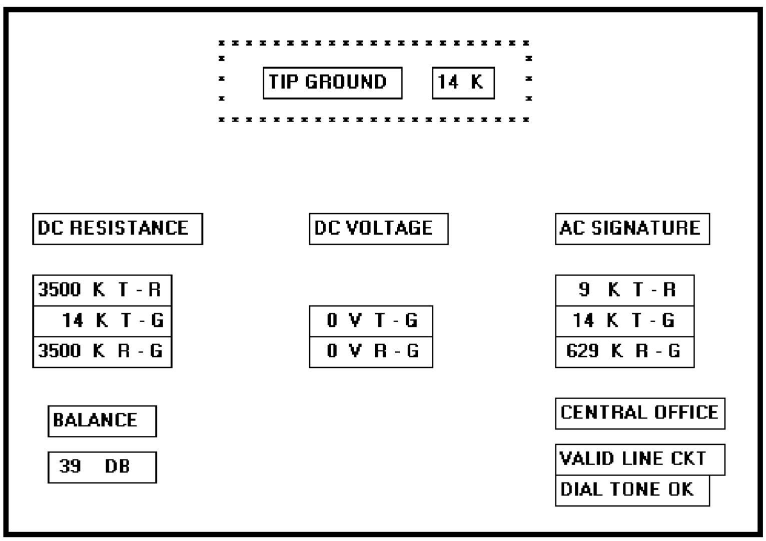


Figure 3.11: Redesigned screen, from Tullis (1981), with title, captions, and data inscribed by rectangles.

https://hemanthrajhemu.github.io

An easier method of calculation, however, yielding similar results, is to count the following: (1) the number of elements on the screen, (2) the number of horizontal (column) alignment points, and (3) the number of vertical (row) alignment points. The sums for the original and redesigned screens by this measure are:

- Figure 3.10 (original):
  - 22 elements
  - 6 horizontal (column) alignment points
  - 20 vertical (row) alignment points
  - 48 = complexity
- Figure 3.11 (redesigned):
  - 18 elements
  - 7 horizontal (column) alignment points
  - 8 vertical (row) alignment points
  - 33 = complexity

By this calculation the redesigned screen is about 31 percent simpler, or less complex, than the original screen.

By both calculations the redesigned screen has a *lower* complexity measure than the original screen. In the Tullis (1981) study, the redesigned and faster-to-use screens had lower complexity measures. This leads to the following complexity guidelines:

- Optimize the number of elements on a screen, within limits of clarity.
- Minimize the alignment points, especially horizontal or columnar.

Obviously, the way to minimize screen complexity is to reduce the number of controls displayed. Fewer controls will yield lower complexity measures. This is unrealistic, however, since ultimate simplicity means nothing is there, which obviously does not accomplish very much. Indeed, Vitz (1966) has found that people have subjective preferences for the right amount of information, and too little is as bad as too much. The practical answer, then, is to optimize the amount of information displayed, within limits of clarity. What is optimum must be considered in light of guidelines to follow, so a final judgment must be postponed.

Alignment. What can be done, however, is to minimize alignment points, most importantly horizontal or columnar alignment points. Fewer alignment points will have a strong positive influence on the complexity calculation. Tullis (1983) has also found, in a follow-up study of some other screens, that fewer alignment points were among the strongest influences creating positive viewer feelings of visually pleasing composition. Grose et al. (1998) also performed a study examining various screen factors and usability. They found that good alignment was related to shorter screen search times and higher viewer preferences for a screen.

Alignment of elements is a powerful means of leading a person through a display of information. Alignment also serves as a visual cue that elements of a display are related. Misalignments and uneven spacing, no matter how slight, can create bothersome unconscious disruptions to our perceptual systems. When things don't align, a sense of clutter and disorganization often results. In addition to reducing complexity,

https://hemanthrajhemu.github.io

alignment helps create balance, regularity, sequentiality, and unity. Alignment also reduces variation on a screen. Common alignments are associated with a group identity of their own. Anything that breaks that alignment is quickly seen as an exception.

In laying out a screen, imagine a grid of horizontal and vertical lines beneath it. Use these grids to position elements and to achieve common alignment points.

Groupings

- Provide functional groupings of associated elements.
- Create spatial groupings as closely as possible to five degrees of visual angle (1.67 inches in diameter or about 6 to 7 lines of text, 12 to 14 characters in width).
- Evenly space controls within a grouping, allowing 1/8 to 1/4 inch between each.
- Visually reinforce groupings:
  - — Provide adequate separation between groupings through liberal use of white space.
  - — Provide line borders around groups.
- Provide meaningful titles for each grouping.

Grouping screen elements aids in establishing structure, meaningful relationships, and meaningful form. In addition to providing aesthetic appeal, past research has found that grouping aids in information recall and results in a faster screen search. The study by Grose et al. (1998) found that providing groupings of screen elements containing meaningful group titles was also related to shorter screen search times. In this study groupings also contributed to stronger viewer preferences for a screen.

The perceptual principles of proximity, closure, similarity, and matching patterns also foster visual groupings. But the search for a more objective definition of what constitutes a group has gone on for years. Tullis, in his 1981 study, described an objective method for establishing groups based on the work of Zahn (1971) using the Gestalt psychologists' law of proximity. For the Tullis (1981) screens shown in Figures 3.12 and 3.13

1. Compute the mean distance between each character and its nearest neighbor. Use a character distance of 1 between characters adjacent horizontally and 2 between characters adjacent vertically (between rows).
2. Multiply the mean distance derived by 2.
3. Connect with a line any character pair that is closer than the distance established in step 2.

This has been done for these inquiry screens and the results illustrated in Figures 3.12 and 3.13. Boxes have been included around the derived groupings.

- Figure 3.12 (original):
  - Mean distance between characters = 1.05.
  - Twice the mean distance = 2.10.
  - A line is drawn between characters 1 or 2 apart, not 3 or more.
  - Resulting number of groups = 3.

https://hemanthrajhemu.github.io

- Figure 3.13 (redesigned):
  - Mean distance between characters = 1.09.
  - Twice the mean distance = 2.18.
  - A line is drawn between characters 1 or 2 apart, not 3 or more.
  - Resulting number of groups = 13.

Tullis found that the redesigned screens had more groupings than the originals.

To use a simplified version of this formula, connect with a line all characters on the screen separated by no more than one space horizontally and no blank lines vertically. Groupings will become immediately obvious.

Tullis also calculated another grouping measure: the average size of each screen's group. The average size of the three groups in the original screen is 13.3 degrees in visual angle, whereas the 13 groups on the redesigned screen average 5.2 degrees visual angle. The redesigned screen group size, interestingly, closely matches the 5-degree visual acuity screen chunk described in Step 1. It seems that groups 5 degrees or less in size can be scanned with one eye fixation per group. Therefore, screens with these size groupings can be searched faster. Groupings larger than 5 degrees require more eye fixations per grouping, slowing down screen scanning time. So, in addition to complexity, the Tullis redesigned screens differ from the original screens by some grouping measures. The more effective redesigned screens have a *greater number of smaller-size* groups.

Tullis, in his 1983 follow-up study, also found that groupings were the strongest determinant of a screen's visual search time. If the size of a group on a screen increased, or the number of groups increased, search time also increased. Number and size of groups have an opposite relationship, however; if the number of groups increases, their size usually decreases. If the size increases, the number of groups usually decreases. What proves to be most effective is a middle-ground solution, a medium number of medium-sized groups. The grouping guidelines described above are based upon this and other research presented.

Functional, semantic groups are those that make sense to the user. Related information should be displayed together. A logical place to "break" a screen is between functional groups of information, but a massive grouping of information should be broken up into smaller groups. The most reasonable point is every five rows. A six- or seven-row grouping may be displayed without a break, if necessary, but do not exceed seven rows.

The 11- to 15-character width limitation must take into consideration the data to be displayed. Confining data to this width makes no sense if it thus suffers a reduction in comprehensibility. Legibility and comprehension are most important.

To give unity to a display, the space between groups should be less than that of the margins. The most common and obvious way to achieve spacing is through white or blank space, but there are other ways. Alternatives include contrasting display features such as differing intensity levels, image reversals (white characters on a black background versus black characters on a white background), borders, and color. Spacing, however, appears to be stronger than color. Two studies (Haubner and Benz, 1983; Haubner and Neumann, 1986) found that adequate spacing, not color, is a more important determinant of ease of use for uncluttered, highly structured inquiry screens.

https://hemanthrajhemu.github.io

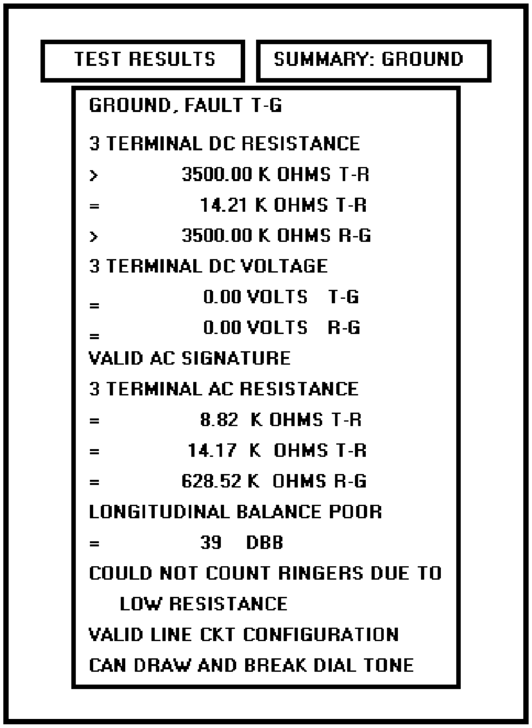


Figure 3.12: Original screen, from Tullis (1981), with grouping indicated by bold boxes.

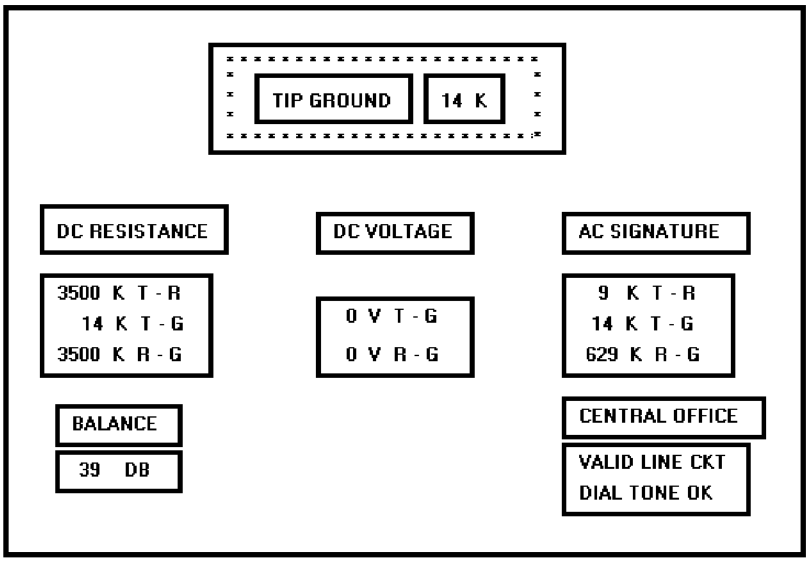


Figure 3.13: Redesigned screen, from Tullis (1981), with grouping indicated by bold boxes.

https://hemanthrajhemu.github.io

Perceptual Principles and Functional Grouping

■ Use visual organization to create functional groupings.				
— Proximity:	000 000 000			
— Similarity:	AAABBBCCC			
— Closure:	[ ] [ ] [ ]			
— Matching patterns:	>> < >			
■ Combine visual organization principles in logical ways.				
— Proximity and similarity:	AAA	BBB	CCC	
— Proximity and closure:	[ ]	[ ]	[ ]	
— Matching patterns and closure:	( )	< >	{ }	
— Proximity and ordering:	1234	1	5	
	5678	2	6	
		3	7	
		4	8	
■ Avoid visual organization principles that conflict.				
— Proximity opposing similarity:	AAA	ABB	BBC	CCC
— Proximity opposing closure:	] [	] [	] [	
— Proximity opposing ordering:	1357	1	2	
	2468	3	4	
		5	6	
		7	8	

Perceptual principles can be used to aid a person in comprehending groupings. Visual organization can establish a relationship between related items or design elements, as illustrated above.

Proximity. The most common perceptual principle used to establish visual groupings is the proximity principle. Elements positioned close together are perceived as a single group, and are interpreted as more related than elements positioned farther apart. A lack of proximity creates the impression of multiple groups and reinforces differences among elements. In the preceding example, the incorporation of adequate spacing between groups of related elements enhances the "togetherness" of each grouping. Space should always be considered a design component of a screen. The objective should never be to get rid of it.

Similarity. The similarity principle can be used to call attention to various groupings by displaying related groupings in a different manner, such as intensity, font style, or color. Elements that are similar in some manner are perceived to be more related than elements that are dissimilar.

Closure. Because people tend to perceive a set of individual elements as a single recognizable pattern instead of a collection of multiple, individual elements, users will close gaps and fill in missing information to derive a meaningful pattern whenever possible. Closure is strongest when elements approximate simple and recognizable patterns. Closure, generally, will not occur if the effort required to identify a form or pattern is greater than effort required to perceive the elements individually. In the preceding example, the perception of boxes is established through the use of differential spacing.

https://hemanthrajhemu.github.io

Matching Patterns. The matching patterns principle involves using lines, borders, and unique symbols to identify and relate common information.

Ordering. Certain elements, such as numbers and the alphabet, possess a very well-learned logical ordering scheme.

Combine visual organization principles in logical ways. Visual organization principles can be combined to reinforce groupings. Proximity, a very strong perceptual principle, can guide the eye through an array of information to be scanned in a particular direction. Scanning direction can also be made obvious through similarity (color, intensity, and so on) or matching patterns (lines or borders).

Avoid visual organization principles that conflict. Principles may not always be compatible, however. When the viewer encounters incompatibilities as previously illustrated, confusion results. In the preceding examples, proximity destroys similarity, proximity overwhelms closure, and proximity overwhelms logical ordering.

The aforementioned perceptual principles exert a strong influence on perceived groupings of screen elements. Beck and Palmer (2002) found that in establishing groupings

- People find it very difficult to ignore color similarity and a common ground.
- Proximity has a stronger influence than a common ground.
- Color similarity has a stronger influence than proximity or a common region.
- Color, shared background, and co-location are stronger grouping cues than outlines.

Grouping Using White Space

- Provide adequate separation between elements through liberal use of white space.
- For Web pages, carefully consider the trade-off between screen white space and the requirement for page scrolling.

Today the term white space is a misnomer, a carry-over from the white paper of printed forms. It might more appropriately simply be called space, areas without text, graphics, and so forth. Whatever it is called, white space should be considered a screen element of equal importance to all others. It is not wasted space. It defines and separates screen elements, and gives a screen proportion and meaningful form, thus assisting in providing the distinctiveness that is so desired. Space is also used to direct attention to adjacent areas that do contain important information. Remember, if a screen is perceived as a homogeneous, cluttered mass, information will only be found through an exhaustive search of the entire screen.

In Web page design the "provide white space" principle directly confronts another principle, "minimize the need for scrolling" (to be discussed shortly). Web pages are typically longer than the maximum visible area of the display on which they are presented. To see the entire page requires screen scrolling. An abundance of white space, therefore, creates a need for excessive scrolling. It also makes page scanning more difficult and moves information further apart spatially, making the determination of information relationships more difficult.

https://hemanthrajhemu.github.io

Web page guidelines addressing this problem are somewhat contradictory. Recommendations include "use white space moderately" (Koyani et al., 2004), "minimize the use of white space in search tasks" (Bailey, 1999c), sites with large amounts of white space and sparse text are rated poorly (Festa, 1998), and finally users prefer a moderate amount of white space and there is no reliable performance difference with different amounts of white space (Bernard et al., 2000).

On the other hand, white space surrounding an element was found to direct attention and enhance recall (Olsen, 2002), use of white space between paragraphs and in the left and right margins increased comprehension by almost 20 percent (Lin, 2004), and a characteristic of the top 100 Web sites is "lots of white space" (Grok.com).

The most practical recommendation at the moment is that for longer Web pages the use of white space should be reduced, but certainly not at the expense of page clarity. Each page must be evaluated in terms of its specific content, the relationships existing between the various pieces of content, and the page's scanning and visual search requirements. A trade-off that best meets all these needs must then be established and implemented.

Grouping Using Borders

- Incorporate line borders for
  - Focusing attention on groupings or related information.
  - Guiding the eye through a screen.
- Do not exceed three line thicknesses or two line styles on a screen, however.
  - Use a standard hierarchy for line presentation.
- Create lines consistent in height and length.
- Leave sufficient padding space between the information and the surrounding borders.
- For adjacent groupings with borders, whenever possible, align the borders left, right, top, and bottom.
- Use rules and borders sparingly.
- In Web page design
  - Be cautious in using horizontal lines as separators between page sections.
  - Reserve horizontal lines for situations in which the difference between adjacent areas must be emphasized.

Line borders. Line borders, or rules, can greatly enhance a screen. Research has found that information displayed with a border around it is easier to read, better in appearance, and preferable. Figures 3.14 and 3.15 illustrate identical screens with and without borders around groupings. While many groupings are obvious without borders, borders certainly reinforce their existence.

Lines or rules assist in focusing attention on related information. They also aid in separating groupings of information from one another. Draw borders around elements to be grouped. Microsoft Windows provides a control called the *Group Box* to establish a frame around a group of related controls (see Step 7).

https://hemanthrajhemu.github.io

Rules also serve to guide the viewer's eye in the desired direction. Use vertical rulings to convey to the screen viewer that information is organized vertically and should be scanned from top to bottom.

Line thickness variations. Too many variations in line thickness on a screen create clutter and are distracting. Use no more than three line weights at one time, or two different styles. Use a standard hierarchy for rules, the thickest to differentiate major components, the thinnest for minor separation. Consider a thin border for individual controls, a slightly thicker border for groupings, and the thickest borders for windows themselves.

Consistent line widths and heights. Similarly, variations in line widths and heights are distracting. Create horizontal lines of equal widths across the screen and vertical lines of equal height whenever possible. This will provide better balance.

Sufficient space padding. When placing information within borders leave "breathing room," sufficient space between the information and the borders themselves. Avoid looking like your 6 pounds are being stuffed into a 5-pound sack.

Alignment. For adjacent groupings with borders, whenever possible, align the borders left, right, top, and bottom. The most important alignment points are left and top. Do not create right and bottom alignment by leaving excessive white space within the area encompassed by the border. This is not visually appealing.

Use lines and borders sparingly. Too many lines and borders on a screen also create clutter and can be distracting. Like any display technique, lines and borders must be used sparingly.

Web pages. In Web page design, be cautious in using horizontal lines as separators between page sections. Users may assume they have reached the bottom of the page, missing what follows. Separator lines may also make the screen look more cluttered. In general, reserve horizontal lines for situations where the difference between adjacent areas must be emphasized.

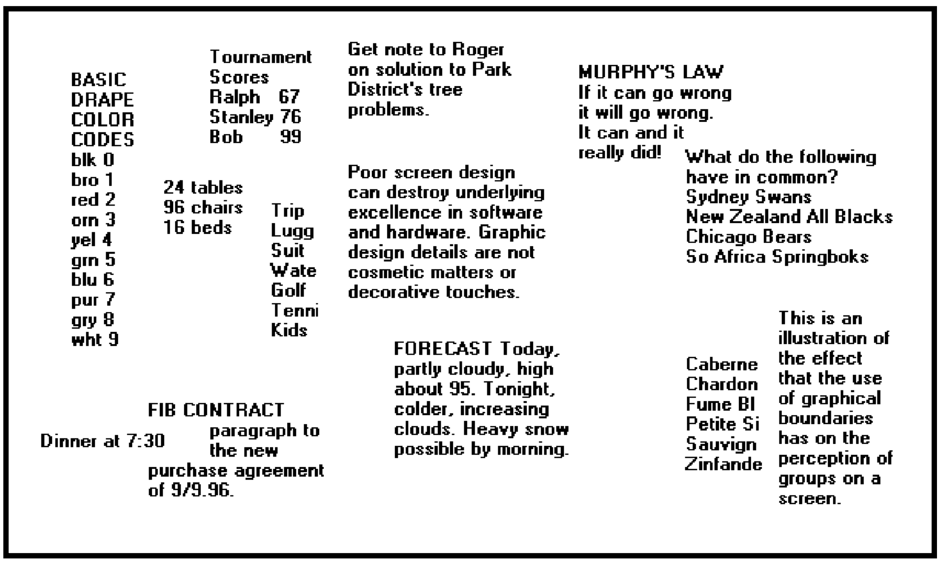


Figure 3.14: The effect of line or graphical borders. Groupings without borders.

https://hemanthrajhemu.github.io

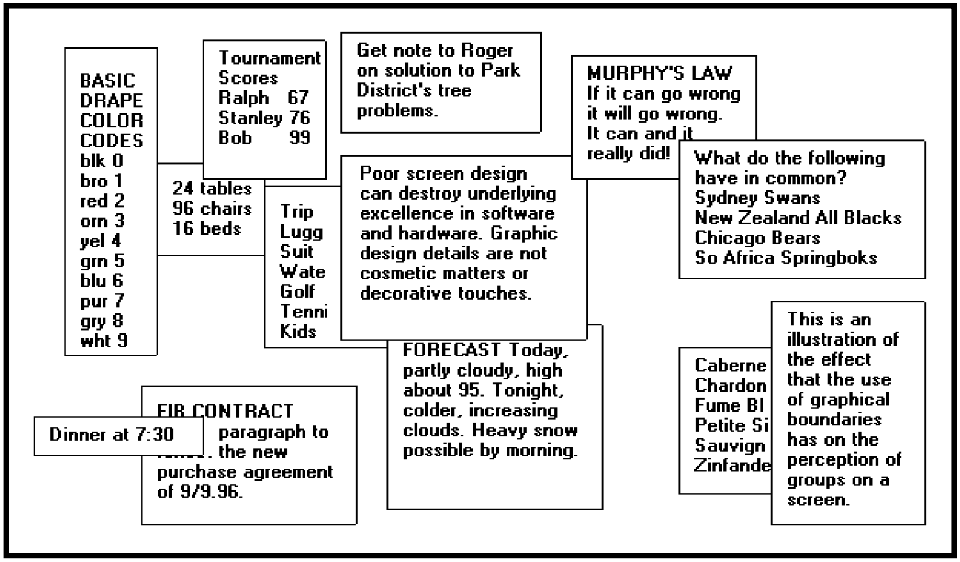


Figure 3.15: The effect of line or graphical borders. Groupings with borders.

Grouping Using Backgrounds

- Consider incorporating a contrasting background for related information.
  — The background should not have the "emphasis" of the screen component that should be attended to. Consider about a 25 percent gray screening.
  — Reserve higher contrast or "emphasizing" techniques for screen components to which attention should be drawn.

Information can also be visually tied together through using a background that contrasts with the remainder of the screen. The background should just be a background, however; visual emphasis should be directed toward foreground material. A common failing of many screens is that the background is too highly emphasized. Consider about a 25 percent gray screening. Always reserve the higher contrast or emphasizing techniques for screen components in need of attention. Be very conservative in the variety of different backgrounds used. Background colors may also be used to relate or separate screen groupings. Color by itself is a poor separator of screen elements, however. A border is always needed to properly set off adjacent areas of different colors. Colors should also be used with caution since the user may have the ability to change them. This may result in undesirable effects. Finally, less variation is always better than more. An additional discussion on color screen backgrounds is found in Step 12.

MAXIM **Working with a system should never be painful; instead it should be so painless you forget what you are doing!**

https://hemanthrajhemu.github.io

Visual Style in Web Page Design

- Maintain a consistent and unified visual style throughout the pages of an entire Web site.
- Base the visual style on
  — The profile and goals of the Web site owner.
  — The profile, tastes, and expectations of the Web site user.

The style of a Web site, its visual characteristics, including color, typography, and graphics; the shape of its design elements; and the relationship of its components to one another, including their locations, should be maintained throughout the entire Web site. This will provide unity and harmony to the Web site and give it a consistent identity. Unity and consistency will also aid users in navigation, reinforcing for them that they are anchored in a specific place within the universe of information space. It will also enable the users to become comfortable and familiar with the Web site.

Visual style must reflect the needs and goals of the Web site owner and will vary depending upon the Web site's purpose: presenting facts, marketing, entertainment, and so forth. The tastes and expectations of the Web site's users are also critically important. It must be appealing to its expected viewers, motivating them to explore its entire contents. Sun Microsystems (1998) in a redesign of their Web site reported that a great visual appearance made users think more highly of their site. Visual elements of a Web site, including layout, use of color, and fonts, influence a site's credibility and quality ratings (Fogg et al., 2002; Ivory and Hearst, 2002). The visual design style must always be evaluated along with other components of the system. (See Step 14.)

Distinctiveness

- Individual screen controls, and groups of controls, must be perceptually distinct.
- Screen controls
  — Should not touch a window border.
  — Should not touch each other.
- Field and group borders
  — Should not touch a window border.
  — Should not touch each other.
- Buttons
  — Should not touch a window border.
  — Should not touch each other.
- A button label should not touch the button border.
- Adjacent screen elements must be displayed in colors or shades of sufficient contrast with one another.

https://hemanthrajhemu.github.io

Elements of screen must be distinct, clearly distinguished from one another. Distinctiveness can be enhanced through separation and contrast. Distinctiveness is achieved by providing adequate separation between adjacent elements and screen boundaries and providing adequate separation between parts of an element. Screen controls, field and group borders, and buttons should not touch window borders or each other. Colors or shades used for adjacent screen elements must also contrast well with one another. Guidelines for color and shading are described in Step 12.

Focus and Emphasis

- Visually emphasize components such as
  — Most prominent elements.
  — Most important elements.
  — Central idea or focal point.
  — Changing elements.
  — Most critical elements.
- To provide emphasis use techniques such as
  — Higher brightness.
  — Reverse polarity or inverse video.
  — Distinctive Typeface.
    - Bold.
    - Italics.
    - Underlining.
  — Blinking.
  — Line rulings and surrounding boxes or frames.
  — Color.
  — Larger size.
  — Animation.
  — Positioning.
  — Distinctive or unusual shape.
  — Isolation.
- De-emphasize less important elements.
- To ensure that emphasized screen elements stand out, avoid
  — Emphasizing too many screen elements.
  — Using too many emphasis techniques.
- Minimize screen clutter.
- In Web page design
  — Call attention to new or changed content.
  — Ensure that page text is not overwhelmed by page background.

Apply a visual emphasis technique to draw attention to the most important, prominent or critical elements of a screen. An emphasized element should contrast with the rest of the screen, calling the user's eyes to it. Olsen (2002) found that abrupt environmental changes (or edges) capture a person's attention. Examples include color transitions or white space surrounding a page element. Wu and Yuan (2003) evaluated the

https://hemanthrajhemu.github.io

effectiveness of highlighting techniques in aiding comparing and finding discrepancies in a pair of data in a matrix. The highlighting techniques of blinking, reverse video, and a different color (red) were compared with no highlighting. Participants were significantly faster at identifying discrepancies when they were highlighted. Highlighting, to be effective, of course, must be appropriate to the task. Silvers and Keiner (1997) found that inappropriate highlighting of text resulted in lower reading comprehension scores. Visual emphasis, however, should always be used sparingly if it is to be effective.

Brightness. A brighter element has a good attention-getting quality and no disturbing features. It may be used to indicate items in error, and increased brightness is the best vehicle for calling attention to data on inquiry screens. Do not use more than two brightness levels on a screen. If brightness has a fault, it is that displays with improperly set manual screen contrast controls can diminish its effectiveness, even causing it to disappear. This can be a major problem for displays placed in exceptionally bright viewing conditions.

Reverse polarity. Inverse video reverses an element's polarity. For example, on a screen with a specific level of dark text and a specific level of light background, certain elements can be emphasized by displaying them in the reverse: a text of the same lightness as that used on the regular screen on a background of the same darkness as is used on the regular screen. For old text-based screens, reverse polarity meant displaying dark text on a light background, or reversing the standard light text on dark background. During the 1980s several studies comparing this reverse were performed and, in general, no differences in performance, eye-scanning behavior, or feelings of visual fatigue were uncovered. One study, however, did find reverse polarity more visually fatiguing, while another found green and orange phosphor reverse polarity screens easier to read but found no differences in white phosphor readability. These studies contributed to the popularity of orange phosphor screens at that time. In those days, it was generally found that people preferred dark background screens to white background screens because of the harshness and excessive brightness of backgrounds. There was one benefit to light background screens, however. Dark background screens can create a viewing problem, their mirror-like surfaces reflecting light from other outside sources back into the screen viewer's eyes. Light background screens absorb most of this light, instead of reflecting it back to the screen viewer.

For elements of screens — pieces of data, messages, and so on — reverse polarity has a very high attention-getting quality. It can be effectively used for items selected, items in error, information being acted upon, or information of current relevance. Some cautions should be taken with reverse polarity. If reverse polarity is used to identify one kind of element, such as a text box or other boxed control, avoid what can best be described as the crossword puzzle effect — the haphazard arrangement of elements on the screen creating an image that somewhat resembles a typical crossword puzzle. An arrangement of elements might be created that tries to lead the eye in directions that the designer has not intended or causes elements to compete for the viewer's attention. The cause of this problem is using reverse polarity for too many purposes or by poor alignment and columnization of elements selected for this emphasis. Conservative use

https://hemanthrajhemu.github.io

and alignment and columnization rules will minimize this effect. If reverse polarity is used to highlight information such as messages or actions to be taken, allow an extra reversed character position on each side of the information. This will leave a margin around the information, improving legibility and giving it a more pleasing look. Last, reverse polarity can make text harder to read if the screen resolution and character sizes are not sufficient. A light screen background can actually bleed into its dark characters, reducing their legibility. This is a phenomenon called iridescence.

Distinctive typeface. Differences in fonts have a moderate attention-getting capability. Their varying sizes and shapes can be used to differentiate screen components. Larger, bolder letters can be used to designate higher-level screen pieces, such as different levels of headings, if the headings are used to search for something. Do not use larger fonts, however, for entry/modification (conversational) and display/read-only screens, because this will place too much emphasis in the headings themselves. Emphasis should go to the screen data. If you are using multiple fonts, never use more than two styles or weights, and three sizes, on a screen. As a highlighting technique, detectable difference between the fonts will be difficult to achieve.

Bolding, Italics, and Underlining. These techniques can be used to provide subtle differences in screen elements. *Bolding* adds minimal noise to a design and highlights important elements and is generally the preferred technique. *Italics*, while adding minimum noise, is less detectable and legible. *Underlining* is a moderate attention-getting mechanism, but it adds noise and can reduce legibility, so it should be used conservatively and carefully. In graphical systems it is commonly used to designate keyboard equivalents or mnemonics. In Web pages it is now used to designate navigation links. It should only be used for that function.

Blinking. Blinking has a very high attention-getting capability, but it reduces text readability and is disturbing to most people. It often causes visual fatigue if used excessively. Therefore, it should be reserved for urgent situations and times when a quick response is necessary. A user should be able to turn off the blinking once his or her attention has been captured. The recommended blink rate is 2 to 5 Hz, with a minimum "on" time of 50 percent. An alternative to consider is creating an "on" cycle considerably longer than the "off," a wink rather than a blink.

Line rulings and surrounding boxes or frames. Use lines to emphasize and guide the user's eye through the screen. Use horizontal rulings as a substitute for spaces when breaking a screen into pieces. Use vertical rulings to convey to the screen viewer that a screen should be scanned from top to bottom. Use rules to surround radio button and check box controls, and other groupings of controls or important single controls. While many groupings are obvious without borders, borders certainly reinforce their existence. Use no more than three line thicknesses or two line styles on a screen.

Colors. Use color to emphasize and assist in the identification of screen components. Some colors appear brighter than others. Parts of images or text comprised of brighter colors seem to gain focus first. Presenting some elements in brighter colors and others in darker colors aids people in determining their relative importance (Koyani et al., 2004). Display no more than four colors at one time on

https://hemanthrajhemu.github.io

a screen essentially alphanumeric in nature, six on a statistical graphics screen. Color considerations are discussed in detail in Step 12.

Size. Larger objects draw attention before smaller ones. People will fixate on larger items first, but may skip certain kinds of images that are thought to be advertisements or decoration (Koyani et al., 2004).

Animation. Movement is the most effective attention-getter. It draws one's eyes. However, movement can be very annoying if it is not relevant or useful. Movement can also distract from other information if it continues after the user's attention is captured (Koyani et al., 2004).

Positioning. Placing an element in a position where the eye first meets the screen can capture attention.

Distinctive or unusual shape. The eye is also drawn to distinctive or unusual shapes.

Isolation. White space around the highlighted items tends to increase their prominence (Olsen, 2002). One's eyes will also be drawn to the start of any text following white space.

De-emphasize less important elements. To designate an element as not applicable or not active, dim it or gray it out.

Avoid too much emphasis. An emphasized element must, of course, attract the user's eye to it. The attraction capability of a mechanism is directly related to how well it stands out from its surroundings. Emphasis will lose its attracting value if too many different items on a screen are emphasized. A few hands raised for attention are much easier to deal with than many raised hands. Focus problems will also be created if too many emphasizing techniques are used within a screen. The user's attention will be drawn to the differences in techniques, and his or her information processing system will try to understand why the differences exist, instead of focusing on the information itself. Minimization of clutter also assists a user in focusing on the most crucial part of a screen. In using emphasis, conservatism and simplicity is the key.

Web page emphasis. The dynamic nature of the Web and its available screen design tools raise some other emphasis considerations. New or changed Web page content should be emphasized to immediately call the user's attention to it when a page is presented. Inappropriate page backgrounds may degrade an emphasis technique's usability. Background graphics, pictures, patterns, or textures may reduce the technique's attention-getting quality, as well as reduce text legibility.

Conveying Depth of Levels or a Three-Dimensional Appearance

- Use highlighting, shading, and other techniques to achieve a three-dimensional appearance.
- Always assume that a light source is in the upper-left corner of the screen.
- Display command buttons above the screen plane.

https://hemanthrajhemu.github.io

- Display screen-based controls on, or etched or lowered below, the screen plane.
- Do not overdo perspective and avoid
  — Using perspective for noninteractive elements.
  — Providing too much detail.

People have learned to perceive many objects in the visual world as three-dimensional, even when they are obviously not actually presented as three-dimensional. On screens, certain techniques can be used to foster the perception of three dimensions. These techniques include the following (Marcus, 1992; Lidwell et al., 2003):

Overlapping. Fully display the window or screen element of current relevance and partially hide beneath it other screen windows or elements, as illustrated in Figure 3.18. The completeness or continuity of outline of the relevant element will make it appear nearer than those partially covered.

Drop shadows. People have a tendency to interpret dark or shaded areas of objects as shadows resulting from a light source above the objects. Lidwell et al. (2003) call this *Top-Down Lighting Bias* and suggest that it likely results from humans having evolved in an environment lit from above by the sun. This bias exists in all age ranges and cultures. In creating shadows, always assume that the light source is in the upper-left corner of the screen. To further aid in the perception of the placement of a pull-down above a screen, or a window above a screen or another window, locate a heavier line along the bottom and right edges of the pull-down or window. This creates the impression of a shadow caused by a light source in the upper-left corner of the screen, reinforcing the nearness of the important element. The light source should always appear to be upper left, the shadow lower right.

Highlighting and lowlighting. Highlighted or brighter screen elements appear to come forward, while lowlighted or less bright elements recede. Attention will be directed to the highlighted element.

Shrinking and growing. Important elements can be made to grow in size, while less important elements remain small or shrink. An icon, for example, should expand to a window when it is selected. The movement, as it expands, will focus attention upon it.

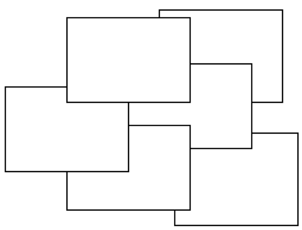


Figure 3.16: Overlapping screen elements.

https://hemanthrajhemu.github.io

Beveled edges. A beveled edge (lines that are not at right angles to the screen element borders) will also give the impression of depth. With beveled edges, windows, buttons, and menu bar choices will appear to rise from the screen. To strengthen the three-dimensional aspect of the screen element, give it a drop shadow by shading the bottom and right sides with either a tone of gray or a darker shade of the basic screen color.

Texture change. Texture is the surface quality of an object. Varying the object's pattern of light and dark areas creates it. Increased density of an object implies a further distance, less density a closer distance. Increase the density of nonapplicable screen elements, and display currently relevant elements less densely. If textures are used as a code on screens, Shurtleff (1993) recommends using no more than six or seven. Also, a texture change should convey information that is not immediately apparent from name, shape, or other physical characteristic of an object. Finally, provide consistency; establish only one meaning for a texture.

Color change. Objects farther away appear hazy and less saturated. Increase haziness as screen element importance diminishes; display currently relevant elements more vividly.

Size change. Objects farther away appear smaller. Decrease the size of nonapplicable screen elements; display currently relevant elements as larger. The size of familiar objects can also be used to provide a clue to indicate the size and depth of unfamiliar objects.

Clarity change. Objects that are bluer and blurrier or fuzzier are interpreted as being farther away; objects less blue and blurrier or fuzzier are seen as being closer. Display nonapplicable elements as blurred, and currently relevant screen elements as clear.

Vertical location. When two objects are presented at different vertical locations, the higher object appears higher, the lower object closer. Present currently applicable screen elements at the bottom of the screen; present nonapplicable elements at the screen's top.

Spacing change. Faraway objects appear more closely spaced, closer objects more widely spaced. Display nonapplicable elements as more closely spaced, currently applicable screen elements as more widely spaced.

Receding lines. Parallel lines converging and receding to a vanishing point imply depth.

Motion change. Objects moving at uniform speeds appear to be moving more slowly the farther away they are.

To visually communicate function, consider displaying command buttons raised above the screen. Conversely, display screen-based controls on, or etched or lowered below, the screen plane. Consistently follow this concept on all screens. One caution: Do not overdo perspective or the effect will be lost, and visual clutter will emerge. Also, avoid using perspective for noninteractive elements, and do not provide too much detail.

https://hemanthrajhemu.github.io

Presenting Information Simply and Meaningfully

- Provide legibility.
  — Information is noticeable and distinguishable.
- Provide readability.
  — Information is identifiable, interpretable, and attractive.
- Present information in usable form.
  — Translations, transpositions, and references to documentation should not be required to interpret and understand information.
- Utilize contrasting display features.
  — To attract and call attention to different screen elements.
- Create visual lines.
  — Implicit and explicit, to guide the eye.
- Be consistent.
  — In appearance and procedural usage.

Following are guidelines for presenting information on screens. The fundamental goals are clarity and simplicity in form, comprehensibility in organization, efficient information assimilation, and pleasantness in tone.

Legibility. Legibility is distinguishableness. Computer technology today presents a seemingly endless array of choices in such aspects as font styles, sizes, and weights. Is the type of the proper kind and of adequate size and clarity for viewers of all ages? Is the contrast between text and its background adequate? While greatly improved in the past few decades, in general, the legibility of screen text still does not always match that of text presented on paper.

Readability. Readability is the degree to which prose can be understood. It is based on the complexity of words and sentences. Readability is established by factors like the length and commonality of words used, sentence length, and the number of syllables and clauses contain within a sentence. In design, is the information written at an understandable level for all users? Is it direct, simple, and easy to comprehend? Is visual interference minimized? Guidelines for readability will be described in Step 8.

When people read, they use the shape of a word as a strong aid in comprehension; often people do not read individual letters but recognize word shapes. Words are given more distinctive shapes by letter "ascenders" and "descenders." A lowercase letter's height is called its "x" height (the height of the small letter x). Other letters are identical in height to the "x," such as the "e," "a," and "n" in the word "explain." Ascenders are letter strokes that rise above the x — the tops of the "l" and "i" in "explain," for example. Descenders are letter strokes that drop below the x — the bottom of the "p" in "explain." Research indicates that in the reading process, the top half of a letter is the most important part of a word in comprehension. The top halves of letters are more distinctive than their bottom halves.

https://hemanthrajhemu.github.io

Usability. Screen information should be presented in a directly usable form. Reference to documentation or other extra steps for interpretation should never be required. In graphical system design, content consisting of words and text is much faster to comprehend and use than content in a graphical form.

Contrasting display features. Use contrasting display features to call attention to different screen components, items being operated upon, or urgent items. Usable features include such things as letter style, size, and color. Features chosen should provide perceptual cues to aid in screen component identification so that attention may be quickly and accurately focused. Perceptual cues clarify structure and relationships, and give hints to the reader. Good readers make great use of the typographic and semantic cues found in well-presented text.

Visual lines. The eye should be guided vertically or horizontally implicitly through the screen through the use of white space and content, typefaces, and control and text alignments. In situations where a large amount of information must be presented on one screen, eye movement direction may also be communicated to the viewer explicitly, through the drawing of actual vertical or horizontal rules. Purposeless, unfettered wandering of the eye should be discouraged.

Consistency. Methods chosen to present information must, of course, always be consistent in visual appearance and procedural usage.

Typography

In typography, by definition a typeface is the name of a font type, such as Times New Roman, Arial, Verdana, or Helvetica. A font has several qualities, including size (Times New Roman 16-point or Arial 12-point) and other characteristics, including case (upper, lower, and mixed), type (serif and sans serif), and styles such as bold, italic, outline or shadow.

In screen design, the terms typeface and font have become somewhat interchangeable. In this discussion, the term font will be used to encompass both types and other font characteristics. A font's characteristics can be used as a tool to

- Communicate the organization of screen elements.
- Identify the most important screen elements.
- Establish a reading order.
- Create a particular mood.

A very large supply of fonts is available for these purposes. Over the past several decades numerous studies have been performed in an effort to establish the font, or fonts, with the best legibility for use on computer systems. Absolute definitive conclusions have always been hindered by the complexity of the issues involved. For example (Koyani, 2004; Rubinstein, 2002):

- Defining text size using pixels will result in differently-sized characters depending upon on the monitor's pixels and its set resolution.
- Windows Web browsers display type 2 to 3 points larger than the same font displayed on a Macintosh.

https://hemanthrajhemu.github.io

- Point size is not comparable between different fonts.
- User viewing distances vary. Generally, the larger the physical display, the farther away people view it. The same size font may then look smaller or larger depending on viewing distance.

These kinds of problems make it difficult to evaluate any single parameter. The guidelines that follow, then, are a compilation that includes the general conclusions of the font research.

Font Types and Families

- Use simple, common, and familiar fonts to achieve the best reading speed.
  — Arial or Verdana Sans Serif.
  — Times New Roman or Georgia Serif.
  — Avoid specialty or "cool" fonts.
- Use no more than two families, compatible in terms of line thicknesses, capital letter height, and so on.
  — Assign a separate purpose to each family.
  — Allow one family to dominate.

Visually simple, common, readable fonts are needed for clarity on most screens, including business system applications and the text content of Web pages. A variety of studies have addressed the reading speeds of various font types, both serif and sans serif. (Serifs are the small cross strokes that appear on the arms of some letters. Letters without the cross strokes are referred to as san serif.) The results of some of these studies can be summarized as follows:

- Tullis et al. (1995) compared MS Sans Serif, MS Serif and Arial (sans serif) in a proofreading task and found no difference between the serif and sans serif fonts. Subjects did prefer the sans serif fonts to the serif font, however.
- Boyarski et al. (1998) compared the reading speeds of people using Georgia (serif), Times New Roman (serif) and Verdana (sans serif) and found no reliable performance differences in reading speeds for comprehension.
- Bernard and Mills (2000) compared Arial (sans serif) and Times New Roman (serif) and found no reliable differences in reading speed or in the detection of word errors.
- Bernard et al. (2001a) compared eight fonts, the serif fonts of Century Schoolbook, Courier New, Georgia, and Times New Roman and the sans serif fonts of Arial, Comic Sans, Tahoma, and Verdana. They found that Arial and Times New Roman were read reliably faster than Courier, Schoolbook and Georgia. All the fonts except Century Schoolbook were reliably preferred over Times New Roman.
- Bernard et al. (2002b), in a follow-up study, added Goudy Old Style (serif) and Agency (sans serif) to the previous eight fonts and found no reliable differences in reading efficiency for any font. Arial, Verdana, and Comic Sans were preferred, however.

The sans serif font of Verdana, designed especially for use on screens, is considered more contemporary. Times New Roman possesses very small serifs, and it is considered

https://hemanthrajhemu.github.io

more conservative and traditional. Fonts with serifs, it is also felt, provide better links between letters in a word, provide a horizontal guideline for the eye, and help in distinguishing one letter from another. What can we conclude from the referenced studies?

- There appears to be no definitive differences in reading speeds for adults when using the common fonts of Arial, Verdana, Georgia, and Times New Roman.
- Most people seem to prefer the sans serif fonts of Arial and Verdana.

It is also recommended that sans serif fonts be used if the type is 10 or fewer points in size, if the display environment is less than ideal, or if the screen is of low resolution. The serifs can wash out under these conditions. Ornate, specialty, or "cool" fonts should be avoided because they are not familiar and may degrade on the screen, reducing legibility.

All typefaces are named, some after their designers (for example, Garamond), some after locations where they originated (Helvetica after Switzerland), and other for the publications for which they were designed (Times). Within each typeface are style variations: regular, italics, boldface, outlines, and shadows. These groupings are called *families*.

A family of styles is designed to complement one another, creating unity in design. An example of a family is that of Times illustrated in Figure 3.17.

Similar typefaces are grouped into what are called *races*. One kind of race is called *roman*, which contains the Times typeface illustrated here as well as the Bookman, Schoolbook, and Palatino typefaces. *Sans serif* is also a race where the typefaces Arial, Verdana, Helvetica, and Avant Garde reside. Another race is named *Old English*. An effective design can almost always be achieved by staying within one typeface race. If it is necessary to mix typeface families on a screen do the following:

- Never mix families within the same race. Typographic noise is created.
- Assign a separate purpose to each family. A sans serif typeface for the title and headings and a roman typeface for the body is a good combination. The opposite is also acceptable.
- Allow one family to dominate.

Times Roman

Times Italic

Times Bold

Times Bold Italic

Times Outline

Times Shadow

Figure 3.17: The Times family of type.

https://hemanthrajhemu.github.io

Professional designers rarely use anything but the traditional, simple typefaces. These fonts have been crafted through centuries of use. When in doubt, use the same font and its variants for all design needs. This is better than a risky aesthetic decision.

Font Size

- Use no more than three sizes.
  — Consider "x" height.
- For graphical systems use
  — 12 point for menus.
  — 10 point for windows.
- For Web pages use
  — 12 to 14 points for body text.
  — 18 to 36 points for titles and headings.
- For line spacing use one to one and one-half times font size.
- Never change established type sizes to squeeze in more text.

Font sizes are described by points — the distance between the top of a letter's ascender and the bottom of its descender. One point equals 1/12 inch. Font sizes should be large enough to be legible on standard monitors. Also, a typeface displayed on a Windows Web browser will look 2 to 3 points larger than the same typeface on a Macintosh. Recent font size research has found the following:

- Tullis et al. (1995) in their study compared font sizes ranging from 6 to 10 points. Font sizes of 9-points and 10-points yield reliably faster performance than the smaller sizes. 10-point Arial and MS Sans Serif were the preferred.
- Bernard and Mills (2000) evaluated 10-point and 12-point Arial (sans serif) and Times New Roman (serif). The 12-point fonts were reliably preferred.
- Bernard et al. (2001a), in their comparison of eight fonts, also evaluated point sizes of 10, 12 and 14. The 12-point fonts were reliably preferred over the 10-point fonts.
- Bernard et al. (2001b) evaluated 12-point and 14-point versions of Times New Roman, Georgia, Arial, and Verdana with older users. (Average age of 70 with a range of 62 to 83 years.) The 14-point fonts yielded better reading efficiency and the 14-point sans serif fonts were preferred.

What can we conclude from these font point size studies?

- The minimum size for Web page fonts should be 10-points.
- A 12-point font yields faster reading speeds and is the preferred size for a wide range of users.
- 14-point fonts work best for older screen viewers.

Variations in type sizes should also be minimized, generally no more than three being the maximum to be displayed at one time on a screen. In selecting a typeface, it is also important to consider its *x-height*, or the height of its lowercase x and other similar

https://hemanthrajhemu.github.io

letters. Two typefaces may share the same point size but one may *appear* significantly larger because its x-height is larger, as illustrated in Figure 3.18. Legibility can be affected by x-height. Unfortunately, user-defined browser settings may also enlarge or shrink a designed font size.

Many systems use a 12-point type for menus and 10-point for windows. Dropping below 10 points appears to degrade legibility. In Web page design, larger type sizes are recommended, 12 to 14 points for body text and 18 to 36 points for titles and headings. Line spacing should be one to one and one-half times font size. For comparison purposes, recommended paper document type sizes and styles are as follows:

Chapter headings:	24-point bold
Section headings:	18-point bold
Subsection headings:	14-point bold
Paragraph headings:	12-point bold
Body text:	10-point
Annotations/footnotes:	8-point

Never change a selected type size for a screen component to squeeze something in. The differences in size will be noticeable and visually disturbing.

Font Styles and Weight

- Use no more than
  — Two styles of the same family.
    - Standard and italic.
    - Italic is best presented in a serif font.
  — Two weights.
    - Regular and bold.
    - Bold is best presented in a sans serif font.
- Use italics when you want to call attention.
- Use bold when you want to call attention or create a hierarchy.
- In Web pages, use an underline only to indicate a navigation link.

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

abcdefghijklmnopqrstuvwxyz

Figure 3.18: Types with same point size and different x heights (from top to bottom, Gatsby, Times New Roman, and Avant Garde).

https://hemanthrajhemu.github.io

Styles. Italics may be used to emphasize something or attract attention on a screen. Because it may be hard to read on many monitors, it should be used sparingly. Boyarski et al. (1998), in their study did compare Verdana Standard with Verdana Italic, but found no differences in reading speed. (Technically, a sans serif type is slanted, not truly "italics".) For the Verdana font, larger use seems justified. Italic is best presented in a serif font. In general, though, restrict italics to words or short phrases; don't use it for large blocks of text. Never use more than two styles at one time. A standard font and its italics is a good combination. Because of degraded legibility, outline and shadow style variations should be used only with extreme caution on screens.

Boldface. Use bold when you want to call attention to something. Typically, screens will be used again and again, and something bold often becomes too visually heavy. Use of too much bold is considered shouting, and nothing stands out. Like italics, use bold sparingly, restricting it to words or short phrases. Don't use it for large blocks of text. Bold's most effective use is for titles, headings, and key items. Restrict a font to two weights, regular and bold, although in Web page design, use of more weights is acceptable. Bold is best presented in a sans serif font. The differing stroke widths of serif fonts cause presentation problems over different font sizes.

Underline. In Web pages, an underline is used to designate a navigation link. It should only be used for this purpose. Its use in any other way will be confusing to the user.

Font Case

- Use mixed-case for
  — Control captions.
  — Data.
  — Control choice descriptions.
  — Text.
  — Informational messages.
  — Instructional information.
  — Menu descriptions.
  — Button descriptions.
- Consider using upper case or capitalization for
  — Title.
  — Section headings.
  — Subsection headings.
  — Caution and warning messages.
  — Words or phrases small in point size.
- Use all lowercase with caution.

The designer often has the choice of whether to display screen components in mixed case or upper case. Upper case means all capital letters. Mixed case usually implies a predominance of lowercase letters with occasional capitalization as needed (initial letter of first word, acronyms, abbreviations, proper nouns, and so on).

https://hemanthrajhemu.github.io

The research on textual material is clear. One of the earliest studies, Tinker (1955), in a study of reading from hard-copy materials, found that mixed-case text is read significantly faster than uppercase text. Subsequent studies also found large advantages in reading speed and reading comprehension for mixed-case text. The advantage of mixed-case over uppercase text is that it gives a word a more distinctive shape. Uppercase letters are all the same height while lowercase letters have different heights. These height differences aid comprehension.

The research on screen captions and menu choice descriptions, however, leans in another direction. Studies have found that captions and menu choice descriptions using uppercase characters are searched faster than those using mixed-case characters.

Why this difference? The materials giving better results for mixed-case text appear to be of a longer, textual nature. The caption materials appear to be single words or short phrases. It may be that the superiority of mixed-case text does not exhibit itself until text of an extended nature is read. Why short uppercase captions were actually superior to mixed-case ones is unknown. In light of this research, the following is recommended.

Mixed case. Always use mixed case for anything textual in nature, including text itself, messages, instructional information, figure and table descriptions, and so forth. Text, messages, and instructions reflect the years of research on readability. Also use mixed case for most other screen components, including control captions, data, control choice descriptions, and menu descriptions. These mixed-case recommendations also reflect what is becoming a de facto standard, found in various product style guides. One choice in using mixed case is whether to use the *sentence style* or *headline style* of presentation. Sentence style is what you are reading now: the initial sentence letter is capitalized and the remainder of the sentence is lowercase (except for acronyms, abbreviations, proper nouns, and so on). Headline style involves capitalization of all significant words in a sentence. For anything more than one sentence in length, the sentence style must be followed in presenting textual information. For short phrases, such as control captions and headings, the more declarative headline style may be used.

Uppercase. Unfortunately, many style guides recommend presenting everything on a screen in mixed case. That this is an extrapolation of the textual reading research to all written words can only be assumed. Contrary to style guide recommendations, on screens capitalization can and should be judiciously used. Consider using uppercase text for the screen title and, most importantly, for all screen headings. Capitalization will set headings off from the many other screen components described above, which are displayed in mixed case. Headings on screens are a learning aid. They enable the user to become familiar with screen organization and relationships. With experience, the screen user finds headings less important. Capitalization will set them off from the remaining screen elements, making them easier to ignore when they are no longer needed. Screen design research does not discount using uppercase. On Web screens, the endless variety of display techniques (different fonts, sizes, weights, and so on) makes the use of capitalization for component differentiation much less necessary. On all screens, however, do consider using capitalization to call attention to important things, including caution and warning messages. Also, capitalization may be

https://hemanthrajhemu.github.io

substituted for mixed case when a small font size is necessary within a screen component, and the small size degrades word legibility.

All lowercase. In an attempt to be different, text or sentences in all lowercase have begun to appear; that is, there is no initial capital letter on the first sentence word. Be cautious in using this approach; it does not conform to the mental model concerning sentence structure we have well-ingrained within us. The visual beginning of a sentence anchor point has disappeared, it looks out of context, and it looks very casual. This style is rarely appropriate for a business application.

Defaults

- For graphical operating systems, use the standard system fonts.
- For Web pages and applications design for the default browser fonts.
- Consider that the user may change the fonts.

Browsers display different default fonts depending on browser type and version, and the operating system the browser runs on. A system's default fonts should be used in design to ensure usability and consistency. PC default fonts are Times New Roman and Arial; for Macintosh, Times, and Helvetica. While some users modify browser preferences to display fonts of their own choosing, these choices are not possible to anticipate. Design must be legible and readable using the default fonts, the font aspect that can be controlled in design.

Consistency

- Establish a consistent hierarchy and convention for using typefaces, styles, and sizes.
  - — Decide on a font for each different level of importance in the hierarchy.
  - — Communicate hierarchy with changes in
    - Size.
    - Weight.
    - Color.

To aid screen learning and improve screen scanning and readability, apply typeface size and style conventions in a consistent manner to all screen components. First, determine the hierarchy or levels of importance to be presented within an application or Web site, then, decide on a font for each level of the hierarchy. In a graphical business system, screen elements include the title, section headings, subsection headings, control captions, data, instructional information, and so forth. In Web site design important elements include the page heading, section headings, body text, icon captions, lists of navigation links, and so forth. Finally, communicate the hierarchy through variations is font size, weight, and color. Ensure that there is adequate visual contrast between elements composing the hierarchy. (Do not forget, however, that screen element location and its positional relationship to other screen elements also aid in an element's identification.)

https://hemanthrajhemu.github.io

Some cautionary reminders: Choose and combine font families with care, and too many font variations will only create visual clutter.

Text Backgrounds

- For rapid reading and understanding present black text on plain, high-contrast backgrounds.

Adequate contrast between text and its background must always be provided. Black text on a plain background has been found to yield faster reading than black text on a medium textured background. Avoid using a patterned background. Studies also have found that people read black text on a white background up to 32 percent faster than reading light text on a dark background (Koyani et al. 2004). Using color for text and backgrounds is discussed in Step 12.

Other

- Always consider the visual capabilities of the user.
- Always verify that the design has succeeded using the selected fonts.

Unfortunately, much-too-small text and poor contrast between text and background remain a serious problem in Web page design, as evidenced by it being the number one reported problem in the Nielsen (2005e) list of Top Design Mistakes. Why is text so difficult to read? Nielsen (2002) proposes that

Most Web designers are young. They have perfect vision and tiny text does not bother them like it does those middle-aged or older.

Web designers often use expensive high-quality monitors. These monitors are easier on the eyes.

Web designers often don't actually read what they create. They simply glance at it to make sure it looks great. If you don't have to read the words, it doesn't matter that the characters are small.

No matter what the reason for poor design, always do the following:

Consider the user's age and possible visual disabilities. Smaller type sizes and text with insufficient background contrast may be more difficult for older people to use. People with visual disabilities must also be considered. (See Step 10.) Type sizes, styles, and backgrounds should always be chosen considering the ages and visual capabilities of the screen users.

Test chosen fonts. Always check in testing that the design has succeeded using the fonts selected in design. Ensure that the screens are legible and readable by all projected users at all viewing distances and angles using all expected monitors. (See Step 14.)

https://hemanthrajhemu.github.io

Application and Page Size

A decision that must be made early in the Web page design process is whether to create fewer long pages that may require extensive scrolling to view all of their contents, or to create a larger number of shorter pages necessitating more frequent movements between pages. The design goal is to allow people to move through an application or Web page as quickly and efficiently as possible. Whether to use paging or scrolling depends on who the primary users of the system are, and what type of tasks is being performed.

Scrolling and Paging

- Scrolling:
  - — Avoid scrolling to determine a page's subject and what it contains.
  - — Minimize vertical page scrolling.
  - — When vertical scrolling is necessary to view an entire page
    - Provide contextual cues within the page that it must be scrolled to view its entire contents.
    - Provide short pages if people are looking for specific pieces of information.
    - Facilitate fast scrolling by highlighting major page items.
    - Provide a unique and consistent "end of page" structure.
  - — Avoid horizontal page scrolling.
  - — Use longer scrolling pages when people are reading for comprehension.
  - — Use paging rather than scrolling if system response times are reasonably fast.
- Paging:
  - — Encourage viewing a page through "paging."
  - — Create a second version of a Web site, one consisting of individual screens that are viewed through "paging."

In screen design before the explosion of the Web, the favored method of asking the user to move between screens of information was through paging. A full screen of information is presented, the user does what is necessary to do to the screen, and then the entire screen is transmitted through a key action. If other user actions are then necessary to accomplish an objective, another full screen is presented and the process continues until an ending is reached. This method of interaction was practical and efficient for computer and monitor technology existing at that time, and it presented the user consistent and meaningful "chunks" of information to work on. Screen scrolling as an interaction method was also used over this time period, but on a much more limited basis.

User performance using paging and scrolling has been the subject of occasional research efforts. One early study found that for novice users, paging through screens yielded better performance and was preferred. Dillon (1992), however, in a review of the research literature found no reliable differences using either method.

The scrolling technique used today in viewing screens was also established through research. There are two possible ways to control and view a scrolling screen: the telescope and the microscope approaches. In the *telescope* method, the model is that of a

https://hemanthrajhemu.github.io

telescope: the window moves around the screen data much as a telescope scans the stars in the night sky. In the *microscope* approach, the screen data appears to move under a fixed viewing window, the way an object placed under a microscope is manually moved around to see it in its entirety. The research found that the telescope approach is more natural and causes fewer errors, and it was implemented. This is why, when scrolling today, clicking the up arrow on the scroll bar causes the data displayed on a screen to move downward. The data is not actually moving; the telescope through which the data is being viewed is moving upward.

Web technological requirements have tilted the scale toward scrolling as the favored method of page viewing. Early in its popular life, scrolling seemed to cause some difficulties for Web users, Nielsen (1996) reporting that only 10 percent of users went beyond what was visible on the screen by using scrolling. Apparently, in 1996 paging was a well-ingrained habit. As user familiarity with the Web increased, so did scrolling familiarity, and in 1999 Nielsen (1999b) reported that most users were now used to scrolling, having gained the understanding that things were often hidden from view. In spite of its seeming acceptance, excessive page scrolling can be cumbersome and slow. It also can disrupt the user's perception of spatial location within a page, especially while the text is scrolling. To minimize these problems, in Web page design do the following.

Avoid scrolling to determine page contents. A page's subject should be immediately recognizable. Elements crucial to identifying a page's contents must be viewable without requiring page scrolling. If not visible when the page is first displayed, these elements may never be seen. Place these content-identifying elements "above the fold," or in the top 4 inches of the page.

Minimize vertical scrolling. Some scrolling may be necessary to view the entire contents of a page. Minimize the requirement for vertical scrolling when defining, organizing, and laying out a page's components. For example, avoid large graphics and excessive amounts of white space. Place closest to the page's top the information most likely to be needed.

Provide contextual scrolling cues. Lower parts of a page may be overlooked, especially if the visible portion appears to satisfy the user's needs and the user erroneously concludes that no more can be done. For pages exceeding one screen in length, provide contextual cues to the user that part of the page is hidden and that viewing the entire page will necessitate scrolling. Organize the information, for example, so that it is obvious there is more to follow. An incomplete alphabetical arrangement of information would be one such clue. Also, be cautious in placing screen-wide horizontal lines between groupings of content. This could convey to a casual user that the page's bottom has been reached. Other possible "scroll stoppers" (Koyani et al., 2004) include horizontal lines, inappropriate placement of screen controls, and cessation of background color.

Provide short pages. Short pages should be provided when people are looking for specific items of information because scrolling can consume much time. Koyani et al. (2004) report that studies have found that older users scroll much more slowly than younger users, and that Internet users spend about thirteen percent of their time scrolling.

https://hemanthrajhemu.github.io

Facilitate fast scrolling by highlighting major page items. Dragging a scroll bar (as opposed to clicking the up and down arrows) can cause a page to move too fast for easy reading of prose text. Major headings can be scanned, however, if they are highlighted and well positioned (Bailey et al., 2000).

Provide a unique end-of-page structure. To identify the page bottom, provide a visually unique and consistent ending on all pages. A row of navigation links and other elements such as copyrights, e-mail address, and other contact information can signify "The End." Do not place these ending elements in other locations within the page or other pages. They will convey falsely to the user that the end has been reached.

Avoid horizontal scrolling. While vertical scrolling is now acceptable in Web page design, horizontal scrolling must be avoided. A page too wide to be completely displayed within the confines of a screen will require continuous scrolling as reading is performed. This is extremely cumbersome and inefficient. Confining scrolling to simple up-and-down movements greatly simplifies the scrolling process.

Use longer scrolling pages when people are reading for comprehension. Koyani et al. (2004) recommend that longer pages requiring scrolling should be used when paging through links introduces a time delay that can interrupt users' thought processes. The context of a message will be maintained better with minimal page delays. They report that if pages have fast loading times, there is no reliable study differences between paging and scrolling in reading comprehension. When reasonable response times exist, then, use paging rather than scrolling.

Encourage use of paging. Full-screen paging on the Web can be done by using the page-up and page-down keys or clicking on the scroll bar page-up or page-down icons. Text is then moved by the number of lines equaling screen size. This is almost always faster than scrolling a line at a time. Some recent studies have addressed the issue of Web page scrolling and paging. Piolat et al. (1998) found, as in past studies, no significant differences between paging and scrolling in text reading. Paging users, however, were better at building mental representations of the text, finding relevant information, and remembering the main ideas. Blinn and Biers (1999) found that, when searching Web sites with shorter pages, information was found faster using paging as opposed to scrolling. Mead et al. (1997) addressed the effects of an inexperienced user's age on paging and scrolling. Older users (64 to 81) were more likely to use a page at a time while reading; younger users (19 to 36) tended to scroll a line at a time. Older users also performed best when using short full pages of information rather than continuous long pages. Paging navigation, it seems, does have advantages for users. Encourage its use.

Paging version. Ensure the availability of full pages for reading and searching on the Web and, by creating a second version of a Web site, one consisting of individual screens that are viewed through "paging."

https://hemanthrajhemu.github.io

Amount of Information to Present

- Present the proper amount of information for the task.
  — Too little is inefficient.
  — Too much is confusing.
- Present all information necessary for performing an action or making a decision on one screen, whenever possible.
  — People should not have to remember things from one screen to the next.
- Restrict screen or window density levels to no more than about 30 percent.

Proper amount of information. Presenting too much information on a screen is confusing; there will be greater competition among a screen's components for a person's attention. Visual search times will be longer, and meaningful structure will be more difficult to perceive. Presenting too little information is inefficient and may tax a person's memory as information contained on multiple screens may have to be remembered.

Present all necessary information. In general, present all information necessary for performing an action or making a decision on one screen. If information located on different screens must be remembered, a person's memory will again be taxed. Developing a screen with all the necessary information requires careful analysis of the user's tasks. Conversely, do not clutter up a screen or Web page with unneeded information.

Screen density. One objective measure of "how much" should go on a screen has been developed: "density." Density, by definition, is a calculation of the proportion of display character positions on the screen, or an area of the screen containing something. Density is clearly related to complexity, since both measure "how much is there." Complexity looks at elements, density at characters, so they should rise and fall together.

In general, studies show that increasing the density of a display increases the time and errors in finding information. There are two types of density to be calculated on a screen: overall and local.

Overall density is a measure of the percentage of character positions on the entire screen containing data. Danchak (1976) stated that density (loading, as he called it) should not exceed 25 percent. Reporting the results of a qualitative judgment of "good" screens, he found their density was on the order of 15 percent. Tullis, in his 1981 study, reported that the density of screens from an up-and-running successful system ranged from 0.9 to 27.9 percent, with a mean of 14.2 percent. Using this and other research data, he concluded that the common upper density limit appears to be about 25 percent.

Thacker (1987) compared screens with densities of 14 percent, 29 percent, and 43 percent. Response time increased significantly as screen density increased. He found, however, that the time increase between 14 percent and 29 percent was much smaller than the time increase between 29 percent and 43 percent. He also found increased error rates with greater density, the 43 percent density screens showing significantly more errors.

https://hemanthrajhemu.github.io

Local density is a measure of how "tightly packed" the screen is. A measure of local density, derived by Tullis, is the percentage of characters in the 88-character visual acuity circle described in Step 1, modified by the weighting factors illustrated below.

```
    012222210
  0123445443210
 023456777654320
01235679+97653210
 023456777654320
  0123445443210
    012222210
```

For every character on the screen, a local density is calculated using the above weighting factors, and then an average for all characters on the screen is established.

Figures 3.19 and 3.20 are the original and redesigned screens from the 1981 Tullis study. Density measures for these screens are

Figure 3.19 (original):

Overall density = 17.9 percent

Local density = 58.0 percent

Figure 3.20 (redesigned):

Overall density = 10.8 percent

Local density = 35.6 percent

```
TEST RESULTS    SUMMARY:    GROUND

GROUND, FAULT T-G
3 TERMINAL DC RESISTANCE
>          3500.00 K OHMS T-R
=            14.21 K OHMS T-R
>          3500.00 K OHMS R-G
3 TERMINAL DC VOLTAGE
=             0.00 VOLTS   T-G
=             0.00 VOLTS   R-G
VALID AC SIGNATURE
3 TERMINAL AC RESISTANCE
=             8.82  K OHMS T-R
=            14.17  K  OHMS T-R
=           628.52 K  OHMS R-G
LONGITUDINAL BALANCE POOR
=            39    DBB
COULD NOT COUNT RINGERS DUE TO
   LOW RESISTANCE
VALID LINE CKT CONFIGURATION
CAN DRAW AND BREAK DIAL TONE
```

Figure 3.19: Original screen, from Tullis (1981).

https://hemanthrajhemu.github.io

In both cases, the more effective redesigned screen had lower density measures. In his 1983 follow-up study, Tullis found a lower local density to be the most important characteristic, creating a positive "visually pleasing" feeling.

The research does suggest some density guidelines for screens. Maintain overall density levels no higher than about 30 percent. This upper overall density recommendation should be interpreted with extreme care. Density, by itself, does not affect whether or not what is displayed "makes sense." This is a completely different question. Density can always be reduced through substituting abbreviations for whole words. The cost of low density in this case may be illegibility and poorer comprehension. Indeed, poorly designed screens have been redesigned to achieve greater clarity and have actually ended up with higher density measures than the original versions. How it all "hangs together" can never be divorced from how much is there.

In conclusion, all this density research was performed using text-based screens. With many boxed or specialized controls found on graphical screens, such as list boxes or sliders, it is much more difficult to calculate density as has just been illustrated. Is it necessary to do so on graphical screens? Not really. The research was described to show the value of reducing density in screen design. From a practical standpoint, if the guidelines for alignment and groupings are adhered to, screen density will usually be reduced to an acceptable level.

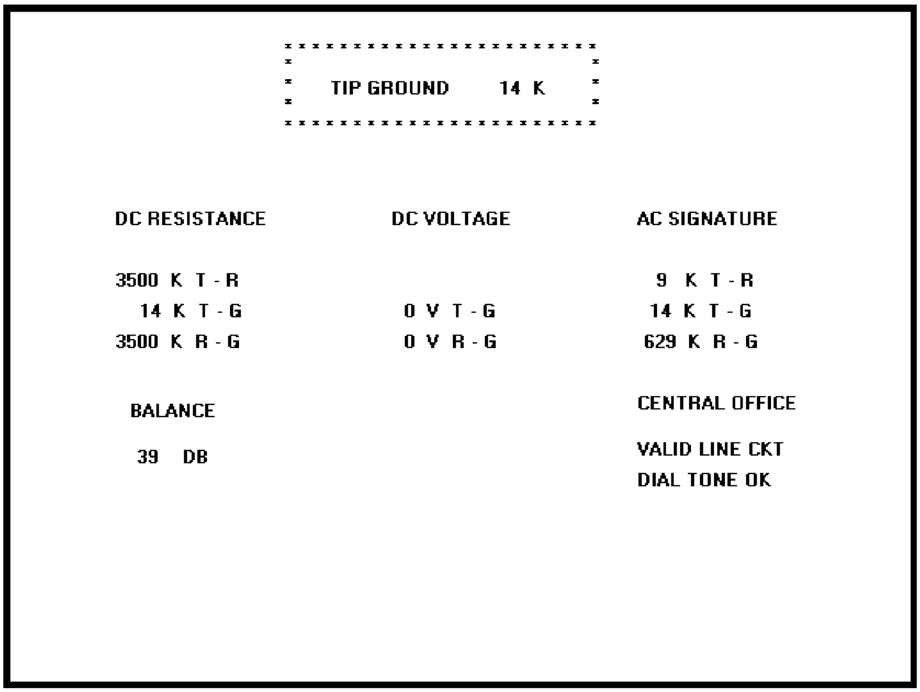


Figure 3.20: Redesigned screen, from Tullis (1981).

https://hemanthrajhemu.github.io

Paper versus Screen Reading

- Provide a simple facility for printing out a hard copy of documents.

Printing technology has been evolving for several centuries. Factors such as type size and style, character and line spacing, and column and margin widths have been the focus of research for a good part of that time. The product of this research is highly readable and attractive printed material. Conversely, CRT-based characters are a relatively new innovation, with many technical limitations. The result is a displayed character that often lacks the high quality a paper medium can provide. This disparity in quality has resulted in performance differences when paper and screen reading of materials have been compared. Various researchers over the years have found screen reading speeds as much as 40 percent slower, and more error prone.

More recent research indicates that, as display resolution improves, the reading speed differences are being reduced, if not eliminated. Reading performance, using both hard-copy paper documents and monitors at different resolutions, was the subject of two studies by Ziefle (1998). In the first study, she compared paper printed at 255 dots per inch (dpi) and monitors whose resolutions were 832×600 pixels (60 dpi) and 1664×1200 pixels (120 dpi). A 19-inch monitor showing black characters on a light background was used, and reading speeds and proofreading accuracy were compared. Ziefle found no difference in performance between the monitors with different resolutions. There was, however, a significant advantage for hard copy over monitor usage. Reading from paper was faster, 200 words per minute to 180 wpm, and proofreading accuracy was higher. Reading from paper was also preferred by 80 percent of the users. The remaining 20 percent preferred the higher resolution screen (1664×1200).

Gujar et al. (1998), however, compared text written on paper with text presented on screens and found no statistically significant differences in reading time and detected errors. Participants, though, rated reading from paper significantly better than reading from screens.

For extended reading, the hard-copy display of material still appears to have significant advantages, however, and will probably continue to for some time. People prefer reading from paper. Necessary screen-reading motor activities (scrolling, paging, and so on) are more cumbersome than the page-turning task of reading from a paper. Paper reading also offers much more convenience, the reader being much less constrained physically and environmentally. (Paper reading can be performed by the pool, in bed, or any place else where significant light exists.) Therefore, always provide a simple facility for printing hard copies of documents.

Application Screen Elements

An application lets people enter, select, delete, modify, and sometimes just view data contained in a database. The elements contained on a screen must support these multiple purposes. Historically in screen design, the entities on screens used for these

https://hemanthrajhemu.github.io

purposes were called fields. They were usually rectangular in shape, varying in size depending upon the amount of information that must be entered, edited, or viewed. With the introduction of graphical systems, however, a variety of other objects were created for people to place information in a database. As a group, these entities are called controls. Many graphical controls present listings from which the user simply selects the appropriate choice, or choices. Some controls, of course, still allow data to be entered within them. The control analogous to the *field* in graphical systems is called a *text box*.

Controls have two attributes in common. (1) They contain a data field, an area in which data or information may be keyed, selected or simply displayed, and (2) they often have a descriptive caption or label identifying the type of data contained within the field.

Controls cannot exist alone on a screen. Other kinds of information are necessary in order to provide context for the user, and to aid in providing and interpreting data. These additional elements include a title, headings, and instructional information. The following guidelines address these screen components. (Proper usage of the different controls and control design guidelines are addressed in Step 7.)

Title

A meaningful title will provide a road map aiding understanding of an application's organization and navigational structure.

Windows

- All windows must have a title located at the top.
  - — Exception: Windows containing messages.
    - Clearly and concisely describe the purpose of the window.
    - Spell it out fully using an uppercase or mixed-case font.
    - If title truncation is necessary, truncate it from right to left.

The window title should be positioned at the top center and fully spelled out using either uppercase letters or mixed-case in the headline style. Using an uppercase font will give it the needed moderate emphasis, aiding in setting it off from the screen body (IBM's SAA CUA and Microsoft's Windows guidelines display the title, like all screen components, in mixed-case letters). A larger font is also desirable. Windows containing messages, however, need not have a title. The title should clearly and concisely describe the screen's purpose. If the window appears as a result of a previous selection, the title should clearly reflect the wording of the selection made to retrieve it. For small windows, where title truncation is necessary, truncate from right to left.

If the title appears above the menu bar, the title's background should contrast with that of the bar. A recommendation is to use the same background color and caption color as the screen body. A title can always be identified by its topmost location on the screen, so using a color different from other screen components may add to visual confusion.

https://hemanthrajhemu.github.io

Web Pages

- All Web pages must have titles located in the browser title bar and on the content pages themselves.
  — Browser bar title and page title should be consistent.
- Titles must be
  — Descriptive.
  — Unique and meaningfully different from other Web pages.
  — Concise.

All Web pages must have a title located in the browser title bar and on each content page as well. Content page titles should be positioned at the top of the page. The browser title bar and page heading should be consistent to avoid confusing users. Titles are used by search engines to identify pages, so titles should be descriptive, unique, and meaningfully different. If titles are not unique and meaningfully different, users cannot differentiate pages. This will also create problems in searching the Favorites listing or scanning the History List for a particular page.

Captions/Labels

How important are good captions or labels in design? Very, according to Resnick and Sanchez (2004). They conducted an experiment to explore the value of good labels in efficiency and perceived ease-of-use of a Health Product shopping Web site. (Headings were also considered as labels in this research. This text describes labels and headings as separate entities.) Resnick and Sanchez also identified two existing structural organization schemes for health products on Web sites — by product (bars, pills, book, etc.) and by task (weight loss, stress reduction, etc.). These schemes were also evaluated.

Study participants, using a card-sorting task, derived three levels of labeling — good, medium, and poor, for both organization schemes. The study found that good labels had a profound effect on performance. Good labels resulted in

- 90 percent more efficiency in finding an item.
- 25 percent fewer clicks to complete a task.
- Significantly less errors (strays from the optimal path).
- Significantly more items found.
- Significantly higher user-satisfaction ratings.

In structural evaluation, good labels resulted in no benefit for one scheme over the other. With bad labels, however, users were more efficient using the product-based structure. The author's study conclusion — the key to good site architecture, and by extension good navigation, is good labels.

- Identify controls with captions or labels.
- Fully spell captions out in a language meaningful to the user.

https://hemanthrajhemu.github.io

- Use a mixed-case font.
- Capitalize the first letter of each significant word.
- End each caption with a colon (:).
- Choose distinct captions that can be easily distinguished from other captions.
  — Minimal differences (one letter or word) cause confusion.
- Provide consistency.

Identify controls with captions. All screen controls should have captions or labels that identify the content of the control. Create labels using words that users use to refer to items and avoid jargon. Captions should be included for most controls. The context in which information is found in the world at large provides cues to the information's meaning. A number on a telephone dial is readily identifiable as a telephone number; the number on a metal plate affixed to the back of an automobile is readily identified as a license number. The same information displayed on a screen, having lost context, may not be readily identifiable.

There are, however, some exceptions to this rule on read-only or inquiry screens. The structure of the data itself in some cases may be enough to identify its meaning. The most obvious example is name, street, city, state, and zip code. Date may be another possibility. Elimination of these common captions will serve further to clean up read-only screens. Before eliminating captions, however, it should be determined that all screen users will be able to identify these fields all the time.

Structure and size. Captions on screens must clearly, concisely, and unambiguously describe the information displayed. Captions are very important for inexperienced screen users. As one becomes more experienced, their importance diminishes. Therefore, captions should be fully spelled out in the natural language of the user. In general, abbreviations and contractions should not be used. To achieve the alignment recommendations to be discussed shortly, an occasional abbreviation or contraction may be necessary, but choose those that are common in the everyday language of the application or those that are meaningful and easily learned. Also, display captions in a moderate brightness or intensity. Visual emphasis will be directed to the screen data or information.

Significant word capitalization. With mixed-case field captions, capitalize the first letter of each significant word using the headline style previously described. A caption is not a sentence but the name for an area into which information will be keyed. This makes it a proper noun. In situations in which a caption is phrased as a question, it is a sentence, and then only its initial letter should be capitalized. Never begin a caption or sentence with a lowercase letter. A capital letter makes it easier for the eye to identify the start of each caption. Unfortunately some style guides do not follow the headline style of using a capital letter for the initial letter of each significant word of the caption. They prefer and recommend the sentence style, capitalization of the initial letter of the first word only (except for acronyms, abbreviations, proper nouns, and so on).

https://hemanthrajhemu.github.io

Designate with a colon. A caption should be ended with a colon (:) to clearly identify it as a caption, and also to clearly distinguishing it from a data field. The colon is unobtrusive, does not physically resemble a letter or number, and is grammatically meaningful, since it is used chiefly to direct attention to matter that follows. Unfortunately, many graphical systems do not follow this convention, and captions visually blend with other screen elements.

Because the recommended entry area for an entry control will be a box, adequately distinguishing the caption from the entry field itself, the inclusion of a colon may seem redundant. However, read-only, display, and inquiry screens are most effective if the data displayed is not presented in a box, making a colon to distinguish caption from data absolutely necessary. Including a colon after all captions, therefore, will provide consistency across all screens.

Distinctiveness. Captions that are similar often repeat the same word or words over and over again. This directs a viewer's attention to the pattern created by the repetitive word, increases the potential for confusion, adds to density, and adds to screen clutter. A better solution is to incorporate the common words into headings, subheadings, or group identifiers, as illustrated in Figure 3.21.

Consistency. Provide the same caption wording for all identical data fields on all screens.

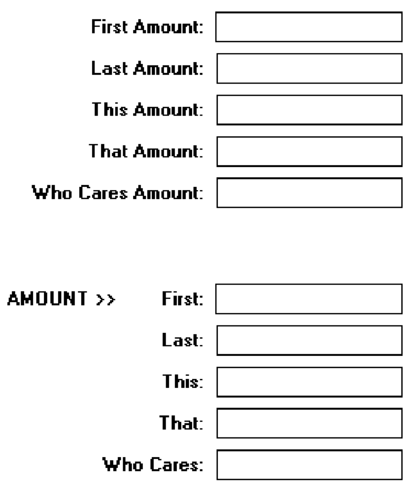


Figure 3.21: Providing better control caption discrimination. (The redundant word "amount" is incorporated into a heading.)

https://hemanthrajhemu.github.io

Data Fields

- For entry or modifiable data fields:
  - — Display data within
    - A line box.
    - A box with a contrasting light-colored background.
  - — Break long structured data items into logical pieces.
  - — Provide a field length commensurate with the size of the entry.
- For inquiry or display/read-only screens containing non-changeable data
  - — Display the data on the normal screen background with no borders.
- For temporarily inactive data fields
  - — Display the data content of the data field lighter than active fields.
  - — Do not change the background color of the entry area or lighten the caption.
- Visually emphasize the data fields.

Three kinds of data fields are found on screens. (1) Those into which data may be entered, or that already contain data that may be modified, (2) those that display data that cannot be changed, and (3) those that may be temporarily inactive and not permitting changes. The design rules for each differ.

Entry or modifiable data field. An entry or modifiable field must possess the following qualities:

- Draw a person's attention to the fact that information must be keyed or selected in it.
- Not detract from the legibility of characters being keyed into it.
- Permit easy designation of the kind or structure of the entry required, such as incorporation of slashes (/) in a date field.
- Provide an indication of the maximum size of the entry required through field length.

In an early study using text-based screens, it was found that people overwhelmingly preferred that something should be displayed on a screen to indicate where data must be entered. In another study, it was found that the best alternatives for defining an entry field were a broken line underscore or a box. An underscore was traditionally used on text-based screens; the box is now recommended for, and should be used on, graphical screens and Web pages.

Break long structured data items into logical pieces. A telephone number broken into three pieces, for example, will be easier to key and review for verification.

Create entry fields that are large enough to show all the entered data without scrolling. Provide the user an indication of the maximum size of the entry required in a data field through field length. If due to space constraints a field must be shortened, provide field scrolling to capture the entire entry. If an entry field has a maximum limit, state that limit adjacent to the field. In longer text entry fields it can be frustrating to the user to bump up against the limit, thus requiring that the entry be edited in order to fit within the field's boundaries.

https://hemanthrajhemu.github.io

Display/read-only screens. For inquiry or display/read-only screens, it is best for the data to be presented on the background of the screen. This permits easier scanning and information location; the reasoning will be discussed in the "Display/Read-Only" screen organization section following shortly.

Temporarily inactive fields. If a displayed field is temporarily inactive, display the data content of the field lighter than the data content of active fields. Do not change the background color of the entry area or lighten the caption.

Visual emphasis. Data or information is the most important part of any screen. It should be highlighted in some manner, either through higher intensity, boldness, or a brighter color. Headings and captions are most important for the new or casual user. As people become familiar with a system and screens, their attention is immediately drawn to the data when a screen is presented. An experienced user will often work with a screen just perusing the data, ignoring captions and headings. Highlight the data so it will attract the user's eyes. Other screen elements will be easier to ignore.

Control Caption — Data Field Differentiation

- Differentiate captions from data fields by using
  - — Contrasting features, such as different intensities, separating columns, boxes, and so forth.
  - — Consistent physical relationships.

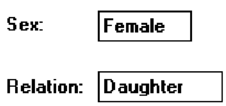


Figure 3.22

- For single data fields
  - — Place the caption to left of the data field.

Figure 3.23

- — Align the caption with the control's data.
- — Alternately, place the caption above the data field.
- — Align captions justified, upper left to the data field.

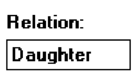


Figure 3.24

- — Maintain consistent positional relations within a screen, or within related screens, whenever possible.

https://hemanthrajhemu.github.io

- For multiple listings of columnar-oriented data, place the caption above the columnized data fields.

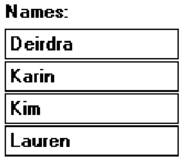


Figure 3.25

Captions must be complete, clear, easy to identify, easy to scan, and distinguishable from other captions and data fields. Captions must also be clearly related to their associated data fields.

Differentiating captions from data. Captions and data should be visually distinguishable in some manner so that they do not have to be read in context to determine which is which. A common failing of many past screens is that the captions and data have the same appearance and blend into one another when the screen is filled. This made differentiation difficult and increased caption and field data search time. Methods that can be used to accomplish differentiation, in addition to designating captions with a colon, are using contrasting display features and consistent positional relationships.

Single data fields. The recommended location for the caption is to the left of the data field and horizontally aligned with the field data. Alternately, the caption for a single data field may be positioned left-aligned above the data field. Maintain consistent positional relationships within a screen, and between multiple related screens whenever possible.

Columnar-oriented listings. For multiple listings of columnar data, place the caption above the data fields. Left/justify the caption above the data fields. Use horizontal caption formats for single fields and a columnar caption orientation for repeating fields to provide better discrimination between single and repeating fields. The single-field caption will always precede the data, and captions for repeating columnar fields will always be above the top data field.

Control Caption — Data Field Justification

- 1. First Approach
  — Left-justify both captions and data fields.
  — Leave one space between the longest caption and the data field column.

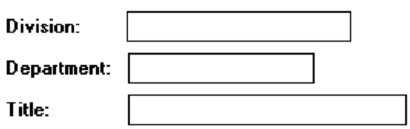


Figure 3.26

https://hemanthrajhemu.github.io

- 2. Second Approach
  — Left-justify data fields and right-justify captions to data fields.
  — Leave one space between each.

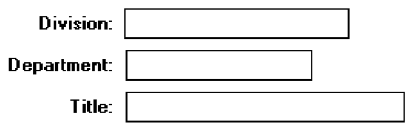


Figure 3.27

Figures 3.28 through 3.44 contain a series of screens in a variety of formats containing either entry/modification fields or display/read-only fields. The author's comments are found with each screen. What are your thoughts?

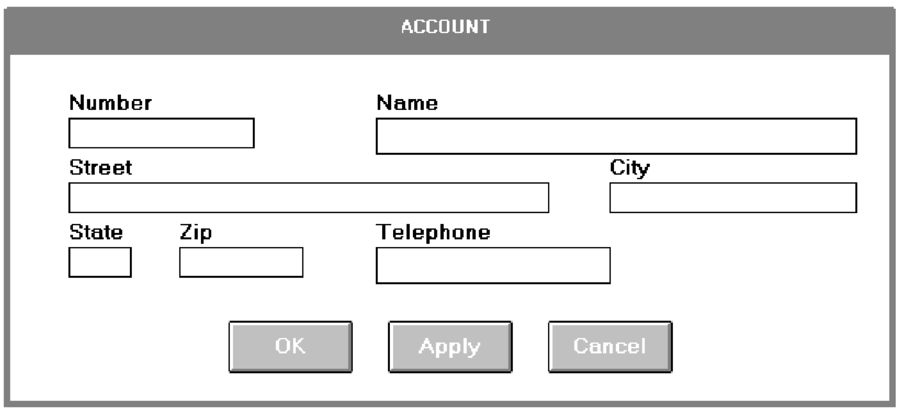


Figure 3.28: Entry screen with captions above single data fields. Captions distinct from data but with poor alignment and organization of fields. Left-to-right orientation and no groupings. Fair readability.

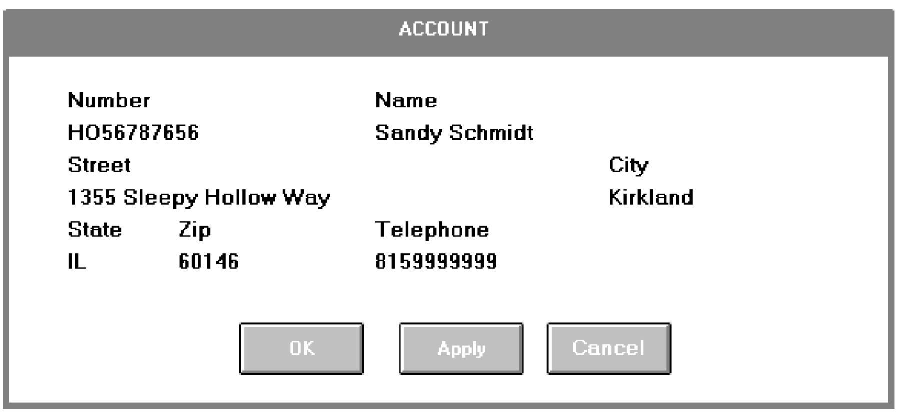


Figure 3.29: Display/read-only inquiry screen maintaining same structure as 3.22. Extremely poor differentiation of captions and data. Crowded look and extremely poor readability.

https://hemanthrajhemu.github.io

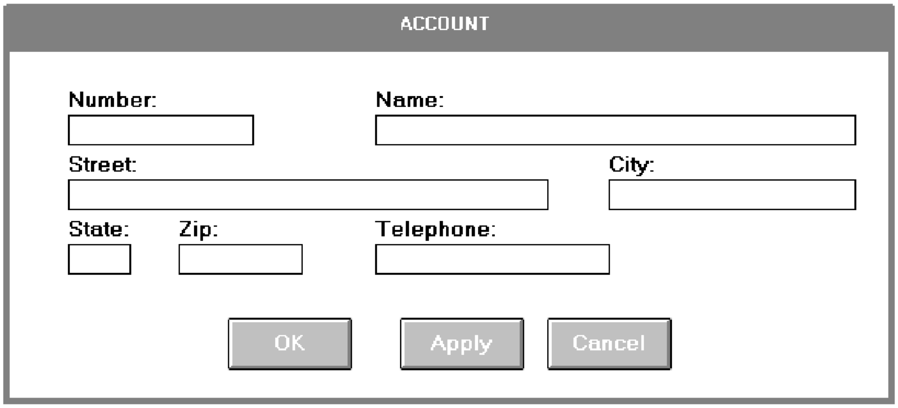


Figure 3.30: Entry screen in 3.28 with colons attached to captions. Captions somewhat more distinctive but still with poor alignment and organization of fields, left-to-right orientation and no groupings. Fair readability.

Figure 3.31: Display/read-only screen maintaining same structure as 3.30. Somewhat better differentiation of captions and data than 3.29 but still with a crowded look and poor readability.

https://hemanthrajhemu.github.io

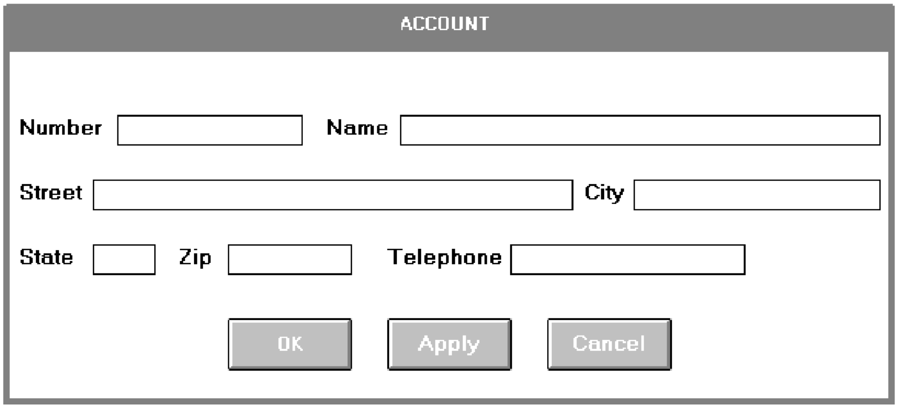


Figure 3.32: Entry/modification screen with captions to left of single-data fields. Captions distinct from data but with poor alignment and organization of fields. Left-to-right orientation and no groupings. Fair readability.

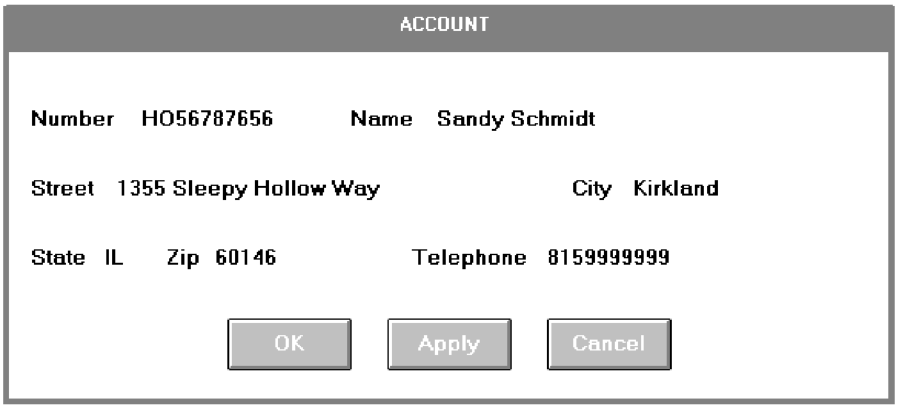


Figure 3.33: Display/read-only screen maintaining same structure as 3.32. Extremely poor differentiation of captions and data. Less crowded look than previous display/inquiry screens but still poor readability.

https://hemanthrajhemu.github.io

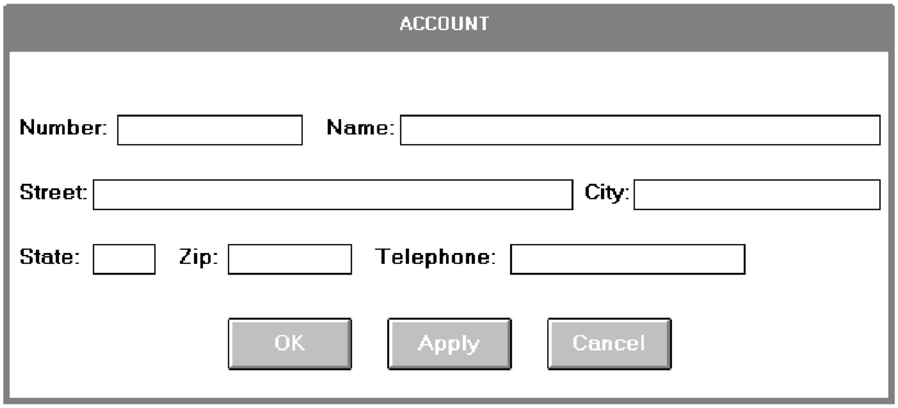


Figure 3.34: Entry/modification screen in 3.32 with colons attached to captions. Captions somewhat more distinctive but still poor alignment and organization of fields, left-to-right orientation, and no groupings. Fair readability.

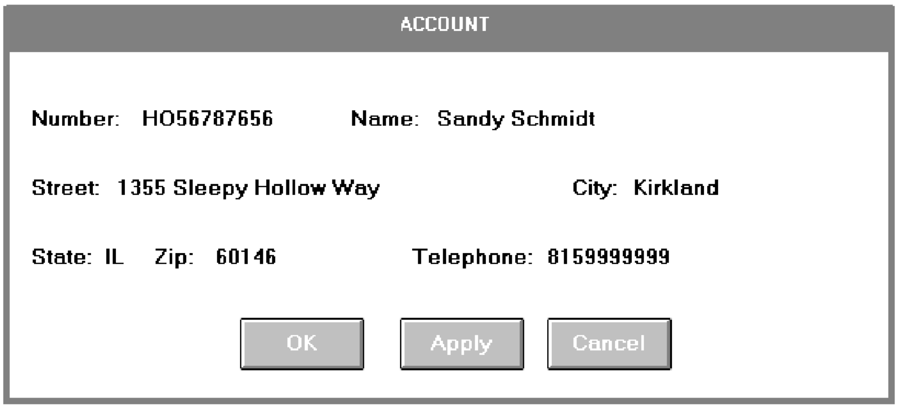


Figure 3.35: Display/read-only screen maintaining same structure as 3.34. Somewhat better differentiation of captions and data than 3.33 but still poor readability.

https://hemanthrajhemu.github.io

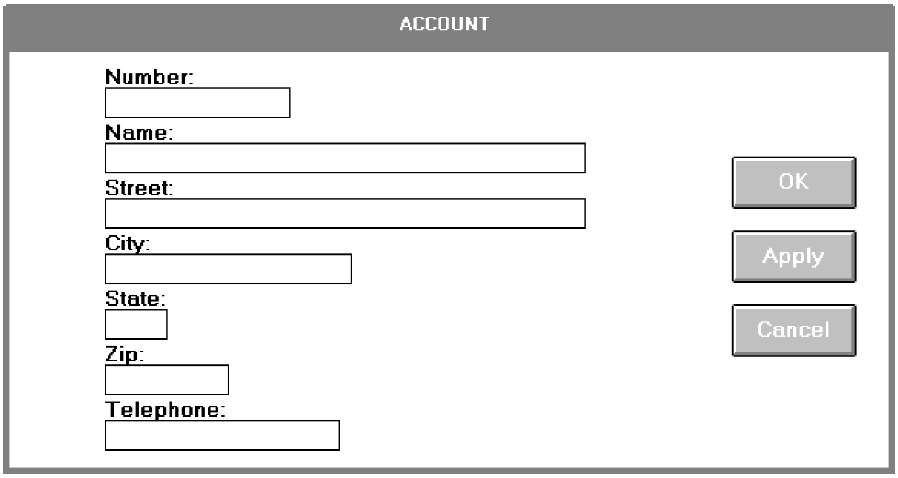


Figure 3.36: Entry/modification screen with much better alignment and readability than previous screens. Captions crowd data fields, however. Also, has no groupings and does not maintain post office suggested format for City, State, and Zip.

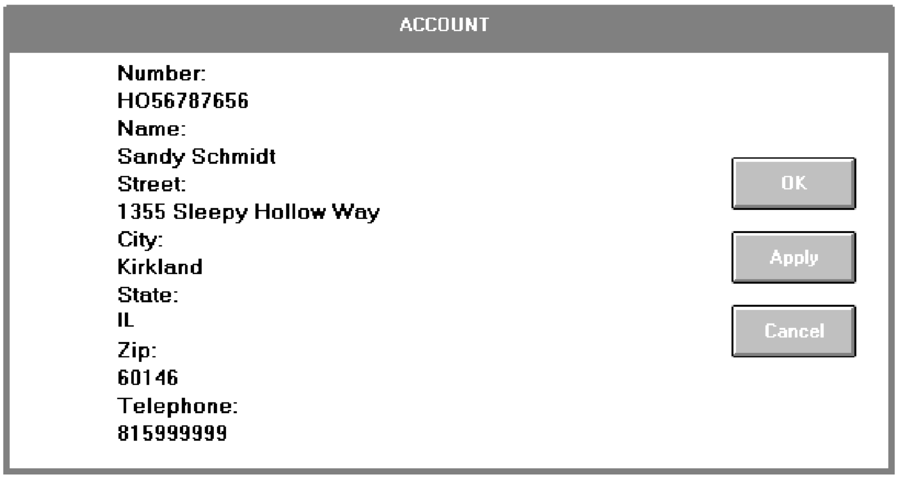


Figure 3.37: Display/read-only screen maintaining same aligned structure as 3.36. Captions not very distinctive and poor readability. Again, it looks very dense and crowded.

https://hemanthrajhemu.github.io

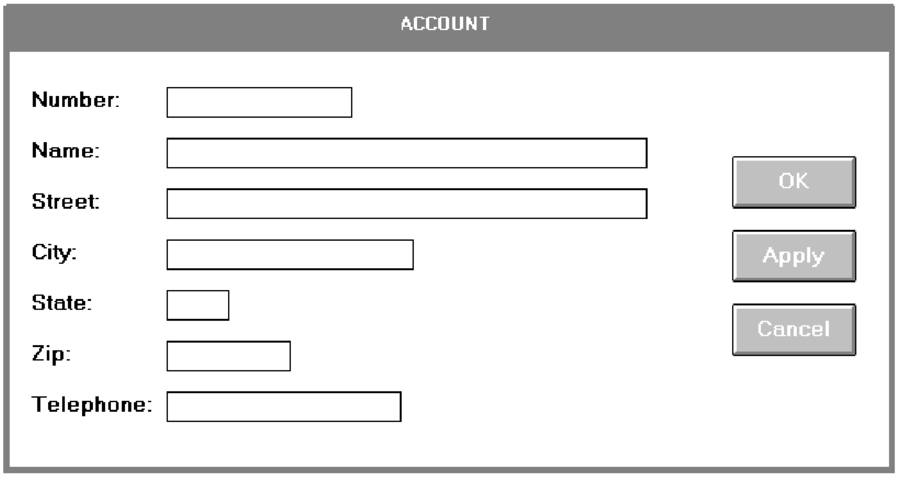


Figure 3.38: Entry/modification screen with the better alignment and readability of 3.36. Caption positioned to left, however, resulting in more distinctive data fields. Still no groupings, though, and does not maintain post office suggested format for City, State, and Zip.

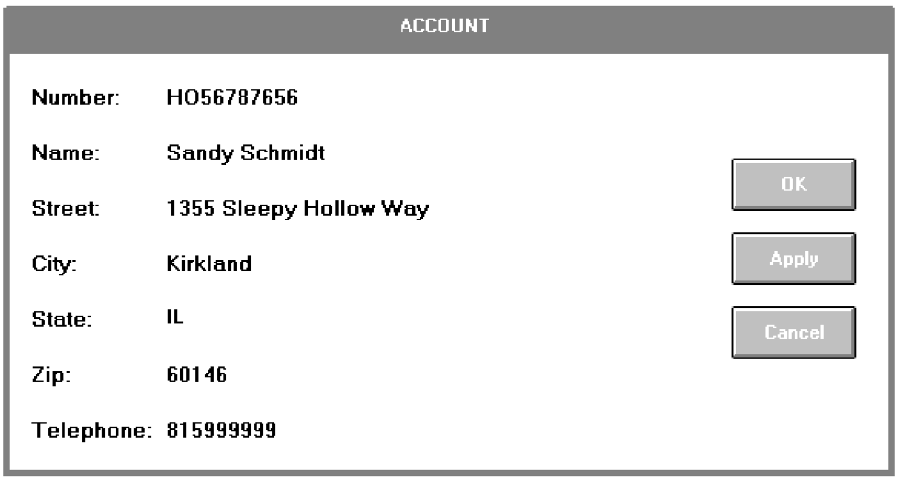


Figure 3.39: Display/read-only screen maintaining same alignment and positioning of captions of 3.38. Captions and data much more distinctive. Still no groupings though, and does not maintain post office suggested format for City, State, and Zip.

https://hemanthrajhemu.github.io

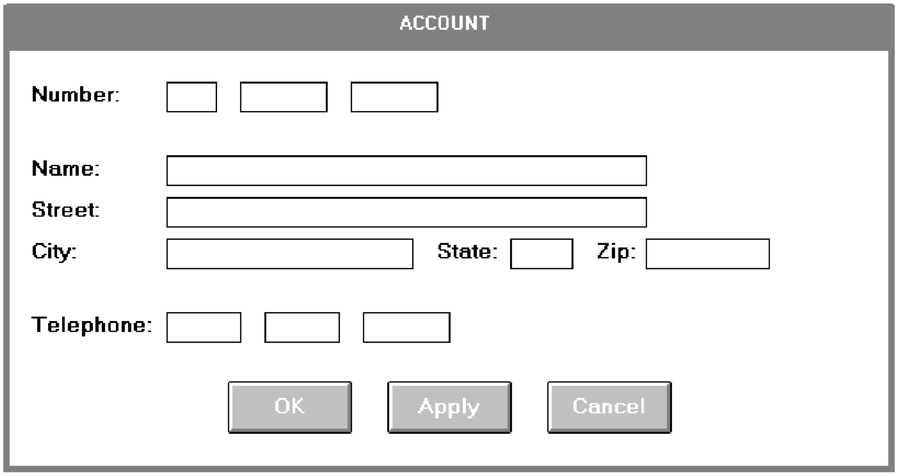


Figure 3.40: Entry/modification screen providing alignment, groupings, and the suggested and familiar post office address format. Data fields also segmented to enhance readability (Number and Telephone).

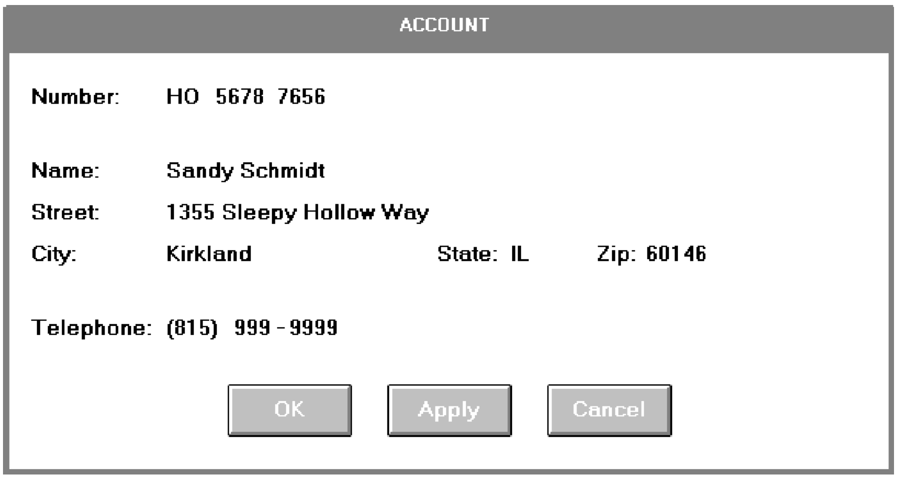


Figure 3.41: Display/read-only screen maintaining same item alignment and positioning, and data field segmentation as 3.40. Some data distinctiveness is lost and minor crowding occurs, however, because of the location of the captions for State and Zip between data fields.

https://hemanthrajhemu.github.io

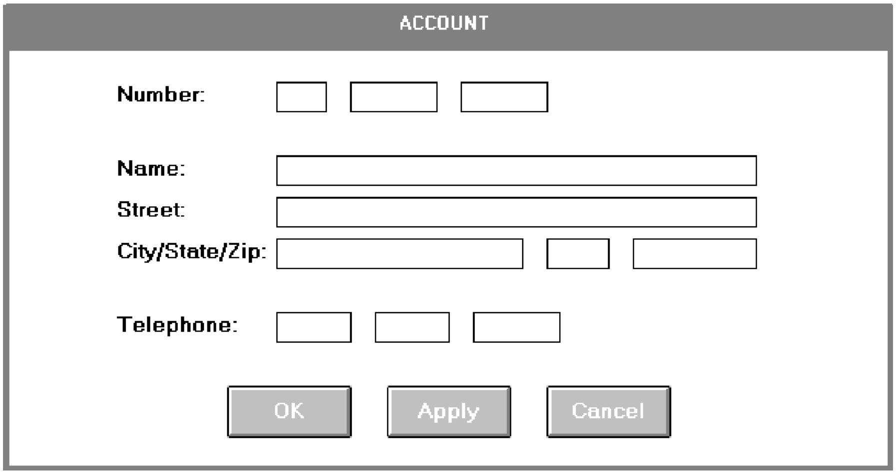


Figure 3.42: Entry/modification screen identical to 3.40 except that captions for State and Zip are stacked with City, enhancing distinctiveness and readability of the data fields. The screen also achieves a more compact and balanced look. The recommended style for this kind of entry screen.

Figure 3.43: Display/read-only screen maintaining same alignment, item positioning, and data segmentation as 3.42. Good readability but the lengthy caption City/State/Zip does impinge upon the distinctiveness for the data.

https://hemanthrajhemu.github.io

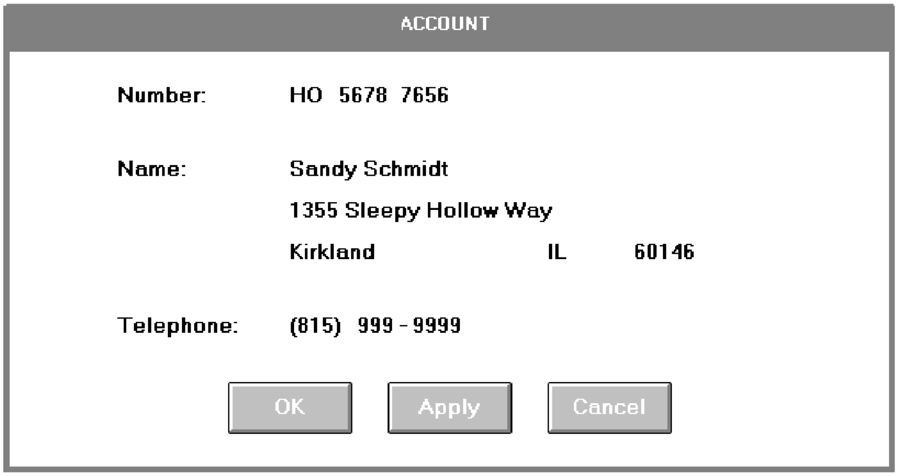


Figure 3.44: Display/read-only screen identical to 3.43 except that the captions Street and City/State/Zip have been eliminated to improve data field distinctiveness. The content of the data should make the identity of these fields obvious. The recommended style for this kind of display/read-only screen.

Justification of single captions and data fields can be accomplished in several ways. These include

A. Left-justifying captions; data field immediately follows caption.

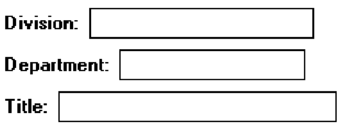


Figure 3.45

B. Left-justifying captions; left-justifying data fields; colon (:) associated with captions.

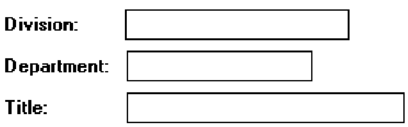


Figure 3.46

C. Left-justifying captions; left-justifying data fields; colon (:) associated with data field.

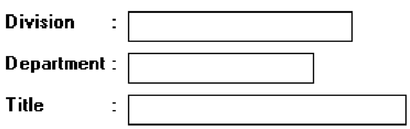


Figure 3.47

https://hemanthrajhemu.github.io

D. Right-justifying captions; left-justifying data fields.

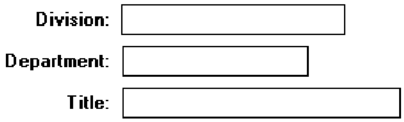


Figure 3.48

Alternatives A and C are not recommended. Alternative A, left-justified aligned captions with data fields immediately following, results in poor alignment of data fields and increases the screen's complexity. It is more difficult to find data when searching data fields. Alternative C, while structurally sound, associates the colon primarily with the data field. The strongest association of the colon should be with the caption.

The two most desirable alternatives are B and D. Alternative B, left-justified captions and data fields, is the first approach illustrated in these guidelines. Alternative D, right-justified captions and left-justified data fields, is the second approach illustrated in these guidelines.

Left-justified captions and data (B). A disadvantage to this approach is that the caption beginning point is usually farther from the entry field than the right-justified caption approach. A mix of caption sizes can cause some captions to be far removed from their corresponding data field, greatly increasing eye movements between the two and possibly making it difficult to accurately tie caption to data field. Tying the caption to the data field by a line of dots (. . . . .) solves the association problem, but adds a great deal of noise to the screen. This does not solve the eye movement problem. Eye movement inefficiencies can be addressed by abbreviating the longer captions. The cost is reduced caption clarity. An advantage to this approach is that section headings using location positioning as the key element in their identification do stand out nicely from the crisp left-justified captions.

Right-justified captions and left-justified entry fields (D). A disadvantage here is that section headings using location positioning as the identification element do not stand out as well. They tend to get lost in the ragged left edge of the captions. Advantages are that captions are always positioned close to their related data fields, thereby minimizing eye movements between the two, and that the screen takes on a more balanced look.

There is no universal agreement as to which is the better approach. Experimental studies have not provided any answers, although most style guides recommend, and illustrate, the left-aligned caption approach.

Examples to follow in this and succeeding chapters reflect both styles. This is done to enable the reader to see and evaluate each. Whichever method is chosen, however, should be consistently followed, through a series of related screens.

https://hemanthrajhemu.github.io

Headings

Used with related controls in applications, headings are primarily incorporated to create a common identity. In addition to providing meaning, they foster the concept of grouping, create visual appeal, and aid screen learning. Used with Web page text, headings are also used to break up large textual blocks, create visual appeal and aid people in scanning and finding what they are looking for.

In organizing screen controls, three kinds of headings may be incorporated: section headings, subsection or row headings, and field group headings.

Section Headings

- Provide a meaningful heading that clearly describes the relationship of the grouped controls.
- Locate section headings above their related screen controls.
- Display in a distinguishable font style and size in mixed case, using the headline style.

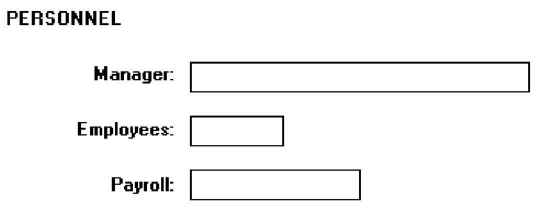


Figure 3.49

— Alternately, headings may be located within a border surrounding a grouping, justified to the upper-left corner.

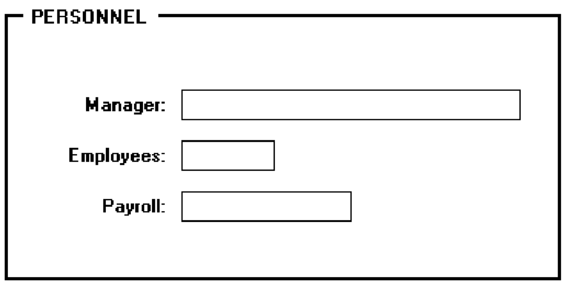


Figure 3.50

- Indent the control captions to the right of the start of the heading.
- Fully spell out in an uppercase font.
- Display in normal intensity.
  — Alternately, if a different font size or style exists, the heading may be displayed in mixed case, using the headline style.

https://hemanthrajhemu.github.io

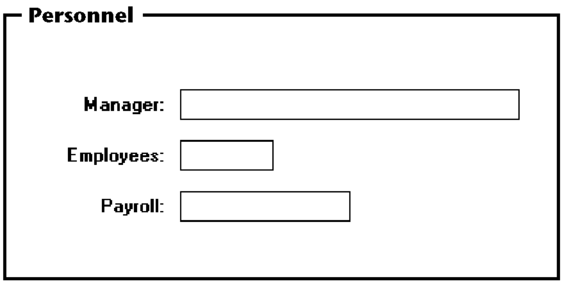


Figure 3.51

Section headings should be visually distinguishable through a combination of location and font style. They should not be overly emphasized, however. In the past, many products have displayed headings in the same type size and style as control captions. This provides very poor differentiation between captions and headings, each equally competing for the viewer's attention. Visual emphasis to section headings in applications should be moderate. Bolding should be reserved for the more important screen data.

Display section headings in an easily distinguishable font. Use mixed case of the headline style (capitalization of all significant words). The method employed should always permit easy, but subtle, discrimination of the section headings from other components of the screen. It should also be visually compatible with other screen components. Whatever styles are chosen, they should be consistently followed throughout a family of screens or a system.

Subsection or Row Headings

- Provide a meaningful heading that clearly describes the relationship of the grouped controls.
- Locate to the left of the
  — Row of associated fields.
  — Topmost row of a group of associated fields.
- Separate from the adjacent caption through the use of a unique symbol, such as one or two greater-than signs or a filled-in arrow.
- Subsection or row headings may be left- or right-aligned.
- Display in a distinguishable font style and size in mixed case, using the headline style.

Figure 3.52

Row or subsection headings may be positioned to the left of a group of related controls. A meaningful convention to designate subsection or row headings is a filled-in arrow or greater-than sign. It directs the viewer's attention to the right and indicates that everything that follows refers to this category. Space should separate different subsections. They may also be right-aligned instead of left-aligned, as shown in Figure 3.53. Display row or subsection headings in an easily distinguishable font style in mixed case using the headline style.

https://hemanthrajhemu.github.io

AUTO >
REGISTRATION >
DRIVER >

AUTO >
REGISTRATION >
DRIVER >

Figure 3.53: Left-aligned and right-aligned row headings.

Field Group Headings

- Provide a meaningful heading that clearly describes the relationship of the grouped controls.
- Center the field group heading above the captions to which it applies.
- Relate it to the captions by a solid line.
- Display in a distinguishable font style and size in mixed case, using the headline style.

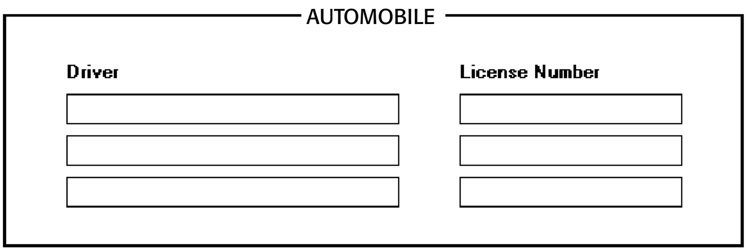


Figure 3.54

Occasionally a group heading above a series of multiple-occurring captions may be needed. It should be centered above the captions to which it applies and related to them through a solid line extending to each end of the grouping. This will provide closure to the grouping. Display row or subsection headings in an easily distinguishable font style in mixed case using the headline style.

MYTH **All we really have to do is make the interface look glitzy.**

Special Symbols

- Consider special symbols for emphasis.
- Separate symbols from words by a space.

DELEGATES >>

Figure 3.55

https://hemanthrajhemu.github.io

Special symbols can be considered to emphasize or call attention to elements on a screen. An icon, for example, can precede an error message, or the greater-than sign can be used to direct attention (DELEGATES >>). Symbols should be separated from words by one space.

Instructions

- Incorporate instructions on a screen, as necessary
  - — In a position just preceding the part, or parts, of a screen to which they apply.
  - — In a manner that visually distinguishes them, such as
    - Displaying them in a unique type style.
    - Displaying them in a unique color.
  - — In a position that visually distinguishes them by
    - Left-justifying the instruction and indenting the related captions, headings or text to the right.
    - Leaving a space line, if possible, between the instruction and the related control, heading, or text.

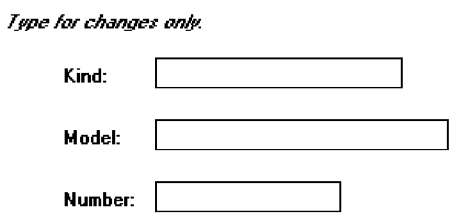


Figure 3.56

  - — Using a mixed-case font.

Instructions to the screen user on what to do with, or how to work with, the screen presented are occasionally necessary. Whether or not to include them will be dependent upon the experience of the user, the frequency of screen use, and the nature of the information itself. Inexperienced or occasional users may need instructions; data that is complex or unfamiliar may also require them. For experienced and frequent screen users, instructions can quickly become visual noise. When you are deciding whether or not to include instructions on a screen, other techniques, such as using Help or the message area should also be considered.

When it is necessary to place instructions on a screen, they must be positioned at the screen point where they are applicable. Instructions placed at the bottom of a screen will probably not be seen. Instructions placed on one screen but applying to another will never be remembered.

When it is necessary to place instructions on a screen, they must be visually recognized as instructions. This will allow them to be easily ignored by the user when they are not needed. Therefore, some visual aspect of the instruction must indicate that it *is* an instruction. Designers of paper forms do this by presenting instructions in a different font kind or font style such as italics. The form user then immediately recognizes them as instructions, and they can be read or ignored as desired.

https://hemanthrajhemu.github.io

To make instructions immediately recognizable on a screen, they may also be presented in a unique font or color. If one of these methods is used, however, cautions concerning the excessive use of different font styles (and colors, as are shown in Step 12) must be heeded. Another, but less visually strong, technique is to identify the technique simply by its location. Begin the instruction to the left of the screen elements to which it applies; the left-justification identifies it as an instruction.

Instructions should be presented in the normal mixed-case sentence style. Guidelines for writing text, including instructions, are discussed in Step 8.

Completion Aids

- Incorporate data field completion aids on a screen, as necessary:
  - — In a position to the right of the text entry control to which they apply.
  - — In a manner that visually distinguishes them, including
    - Displaying them within parentheses ( ).
    - Possibly displaying them in a unique font style.
  - — If the controls are arrayed on the screen in a columnar format, position the completion aid, or aids
    - Far enough to the right so as to not detract from the readability of the entry controls within the column.
    - But close enough to the related control so that they easily maintain an association with the related control.
  - — Left-alignment of completion aids in a column of controls is desirable but not absolutely necessary.

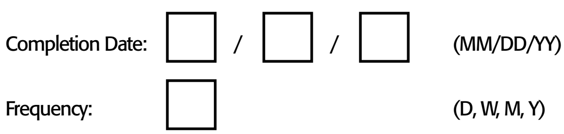


Figure 3.57

Completion aids are a form of instruction, but they are directed to the contents of a specific entry field control and the content's format. A date, for example, may require entry of a specified number of characters in a specific order, and it may be necessary to present on the screen a reminder of this format for key entry.

As with instructions, the decision whether or not to include text entry control completion aids will be dependent upon the experience of the user, the frequency of screen use, and the nature of the information itself. Inexperienced or occasional users may need aids; data that is complex or unfamiliar may also require them. For experienced and frequent screen users, however, aids can quickly become visual noise. In deciding whether or not to include completion aids on a screen, other techniques, such as using Help or the message area should also be considered.

https://hemanthrajhemu.github.io

When it is necessary to place entry control completion aids on a screen, they must be recognized as such. This will allow them to be easily ignored when they are not needed. Therefore, some visual aspect of the completion aid must indicate that it is an aid.

To make completion aids immediately recognizable on a screen, display them within parentheses ( ). A distinguishing font may also be used but parentheses are visually strong enough to stand by themselves, providing an adequate indication that what is contained within is a completion aid.

The best location for a completion aid is to the right of the entry control that it applies to. Right positioning optimizes the screen layout for the expert user by placing the aid outside of the "working area" of a group of columnized controls. Alternate positioning, such as placing the aid within the caption itself, pushes the caption farther away from the entry control, and for the expert this is less efficient and also creates visual noise. Placing the aid above or below the entry control detracts from the readability of the entry control fields, creates an association problem (Is the aid related to the field above or below?), and yields a less efficient screen organization. For the novice or infrequent user, positioning the aid to the right of the entry field is less efficient because his or her eyes must move right to read it, but these kinds of users will be less efficient, anyway.

In a columnized group of controls, position completion aids far enough to the right so as not to detract from the readability of all the entry controls contained in the column. Positioning, however, must be close enough to the related control so that the aid easily maintains an association with its related control. Left-alignment of completion aids in a column of controls is desirable but not absolutely necessary, since the sizes of entry fields may vary significantly. Final positioning of the completion aid must balance all the above factors.

Required and Optional Data

- Use required fields sparingly.
- Request required information at the necessary point in the process.
- Provide defaults for previously captured information.
- Permit unfinished applications to be saved.
- Designate required fields in a standard and consistent way.
- Provide polite feedback to request missing required data.

Required information is information that must be provided by the screen user before the screen's contents will be accepted by the system. Information is specified as required if it is necessary to the successful completion of an application. Required information must also be complete and valid.

Asking for too much information can be frustrating to a casual user of an application or system. It can be especially frustrating if the user cannot see the value of the requested information or knows that the information has been provided before or could easily be calculated or created by the system. Required fields can also create problems because:

https://hemanthrajhemu.github.io

- Inexperienced users may not understand the difference between required and optional data.
- The information may not be completed because the amount of data seemingly requested seems excessive or daunting.
- Some information incorrectly assumed as being required may be considered as a threat to privacy.

Problems can also occur if people don't have, or can't find, the correct data to put in a required field. A solution to this problem may be to enter anything the field will accept.

Use sparingly. Only designate fields as required when absolutely necessary. Never make a field mandatory unless it truly is.

Request at necessary points. In organizing application information, only request required information at the points in the data collection process where the information becomes relevant. Why collect shipping information, for example, before the user has made a decision whether or not to purchase a product?

Provide defaults. Provide defaults based upon what is already known about the user. Known information may already be stored in a database, or be predictable based upon a user's address or age, for example.

Permit saving. Allow the user adequate time to find and enter required information. Determine in task analysis how much time it may take to find and enter the required information. Do not establish quick time-outs, destroying everything the user has already entered. Allow the user to save unfinished applications until the necessary information can be found.

Designate consistently. Provide indications of required fields an obvious way. Common methods today include displaying symbols such as asterisks (*), check marks, or chevrons (>>) at the beginning of the caption or data field. Other methods include inscribing, the word "Required" near the field caption, displaying captions in a unique color, and displaying the caption bold. A distinctive completion aid message must, however, be prominently displayed at the beginning of the data fields for methods other than the *Required* word method. Be cautious in the use of *Required* because visual noise is added to the screen. A different color as an indication of a required field can cause a problem with people who have a color-viewing deficiency. One study (Tullis and Pons, 1997) found that bolded text was preferred to the use of chevrons, check marks, or color. Frequent users of applications do not have to be continually reminded with messages that required data fields are indicated in a certain way. They will be learned. However, for casual users required fields will have to be designated clearly. Unfortunately, people do not always read these messages, so...

Provide polite feedback. Feedback for omissions should be non-threatening, non-chastizing, and clearly indicate the erroneous data field. Message writing is discussed in Step 8. Feedback is discussed in Step 9.

https://hemanthrajhemu.github.io

Lists

- Present a collection of related items in a vertical list.
- Use sentence or headline style capitalization in a consistent manner.
- Provide a heading for each list.
- Order lists in a meaningful way.
  — For items of equal value with no discernable order, arrange items alphabetically or designate each item with a bullet.
  — If important or frequently chosen items exist, place at the top of the list.
  — For items possessing a particular order, identify each with a number beginning with one (1).
- Format lists for easy identification and scanning through use of surrounding borders, groupings, and white space.

Create lists. Application or Web page information can easily be formatted into lists. An informational list, when compared to paragraph style textual presentation, greatly reduces page density and permits much faster and easier scanning of its contents. Array lists vertically as illustrated in Figure 3.58. One study found that scanning a horizontal list takes people twenty percent longer than scanning a vertical list (Koyani et al., 2004).

Word capitalization. Either sentence style (first word capitalization) or headline style (all significant words capitalized) may be used. The method chosen should be consistently followed. Lower-case-only listings should be avoided.

Headings. Provide a descriptive heading for each list, as illustrated in Figure 3.59. This will help people understand the reason for the list and how the items are related.

ARRAY LIKE THIS:	NOT LIKE THIS:			
Amsterdam	Amsterdam	Berlin	Brussels	Lisbon
Berlin	London	Madrid	Paris	Vienna
Brussels				
Lisbon				
London				
Madrid				
Paris				
Vienna				

Figure 3.58: List Formats.

https://hemanthrajhemu.github.io

Figure 3.59: A List Heading.

Ordering lists. Establish a list ordering that is most meaningful to the user. If a list contains important or frequently used items, locate them at the top of the list as illustrated in Figure 3.60. Studies show that experienced users usually look at the top item in a list first, and almost always look at the first three items before scanning down the list. Research also indicates that users stop scanning a list as soon as they see something relevant, reinforcing the need to place important items at the top (Koyani et al., 2004). Provide a consistent ordering scheme on all lists with similar content.

For list items of equal value, designate each item with a bullet or arrange alphabetically. For list items possessing a particular order, identify each item with a number beginning with the number 1, not 0. Numbered lists are especially important when the listing is a series of instructions.

Formatting. A list should be easily identifiable as a list. Display techniques aiding a list being readily identified as a list include its vertical structure, a surrounding border, consistent backgrounds, a sufficient amount of surrounding white space, and appearance consistency between lists. See Figure 3.61. Scanning is aided by use of fonts of the proper type, size, and styles, and good character-background contrast.

Figure 3.60: Most important or frequently chosen list items at top.

https://hemanthrajhemu.github.io

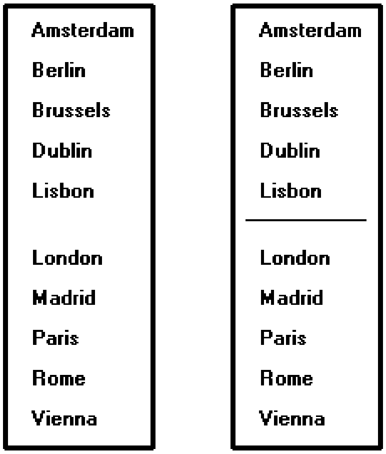


Figure 3.61: Lists with border and white space and divider rule separators.

Keying Procedures

For large-volume data entry applications substantial keying may still be required. The following must be considered in establishing keying procedures.

Keystrokes

- Do not focus on minimizing keystrokes without considering other factors such as
  — The keying rhythm.
  — The goals of the system.

A sought-after goal in many past data entry applications has always been to minimize keystrokes. Fewer keystrokes have been synonymous with faster keying speeds and greater productivity in the minds of many practitioners. But this is not always true. Fewer keystrokes may actually decrease keying speeds and reduce productivity in many cases.

One research study compared manual tabbing with auto skip in a data entry application. Auto skip, while requiring fewer keystrokes, was found to result in longer keying times and more errors than manual tabbing because it disrupted the keying rhythm. This study is described in more detail in the following section.

Another study, in an information retrieval task, compared input keystrokes to the time needed to evaluate the system output. They found that more keystrokes yielded more meaningful inputs. This yielded more precise and informative outputs, which resulted in faster problem solving.

So the number of keystrokes, and selections, must be considered in light of keying rhythms and the objectives of the system. Fewer are not necessarily always better.

https://hemanthrajhemu.github.io

Tabbing

- Initially, position the cursor in the first field or control in which information can be entered.
- Tab in the order in which the screen's information is organized.

When a screen is first presented, the cursor must be positioned in the first field or control in which information can be entered. Tabbing order must then follow the flow of information as it is organized on the screen.

Manual Tab versus Auto Skip

- Define fields to permit manual tabbing.
- Consider using Auto Skip for
  — Expert users.
  — Easily learned entry screens.
  — Screens containing fields always completely filled.
  — Moving within common fixed-length fields segmented into parts.

Auto skip is a feature that causes a cursor to automatically move to the beginning of the next text entry control field once the previous field is completely filled. Auto skip obviates manual tabbing and requires fewer keystrokes to complete a screen. Theoretically, keying speeds should increase with auto skip. In practice, they do not always do so.

Rarely are many text entry screen fields completely filled to their maximum length with data. When an entry field is not full, the user must still press the tab key to move the cursor to the next entry field. Figure 3.62 illustrates auto skip functioning.

Auto skip, therefore, imposes decision-making and learning requirements. After keying text into each field, one must determine where the cursor is and whether to press the tab key or not to go to the next field. Only then can the next keying action be performed. As illustrated in Figure 3.63, manual tabbing requires extra keystrokes but no decisions must be made. The keying task is rhythmic. One study, comparing auto skip with manual tabbing, found that manual tabbing resulted in faster keying performance and fewer keying errors.

Auto skip can delay detection of one particular kind of error. If an extra character is inadvertently keyed into a field, the cursor will automatically move to the next field while keying continues. The error may not be immediately detected, and spacing in subsequent fields may also be one position off, at least until the tab key is pressed. Were this situation to occur while using manual tabbing, the keyboard would lock as soon as the entry field was full. The error would be immediately detected.

Auto skip, despite its limitations, can be useful for expert users who have learned field structures or if all screen fields are always completely filled. Common fixed-length fields segmented into logical parts may be auto tabbed between the parts (for example, a segmented social security number or a standard segmented numeric date). Nevertheless, most high-volume data entry applications would not appear to meet these criterion.

https://hemanthrajhemu.github.io

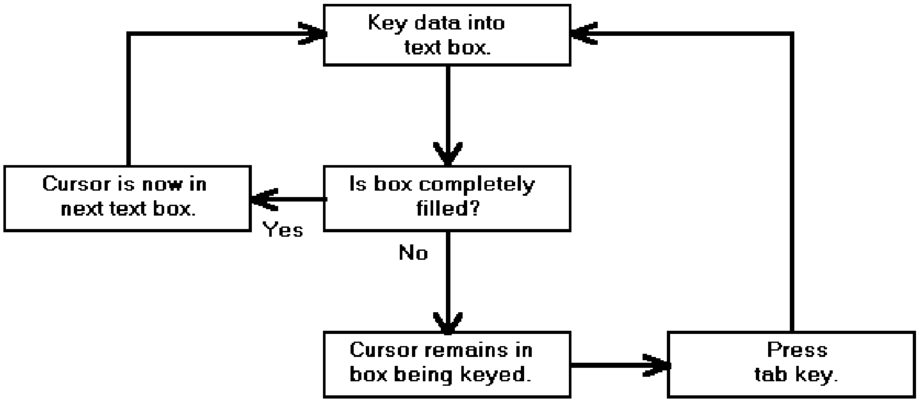


Figure 3.62: Text entry using auto skip.

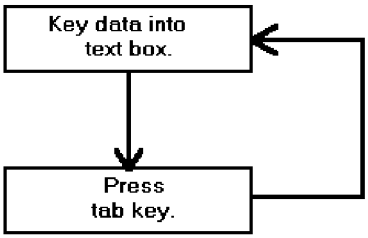


Figure 3.63: Text entry using manual tabbing.

Keying Rules

- Do not require recoding, changing, omitting, or including data based on special rules or logical transformations.
- Do not require formatting of data.
- Do not make keyed codes case-sensitive.
  — Exception: Passwords.
- Do not require units of measurement to be keyed.
- Minimize use of the Shift key.
- Ensure that double-clicking will not cause problems.

In large-volume entry applications, decisions that must be made during the keying process impose learning requirements and greatly slow down the entry process. The fewer rules and decisions involved in keying, the faster and more accurate entry will be. Coding, omitting, changing, and including data by special rules or transformations as a group represent probably the greatest single decrement to data entry speed.

https://hemanthrajhemu.github.io

Requiring the formatting of data should also be avoided. The software should

- Left-justify alphabetic data.
- Right-justify numeric data.
- Justify the entry around the decimal point if it is decimal.

Upper- and lowercase letters should be treated as equivalents when codes are keyed, unless there is a valid reason for not doing so (for example, secure passwords). If codes must be entered in a case-specific manner, inform the user using a field Completion Aid. When storing user-keyed data, show the data as it was keyed.

Never require units of measurement to be keyed because keystrokes will be reduced and keying speed increased. Incorporate all units of measurement such a pounds, inches, etc. within the caption or specify in a Completion Aid. Entry of characters that require the use of the Shift key should be avoided. To use the Shift key requires extra user attention and is prone to errors.

Often people double-click when only one click is needed. If an erroneous double-click is made and both are detected by the computer, a double action can possibly be implemented. This should be prevented from happening.

Data Output

Data output includes

- *Reports*, a list of records designed to be printed out.
- *Tables*, printouts of a listing of records derived from database queries.

Being able to print out a table or report is a very common user requirement. Fowler and Stanwick (2004), upon whom many of the following guidelines are based, describe paper printouts as advantageous because

- Items are easier to check off or mark than items on a screen.
- Items can all be seen at once. Cumbersome scrolling is not necessary.
- Easier discussion of the information with other people or at meetings is permitted.
- Printouts can be easily perused on a bus, train, or airplane.
- Printouts provide hard-copy documentation and a more visible audit trail.
- A more visible sense of completion or closure results.
- A printout is a better external memory than a screen.

In regards to the latter, Sellen and Harper (2004) in an analytical study, confirmed obvious human behavior. When people write a report they study, highlight, and annotate printouts, spread printouts out over work surfaces, and refer back to the printouts continually. Printed copies accommodate these activities, and human memory limitations, much more readily than screen versions do.

https://hemanthrajhemu.github.io

Reports

- Report Body
  - — Provide clear column headings.
  - — Show units of measurement.
  - — Use the right fonts.
  - — Clearly identify rows.
  - — For a body that is too wide:
    - Let users move or resize the columns.
    - Wrap information in a column of cells.
  - — Break up pages logically.
- For headers:
  - — Minimally include:
    - Print date.
    - What or whom the report is for.
    - Title.
  - — Consider including:
    - Report or file parameters.
    - Print or retrieval time.
    - Logo and other organization identity items.
- For footers:
  - — Minimally include:
    - Current Page Number.
  - — Consider including:
    - Number of pages in report.
    - Print date (if not in the header).
    - Data source (file, database or table name).
    - Report format name.
    - URL or other location information.
    - Legal information.
  - — Repeat at the bottom of every page.

Body. A report body usually contains rows of columnized data and column headings. Page and data breaks may also require inclusion of subheadings, subtotals, or summaries at each break. Design guidelines for a report body are as follows.

Column headings. Column headings must clearly describe the column's contents. Avoid abbreviations in column headings whenever possible. If an abbreviation is necessary, use a ToolTip to spell out an abbreviation. (ToolTips are described in Step 7.)

Units of measurement. Some types of data can be described by different units of measurement, for example: inches or centimeters; miles, or kilometers? If there is any possibility of ambiguity, always include the unit of measurement in a column heading. Data interpretation errors will be prevented.

https://hemanthrajhemu.github.io

Fonts. Alignment of numeric data is important for any report containing numbers. Data comparisons will be easier to perform, and errors easier to detect. In reports, numbers in columns should always be presented right-aligned using monospaced fonts. A monospaced type is fixed-width, each number possessing the same width. The standard PC monospaced typeface is Courier. Monaco is the standard on the Macintosh. Courier, however, is a very wide type. The numbers in some of the common software fonts are always presented as monospaced, even though the letters are not. Examples include Arial, Times New Roman, and Verdana. Always use proportional fonts (variable-width letters depending on character width) for textual information because monospaced text will consume about 30 percent more real estate on a screen.

Rows. Visually distinguish individual rows by displaying alternate rows in a light color. As described previously in *Tables*, Tullis (2003) found that alternate-row shading yielded the best performance (speed and accuracy of response) and was rated as the best alternative by study participants. If alternate-row spacing cannot be implemented, or if more than seven rows are presented, insert extra white space after every fifth row.

Wide body. If a report is too wide to be completely seen on a screen, or printed on a paper, it may be narrowed in a couple of ways. One solution is to allow users to move, shrink, or expand columns so all information, or at least the important information, can be seen simultaneously. While some columns may have to be fixed in size, many can be modified. Another solution is to wrap a wide cell's content into two lines, as illustrated in Table 3.2.

Page breaks. For multipage reports, break pages at logical points in the presented information.

Headers. Headers usually contain information about what is contained in the printout. Minimally include the date the report is printed, or, for onscreen reports, the retrieval date. Also include the name of the intended recipient and the report title. Recipient name and title may be wrapped, if necessary, on more than one line. Optional information that may be included are report or file parameters (such as a date range), print or retrieval time, and organization identity graphics and logos.

Footers. Footers are strictly for reference, containing information about the printout itself. As such, they should not draw attention to themselves. A smaller non-bold, non-italic font is recommended. Minimally, a footer should contain a page number. Optional information that may be included are print date (if not in header), the data source (file, database or table name), the report format name so that the same format can be found to update the report, URL or other location information, and legal information such as copyrights, and so on. Repeat the footer at the bottom of each page. If a footer is rarely printed, it may not be necessary to create a specialized footer. If printed, the browser will add page numbers to the bottom and the URL to the top by default.

https://hemanthrajhemu.github.io

Tables

Large amounts of information that must be viewed and compared can be displayed in a table. Like a data entry grid, a table is a matrix of information arranged in columns and rows. Tables usually consume less screen space than do individual data elements. A table may be created from a query but may also be fixed in format.

- Table Title and Headings:
  - — Provide a clear and descriptive title, headings and, where appropriate, subheadings for columns and rows.
    - Do not include colons (:) after the headings.
    - Show units of measurement.
  - — Justify column headings according to the data presented in the table cells.
    - Left-justify for columns containing text.
    - Right-justify for columns containing numbers.
  - — Left-justify row headings.
- Table Data or Information:
  - — Organize the presented data or information logically and clearly.
    - Place similar information together.
    - Place most important or frequently used at the top.
    - Arrange chronologically or sequentially.
  - — Justify the data presented in a column according to its content.
    - Left-justify textual data.
    - Right-justify numeric data.
    - To reduce table width, wrap information in a column of cells.
- Table Format:
  - — Provide alternate-row shading.
  - — Use light backgrounds.
  - — Highlight a particular cell, column, or row using a contrasting display technique.
    - Avoid scrolling, if possible.
- Printouts:
  - — The entire record listing must be printed, not simply what is currently visible.
  - — Users must be able to modify the font size and type and/or column width so the printout completely fits on the available printer paper.
  - — The printout must maintain the users choices of columns, column locations, and sort, not resetting these elements to a default.
  - — The printout must have a header and footer.

A *table*, illustrated in Table 3.1, should possess the following qualities:

Descriptive headings. Provide clear and descriptive headings and, where necessary and appropriate, subheadings for columns and rows. Do not include colons (:) after the headings. Avoid abbreviations in column headings whenever possible. If an abbreviation is necessary, use a ToolTip to spell out an abbreviation. (ToolTips are described in Step 7.) Some types of data can be described by different units of measurement (for example, inches or centimeters, miles or kilometers?). If there is any possibility of ambiguity, always include the unit of measurement in a heading. Data interpretation errors will be prevented.

https://hemanthrajhemu.github.io

Heading justification. Justify column headings according to the data presented in the table cells. Use left-justification for columns containing text and right-justification for columns containing numbers. Row headings should be left-justified.

Table organization. Organization of table data or information will follow the general organization guidelines described previously. Organize table data logically and clearly so it can be quickly identified, scanned, and compared. Place most important or frequently used information at the top, place together information that must be directly compared, and arrange it chronologically or sequentially.

Data justification. Justify the data presented in a column according to its content. For textual data, use left-justification; for numeric data, use right-justification. Alignment of numeric data is important for any report containing numbers. Data comparisons will be easier to perform, and errors easier to detect. In reports, numbers in columns should always be presented right-aligned using monospaced fonts. A monospaced type is fixed-width, meaning each number possesses the same width. The standard PC monospaced typeface is Courier. Monaco is the standard on the Macintosh. Courier, however, is a very wide type. The numbers in some of the common software fonts are always presented as monospaced, even though the letters are not. Examples include Arial, Times New Roman, and Verdana. Always use proportional fonts (variable-width letters depending on character width) for textual information because monospaced text will consume about 30 percent more real estate on a screen.

Table width. To reduce the width of a table, a cell's content can be wrapped into two lines, as illustrated in the "LANGUAGES" column in Table 3.2.

Alternate-row shading. Visually distinguish individual rows by displaying alternate rows in a light color. Tullis (2003) evaluated three methods of distinguishing rows and/or columns: (1) alternate-row shading using alternating white and light gray backgrounds; (2) horizontal and vertical line row and column separators; and (3) horizontal line separators only. These methods were compared with a fourth alternative: space as the only separator. The finding — in the visual search task alternate-row shading yielded the best performance (speed and accuracy of response) and was rated as the best alternative by study participants. Borders apparently interfere with scanning and readability. If alternate-row spacing cannot be implemented, or if more than seven rows are presented, insert extra white space after every fifth row.

Table 3.1: A Table

EUROPEAN UNION NEW MEMBER STATES IN 2004

COUNTRY	CAPITAL	GOVERNMENT	POPULATION	AREA (SQ.MI.)
Czech Republic	Prague	Republic	10,320,000	30,450
Estonia	Tallinn	Republic	1,450,000	17,413
Hungary	Budapest	Republic	9,963,000	35,920
Latvia	Riga	Republic	2,452,000	24,595

https://hemanthrajhemu.github.io

Table 3.1 *(continued)*
EUROPEAN UNION NEW MEMBER STATES IN 2004

COUNTRY	CAPITAL	GOVERNMENT	POPULATION	AREA (SQ.MI.)
Lithuania	Vilnius	Republic	3,639,000	25,174
Poland	Warsaw	Republic	38,915,000	120,728
Slovakia	Bratislava	Republic	5,385,000	18,933
Slovenia	Ljubljana	Republic	1,947,000	7,819

Highlighting cells. If highlighting a cell or cells is necessary, use a contrasting display technique that does not diminish the legibility of the displayed data.

Avoid scrolling. Avoid long vertical tables if possible. A table should be entirely visible for readability and making data comparisons.

Printouts. Like reports, a table to be printed must be capable of being resized for paper and possess a header and footer. Additionally, non-visible elements must also be printed and the printout must maintain user choices, not resetting them to a default.

Table 3.2: A Table with Wrapped Cells
EUROPEAN UNION NEW MEMBER STATES IN 2004

COUNTRY	CAPITAL	GOVERNMENT	POPULATION	LANGUAGES
Czech Republic	Prague	Republic	10,320,000	Czech, Slovak
Estonia	Tallinn	Republic	1,450,000	Estonian, Russian
Hungary	Budapest	Republic	9,963,000	Hungarian
Latvia	Riga	Republic	2,452,000	Latvian, Russian
Lithuania	Vilnius	Republic	3,639,000	Lithuanian, Russian, Polish
Poland	Warsaw	Republic	38,915,000	Polish
Slovakia	Bratislava	Republic	5,385,000	Slovak, Hungarian
Slovenia	Ljubljana	Republic	1,947,000	Slovene

https://hemanthrajhemu.github.io

Organization and Structure Guidelines

What follows is a series of organization and structure guidelines for specific kinds of screens. They are Information Entry and Modification (Conversational), Entry from a Dedicated Source Document, and Display/Read-Only screens.

Information Entry and Modification (Conversational) Screens

- Organization:
  - — Logical and clear.
  - — Most frequently used information:
    - On the earliest screens.
    - At the top of screens.
  - — Required information:
    - On the earliest screens.
    - At the top of screens.
- Captions:
  - — Meaningful.
  - — Consistently positioned in relation to data field controls.
  - — Left- or right-aligned.
  - — Mixed case using headline style.
- Text boxes/selection controls:
  - — Designate by boxes.
- Spacing and groupings:
  - — Create logical groupings.
  - — Make them medium in size, about five to seven lines.
- Headings:
  - — Uppercase or headline-style mixed case.
  - — Set off from related controls.
- Control arrangement:
  - — Align into columns.
  - — Organize for top-to-bottom completion.
- Required and optional input:
  - — Consider distinguishing between required and optional data input through
    - Placing required and optional information within different screens, windows, or groups.
    - Identifying information as required or optional in a completion aid.
    - Identifying required information with a unique font or symbol.
- Instructions and completion aids:
  - — Include as necessary.
    - Position instructions before the controls to which they apply.
    - Position completion aids to the right of the controls to which they apply.
- Use a Grid for entering large amounts of data or information.

Information entry and modification (conversational) screens are used to collect and modify information, either by entry or selection. These screens are sometimes referred

https://hemanthrajhemu.github.io

to as conversational screens. They guide a person through a task or process. The screen itself is the user's focal point for working with information. The viewer is driven by what is presented on the screen in the information collection and designation process. The information needed to complete a screen may be collected from, but is not limited to, these kinds of sources:

- A person being interviewed or queried at a desk or workstation.
- A person being queried or interviewed over a telephone.
- A collection of notes and written materials.
- An unstructured form.
- The mind of the user.

These guidelines summarized here have been presented earlier in this chapter.

Organization. Organize these screens logically and clearly, reflecting the exact information needs of the user for the task being performed. In general, place the most frequently used information, or required information, on the earliest screens and at the top of screens.

Captions. Provide meaningful captions, clearly identifying the information to be entered or selected. Use the headline style to display them (all significant words capitalized). Consistently position all captions in relation to their associated controls. They may be left- or right-aligned.

Text boxes/selection controls. Designate by boxes, using either a line border or polarity reversal.

Spacing and groupings. Create logical medium-size groupings of from five to seven elements.

Headings. Provide headings to identify groupings. Set off from their related controls using uppercase or mixed-case headline style.

Control arrangement. Align controls into columns. Maintain a top-to-bottom, then left-to-right arrangement.

Required and optional input. Distinguishing between required and optional data input may or may not be necessary on these screens. The decision on whether or not to distinguish these types of data should be based on the experience of the user doing the key entry, and the information's familiarity. When a technique to distinguish them is included on a screen, it is a form of completion aid, so the arguments for and against completion aids are applicable here as well. When it is necessary to differentiate required and optional data, consider the following alternatives. First, determine the feasibility of placing the two kinds of data on separate screens or within separate screen windows or groupings. This is the best and cleanest solution. If a meaningful screen organization of information will not permit this, then describe the individual pieces of data as required or optional within a completion aid. The last choice is to identify required information with a unique font or symbol. This alternative, however, requires the user to learn the convention to use it effectively. Displaying a unique font might also lead to screen clutter, if too many different fonts and styles are used on the screen.

Instructions and completion aids. Include these on screens as necessary. Locate instructions so they precede the controls to which they apply. Locate completion aids to the right of the controls to which they apply.

https://hemanthrajhemu.github.io

Grids

- Usage:
  - — To enter large amounts of related data or information.
- Design guidelines:
  - — Provide descriptive headings and, where appropriate, subheadings for columns and rows.
    - Do not include colons (:) after the headings.
  - — Justify column headings according to the data presented in the table cells.
    - Left-justify headings for columns containing text.
    - Right-justify headings for columns containing numbers.
  - — Left-justify row headings.
  - — Organize the data or information to be entered logically and clearly.
    - Place similar information together.
    - Place most important or frequently used information at the top.
    - Arrange information chronologically or sequentially.
  - — Grid format:
    - Provide alternate-row shading.
    - Use light backgrounds.
    - Avoid scrolling.

Usage. Large amounts of related data that must be entered can be collected in grids. Like a table, a grid is a matrix of entry fields arranged in columns and rows. Grids usually consume less screen space than do individual data elements.

Headings. Provide descriptive headings and, where necessary and appropriate, subheadings for columns and rows. Do not include colons (:) after the headings. Justify column headings according to the data presented in the table cells. Use left-justification for columns containing text, right-justification for columns containing numbers. Row headings should be left-justified.

Organization. The organization of grid data or information will follow the same general organizational guidelines previously described. Organize grid data logically and clearly. Place most important or frequently used information at the top and arrange it meaningfully.

Rows. Visually distinguish individual rows by displaying alternate rows in a light color. As described previously, Tullis (2003) found that alternate-row shading yielded the best performance (speed and accuracy of response in finding relevant information) and was rated as the best alternative by study participants. Including alternate-row shading in a data entry task will aid the user in finding information when reviewing screen content. If alternate-row spacing cannot be implemented, or if more than seven rows are presented, insert extra white space after every fifth row.

Backgrounds. For legibility, present table data on light backgrounds, either off-white or light gray. Use black for the data or information.

Avoid scrolling. Avoid long vertical grids requiring scrolling. A table grid should be entirely contained within the borders of one screen.

https://hemanthrajhemu.github.io

Text Entry from a Source Document

Occasionally, it may be necessary to key information directly from a source document or paper form into a screen. The document may take the form of an application for insurance, an application for a bank loan, a request for service, and so forth. The key issue for this function is that the document should be *dedicated* to the screen, and permit "head down" entry of data from the document to the screen, with the screen seldom being the point of the user's attention. An entire screen should be capable of being completed without the keyer ever looking at the screen. The design guidelines are based upon this assumption. Ideally, the document and screen should be created together so that a document-screen image relationship can be easily achieved. Creating them together permits trade-offs between the document and the screen to achieve this fit.

Sometimes, but not often, an existing document will allow the creation of a screen in its exact image. When this happens, this document can also be considered as dedicated and will follow these rules. Most documents, however, because they were not designed with a screen in mind, cannot be easily matched to a screen. In this case, their corresponding screens should be considered as entry/modification (conversational) screens and should be designed accordingly because the screen, not the document, will drive the keying process. The required information on the screen must be searched for and found on the form.

If an existing document or form is being converted to a screen format, and the existing document will no longer be used, its screens should also be designed following the entry/modification (conversational) guidelines. This is a much more effective approach for information collection, as discussed earlier in this chapter.

Dedicated Source Document Screens

- Organization:
  - — Image of associated source document.
- Captions:
  - — Abbreviations and contractions.
  - — Consistently positioned in relation to data fields.
  - — Right-aligned.
- Data fields:
  - — Designate by boxes.
- Spacing and grouping:
  - — Logical groupings found on source document.
- Headings:
  - — Include if on source document.
  - — Uppercase or headline-style mixed case.
  - — Set off from related controls.
- Control arrangement:
  - — As arranged on source document.
  - — Left-to-right completion.

https://hemanthrajhemu.github.io

- Keying procedure:
  — Use manual tabbing.
- Required and optional input:
  — Not necessary to differentiate.
- Instructions and completion aids:
  — None needed.

Organization. The screen must be an exact image of its associated source document. Skipping around a source document to locate information adds a significant amount of time to the keying process. It also imposes a learning requirement on people, because the order and location of screen fields must be mastered. Having the source document and screen in the same image eliminates these problems. Cursor position on the screen is always known because it corresponds with a person's eye position on the source document.

Captions. To allow the screen to be in the image of a source document, screen captions usually must consist of abbreviations and contractions. The document will always be available to assist in identifying unclear captions. Because text boxes fit fairly tightly on these kinds of screens, captions must be right-aligned so they are associated with the proper box.

Data fields. Designate by boxes, using either a line border or background color.

Spacing and grouping. Create the same groupings as exist on the document. Set off groupings as this is done on the form, through use of either white space and/or borders.

Headings. Include the same headings as are found on the source document. Capitalize or use mixed-case headline style (all significant words capitalized) to set them off from the remainder of the screen. Headings should not be a dominant element on this type of screen.

Control arrangement. Control positioning and alignment on the screen should match that of the source document. Position controls in the same manner, or as close to the same manner as possible, to facilitate eye movements between the document and screen. (A well-designed document should have aligned elements, too. If not, still follow the form alignment.) Maintain a left-to-right entry arrangement, if the form is organized for completion in this direction. If, per chance, the document is organized top-to-bottom, then follow this top-to-bottom scheme.

Keying procedure. Use manual tabbing, not auto skip, to permit a rhythmic keying process. Keying will be faster and less error-prone.

Required and optional input. Distinguishing between required and optional data input is not necessary on these screens. This information will have been included within the source document design.

Instructions and completion aids. These will not be necessary. Instructions and completion aids will be located on the source document.

For more detailed information concerning document screen design, see Galitz (1992).

https://hemanthrajhemu.github.io

Display/Read-Only Screens

Display/read-only screens are used to display the results of an inquiry request or the contents of computer files. Their design objective is human ease in locating data or information. Thus, they should be developed to optimize human scanning. Scanning is made easier if eye movements are minimized, required eye movement direction is obvious, and a consistent viewing pattern is established. Next is a guideline summary.

- Organization:
  - Logical and clear.
  - Limit to what is necessary.
- Most frequently used information:
  - On earliest screens.
  - At the top of screens.
- Captions:
  - Meaningful.
  - Consistently positioned in relation to data fields.
  - Left- or right-aligned.
- Data fields:
  - Do not include a surrounding border or box.
- Spacing and grouping:
  - Create logical groupings.
  - Make them medium-sized, about 5 to 7 lines.
- Headings:
  - Uppercase or headline-style mixed case.
  - Set off from related controls.
- Data presentation:
  - Visually emphasize the data.
  - Display information in a directly usable format.
- Data arrangement:
  - Align into columns.
  - Organize for top-to-bottom scanning.
- Data justification:
  - For text and alphanumeric data, left-justify.
  - For numeric data, right-justify.
  - Create a data "ladder."
- Data display:
  - Consider not displaying no, or null, data.
  - Consider "data statements."

https://hemanthrajhemu.github.io

More detailed guidelines for screen organization, and data presentation, arrangement, justification, and display are included in the following discussion.

Organization

- Only display information necessary to perform actions, make decisions, or answer questions.
- Group information in a logical or orderly manner, with the most frequently requested information in the upper-left corner.
- For multiscreen functions, locate the most frequently requested information on the earliest screens.
- Do not pack the screen. Use spaces and lines to balance the screen perceptually.

Information contained on a display/read-only screen should consist of only what is relevant to the question for which an answer is sought. Forcing a person to wade through volumes of data is time consuming, costly, and error-prone. Unfortunately, relevance is most often situation-specific. An item that is relevant one time a screen is displayed may be irrelevant another time it is recalled.

Organization should be logical, orderly, and meaningful. When information is structured in a manner that is consistent with a person's organizational view of a topic, information is comprehended better and faster.

Finding information on a display/read-only screen can be speeded up by a number of factors. First, if information is never used, do not display it. Limit a screen to only what is necessary to perform actions, make decisions, or answer questions. Second, for multiple-window functions, locate the most frequently sought information on the earliest screens and the most frequently sought information on a screen in the upper-left corner. Never pack a display/read-only screen with information.

Captions. Provide meaningful captions clearly identifying the information displayed, unless the identity of data is obvious by its shape or structure (for example, an address). Use the headline style (all significant words capitalized), and consistently position all captions in relation to their associated data. Captions may be left- or right-aligned.

Data fields. Do not place a border around display/read-only information; inscribe the data so that it appears on the normal window background. It will be much more readable presented in this manner.

Spacing and grouping. Provide easily scanned and identifiable logical groupings of information. Create groupings of a medium size (five to seven lines) and set them off through liberal use of white space and conservative use of line borders.

Headings. Provide headings to identify groupings. Set off from their related controls using uppercase or mixed-case headline style.

https://hemanthrajhemu.github.io

Data Presentation

- Provide visual emphasis to the data.
- Display information in a directly usable format.
  - — Spell out any codes in full.
  - — Include natural splits or predefined breaks in displaying data.
  - — Provide multiple data formats when necessary.

~~338302245~~	~~072179~~	~~162152~~
338-30-2245	07/21/79	16:21:52

Figure 3.64

- Format common items consistently.
- For data strings of five or more numbers or alphanumeric characters with no natural breaks, display in groups of three or four characters with a blank between each group.

~~K349612094~~ K349 612 094

Figure 3.65

Data should be visually emphasized to attract attention. This will enable the viewer to quickly isolate the data and begin scanning the display for the needed information. A bolder presentation or bright color is recommended to accomplish this.

Present information in the most useful format possible, staying consistent with familiar standards and conventions. Never require the user to convert, transpose, compute, or translate data into other units. Also, never require users to refer to documentation to determine the meaning of an item of data. Allow the user to choose the format that data with varying formats is displayed in (for example, dates, time). Fully spell out any codes and include natural splits or predefined breaks in displaying common pieces of data such as telephone numbers and dates.

Common items such as telephone numbers and dates should be formatted in a consistent manner on all screens.

A data display should also reinforce the human tendency to break things into groups. People handle information more easily when it is presented in chunks. Display data strings of five or more alphanumeric characters with no natural breaks in groups of three or four, with a blank space between each group.

Data Arrangement

- Align data into columns.
- Organize for top-to-bottom scanning.

https://hemanthrajhemu.github.io

To aid scanning, align data into columns with a top-to-bottom, left-to-right orientation. This means permitting the eye to move down a column from top to bottom, then move to another column located to the right, and again move from top to bottom. This also means, if the situation warrants it, permitting the eye to move easily left to right across the top of columns to the proper column, before beginning the vertical scanning movement.

Top-to-bottom scanning will minimize eye movements through the screen and enable human perceptual powers to be utilized to their fullest. Display/read-only screens are often visually scanned not through the captions but through the data fields themselves. A search for a customer name in a display of information frequently involves looking for a combination of characters that resembles the picture of a name that we have stored in our memory. The search task is to find a long string of alphabetic characters with one or two gaps (first name, middle initial, last name, perhaps). A date search might have the viewer seeking a numeric code broken up by slashes. Other kinds of information also have recognizable patterns and shapes. Control captions usually play a minor role in the process, being necessary only to differentiate similar-looking data fields, or for new screen users.

Vertical scanning has led to two key requirements in the design of display/read-only screens: call attention to data fields, and make the structural differences between data fields as obvious as possible. Differences are most noticeable in a columnar field structure, since it is easier to compare data when one piece is above the other.

Data Justification

- Left-justify text and alphanumeric formats.

Name:	~~Bill Watters~~	Name:	Bill Watters
Street:	~~612 Hidden Valley~~	Street:	612 Hidden Valley

Figure 3.66

- Right-justify lists of numeric data.

Charge:	~~645,194.88~~	Charge:	645,194.88
Federal Tax:	~~19,235.16~~	Federal Tax:	19,235.16
State Tax:	~~5,204.03~~	State Tax:	5,204.03
Local Tax:	~~1.24~~	Local Tax:	1.24
Total Cost:	~~669,635.31~~	Total Cost:	669,635.31

Figure 3.67

- Create a data "ladder."

Tree:	~~Pine~~	Tree:	Pine
Age:	~~14~~	Age:	14
Number:	~~422,598~~	Number:	422,598
Class:	~~C~~	Class:	C
Location:	~~NW~~	Location:	NW

Figure 3.68

https://hemanthrajhemu.github.io

In general, columnized text and alphanumeric data should be left-justified, and numeric data should be right-justified. In aligning data fields, keep in mind how the pieces of data will look in relation to one another when they contain typical information. The visual scan should flow relatively straight from top to bottom. This may require that some data fields be right-justified in the column that is created, not left-justified, or vice versa. The objective is to create what looks like a ladder of data down the screen.

Data Display

- Consider not displaying data whose values are none, zero, or blank.

Elephants:	612		Elephants:	612
Lions:	123		Lions:	123
~~Hippos:~~	~~0~~		Giraffes:	361
Giraffes:	361			
~~Kudus:~~	~~0~~			

Figure 3.69

- Consider creating "data statements," in which the caption and data are combined.

Elephants:	612	612	Elephants
Lions:	123	123	Lions
Giraffes:	361	361	Giraffes

Figure 3.70

Consider not displaying fields containing no data. When displayed on a display/read-only screen, some data fields may be blank or contain a value such as zero or none. In many situations it may not be important to the screen viewer to know that the field contains no data. In these cases consider not displaying these screen elements at all. Present on the screen only the fields containing data, thereby creating less cluttered screens.

If this alternative is chosen, space on the screen must be left for situations in which all fields contain data. To avoid large blank screen areas, a useful rule of thumb is to allow enough space to display clearly all data for about 90 percent of all possible screens. For situations in which screens must contain more data than this, going to an additional screen will be necessary.

This nondisplay alternative should only be considered if it is not important that the viewer know something is "not there." If it is important that the viewer know that the values in a field are zero or none, or that the field is blank, then the fields must be displayed on the screen.

You may also want to consider displaying data statements. The traditional way to display data on an inquiry screen is the "caption: data" format, for example, "Autos: 61." Another alternative is to create data statements where the caption and data are combined: "61 Autos." This format improves screen readability and slightly reduces a screen's density. If this data statement format is followed, consider the statement as data and highlight it entirely.

https://hemanthrajhemu.github.io

The Web — Web sites and Web Pages

The Web has an almost unlimited supply of information — for those who can find it. The dilemma for the user is how to navigate within the Web, deal with the overwhelming amount of information presented, and finally locate the elusive answer. The magnitude and structure of the Web seems to be creating a user-interaction pattern with these characteristics:

- The most sought-after Web commodity is information content.
- Behavior is often goal-driven.
- Reading is no longer a linear activity.
- People scan and select.
- Impatience.
- Frequent switching of purpose.

Web users access a site for different reasons: a focused search for a piece of information or an answer, a less focused browsing, or simply to surf. But information content is what most people come to see. High-tech capabilities, fancy graphics, and a rainbow of colors do not compensate for insufficient or poor content. All innovations are judged by how well they support the presented information. Users are also strongly goal-driven, often looking only for the one thing they have in mind. In foraging through the Web, reading is no longer a linear activity; instead, information is acquired in scattered bits and pieces.

Easy information scanning is very important. People seldom read more than a few words as they seek items of interest. A study found that 79 percent of users skimmed and scanned Web pages. Because people are goal-oriented, they want to get there as fast as possible. Having to read in depth is an obstacle to reaching the goal quickly. This is not unlike users of paper documents. Schriver (1998) found that 81 percent of workplace documents are used in a similar way.

So, the user is impatient, with little time to waste. Things like a slow download, pages that are not easily scannable, and confusing navigation, will quickly drive people away from a Web site. The user also frequently switches purpose and direction, browsing one moment, searching the next, now in one site, then in another, then back again, and on and on. Design must easily accommodate people who change directions, and leave a site and return frequently to it.

Dimensions of a Web Site

Web sites generally vary in three dimensions: structure, navigation, and information content. Important s*tructure* issues include how the information on a Web site is organized, the number of pages it contains, and the length of its pages. *Navigation* issues revolve around how the user moves around the site, including the methods used and the support the site provides to aid navigation. *Information content* issues are what information is included, how much information is included, and how the information is presented.

https://hemanthrajhemu.github.io

Web User Interaction Styles

In recent years the specific activities of Web users have come under closer scrutiny. Sellen et al. (2002), in a task analysis of knowledge users, found three different Web interaction approaches were demonstrated: browsing, information finding, and information gathering. Each approach had a different flow and activity behavior characteristics.

Browsing is non-specific surfing. People move through a Web site at their own pace and according to their own interests. Browsing is analogous to shopping. A person walks into a store (Web site), looks around (the page), gets a feel for the place (presentation style, layout, and so on), looks for clues or signs of interest (headings, summaries, and so on), wanders at whim (follows a link), and then decides to stay and linger for a while, or leave. The person may leave empty-handed, or have picked up various products (or scattered bits and pieces of information) during these wanderings. Upon leaving, the person's destination may be another store (Web site) in the mall from which he or she may soon return after doing some "comparison shopping." In the shopping bag may now be other products (or scattered bits and pieces of information) that have been picked up. This sort of "interlaced" browsing behavior must be supported in Web site design.

Information finding involves seeking specific answers to specific questions or needs. Nielsen (2006b) provides evidence that no single Web site incorporates the total user experience. In one study (finding information for a specific product) Nielsen reports that one user, over about 45 minutes, visited fifteen sites and viewed ninety-three pages, an average of 29 seconds per page. He concludes that studies have validated that the Web site user experience involves: 1) flittering between sites gathering snippets of information from multiple competing sites to build a thorough understanding of a specific problem; and 2) interleaving visits to different site genres, alternating between vendors, retailers, and reviews using search engines as a hub.

Information gathering involves doing reconnaissance for future activities to be performed, or information to be looked for at a later time.

Table 3.3: Reported User Web Activities

BROWSING ACTIVITIES (BYRNE ET AL., 1999)	APPROXIMATE PERCENTAGES
Using information, including reading, printing, and downloading	58
Visually searching for information on pages	12
Providing requested information	18
Waiting or scrolling	12
PRIMARY REASONS FOR IMPORTANT USES OF THE WEB (MORRISON ET AL., 2001)	
Evaluate multiple products or answers to make a decision (Comparing or choosing)	51
Get a fact or document, find out something, or download something (Finding)	25
Gain understanding of some topic, including locating it (Understanding)	24

(continued)

https://hemanthrajhemu.github.io

Table 3.3 *(continued)*

METHODS USED TO ARRIVE AT INFORMATION FOR IMPORTANT WEB USES (MORRISON ET AL., 2001)	
Searching for multiple pieces of information, not looking for one	71
Particular answer (Collecting)	
Searching for something specific (Finding)	25
Looking around or browsing without a particular goal (Exploring)	2
Visiting the same Web site looking for updated information (Monitoring)	2

Byrne et al. (1999) looked at what people do when browsing, and Morrison et al., (2001) analyzed the methods people reported they used to arrive at information they needed for important tasks, and their reasons for the use of the Web. The results are summarized in Table 3.3.

In the Byrne et al. study, a significant amount of time was spent waiting and scrolling. This percentage was actually greater than reported above because waiting time between successive page presentations was not reported. In the Morrison et al. study, 96 percent of the activities were goal-driven; that is, users were addressing a specific need. It is also interesting to note that more than two-thirds of the time people were looking for multiple pieces of information, not single pieces.

Cothey (2002) found that experience modifies a Web user's behavior. More experienced users

- Access the Web less often and more sporadically.
- Access fewer sites.
- Tend to browse to sites (either directly or via other trusted Web sites) rather than getting there via search.

Fogg et al. (2002) found that users who do not have in-depth knowledge of the content domain evaluate a site's credibility based upon its design look as well as information design, structure, and focus.

The Web Experience

A person's Web experience is shaped by many things, including how well the Web site design reflects his or her expectations, capabilities and limitations, the Web site's visual appearance and content, and the Web site's functionality. The following general factors are important in creating for the user an efficient, effective and pleasant experience. Some of these factors have been discussed in great detail in earlier chapters; others will be more fully discussed in pages or chapters that follow. The discussion below is partially based upon principles described by Nielsen (2000), Koyani et al. (2004), and Stone et al. (2005).

https://hemanthrajhemu.github.io

Expectations

- Meet user expectations concerning content, organization, and navigation.

People will probably have expectations based upon prior knowledge or past experience. Use of familiar formatting and navigation schemes makes it easier for a person to learn a site's organization and structure, and to remember how to use it. Some users will not use a Web site often enough to learn to use it efficiently. Using familiar concepts will improve ease of use.

Tasks

- Assure that design reflects the user's capabilities and limitations.
- Standardize task sequences.
- Provide ease of use.

A Web sites design must reflect the capabilities and limitations of its users, as described in Step 1. Factors include the proper allocation of functions to people and computers to take advantage of the respective strengths of each, and designing for human memory limitations. Standardizing task sequences allows people to do things in the same manner and order in similar situations, thereby speeding learning. People also want to do what they need to do easily and quickly.

Visual Appearance

- Provide a visual appearance that is
  — Attractive
  — Consistent

A Web page must be attractive and visually pleasing. The human perceptual mechanism seeks order and meaning, trying to impose structure when confronted with uncertainty. Whether a page has meaningful and evident form or is cluttered and unclear is immediately discerned. A Web site must also be consistent in appearance, using related visual elements throughout. A consistent visual style provides a sense of unity and helps people maintain orientation. Two studies have found that visually inconsistent displays have higher error rates than visually consistent displays. Inconsistency in element size, however, apparently does not negatively impact performance or preferences. A study found that different-sized controls or widgets had no impact on either performance or preference (Koyani et al., 2004). To create attractive and consistent sites, follow the text, layout, color, and graphics guidelines scattered throughout this text.

https://hemanthrajhemu.github.io

Information Content

- Assure that the content meets all the user's needs.
- Assure the most important information is
  — Positioned prominently.
  — Visible without scrolling.
- Assure that the content is updated frequently.

The site must provide information or functionality that people desire. Information content must be engaging, relevant, and appropriate to the user. One study found that the most critical element of a Web site is content; other studies have found that content is even more important than visual design, functionality and navigation (Koyani et al., 2004). Without content a Web site has no real purpose.

Important information should be positioned prominently on a page. In newspaper parlance, it should be located above the fold, the area on a page that is always visible when a page is displayed. The user may miss important information if scrolling is required to see it.

A site must be regularly updated commensurate with the volatility of the content. A news Web site may be updated every few minutes; an online encyclopedia may remained unchanged for months or longer. A product Web site may change only when new products are introduced or prices modified. Regular updates will increase its value and provide an incentive to return. Without timely updates, a site will fail to achieve its objective. Web sites subject to frequent updates should always have the update date displayed prominently on the homepage.

Web Page Headings

- Control headings:
  — For groupings of controls on Web pages, follow the control heading guidelines.
- Content and information headings:
  — Usage:
    - Create headings to classify and organize information.
    - Provide meaningful headings and subheadings that clearly describe the content and nature of the information that follows.
    - Use liberally throughout the Web site.
    - Use as introductory links.

  — Presentation:
    - Establish a hierarchy of font styles, sizes, and weights dependent upon the organization created and the importance of the text content.
    - Settle on as few sizes and styles as necessary to communicate page content and organization to the user.
    - Do not randomly mix heading levels or skip heading levels.

Control. Headings for controls located on Web pages will generally follow the guidelines for controls previously described.

https://hemanthrajhemu.github.io

Content. Well-labeled Web page headings will provide a roadmap aiding understanding a site's organization and navigation structure. They also break up the text and facilitate page skimming and scanning. Poorly constructed and similar looking or worded headings are a common Web site problem, making it difficult for people to find what they are looking for. Unclear headings often cause people to start reading to determine content relevance, unnecessarily slowing them down.

Web page headings should be used to classify and organize information into related or common functions, and to provide cues concerning page structure. Headings should be meaningful, clearly describing the content that follows. Liberal use of headings is also recommended to aid user scanning.

Headings should also be used as introductory links. This gives people a summary or overview of the content and makes it easy to jump to the right place to find the information that they need. The guidelines for creating meaningful heading text will be addressed in Step 8.

In general, the presentation of page and text headings on Web pages will follow the practices used for paper documents. A hierarchy of fonts styles, sizes, and weights will be developed based upon the Web page's organization and the contents' importance. The guidelines presented earlier in the "Typography" section can be used for this purpose. To avoid clutter, only use as few sizes and styles as necessary to communicate page content and organization to the user. Do not randomly mix or skip heading levels. If overdone, the page's organization will be more difficult to ascertain, and user confusion may result.

Platforms

- Design for different platforms and screens.
- Specify an image-safe area.

In a Graphical User Interface it is possible to control the exact appearance of a screen. For Web sites this is not possible, because pages may be viewed on different size screens and/or with different size fonts. The result is that for some users preventing scrolling may be impossible. The best solution, according to Stone et al. (2005) is to anticipate the typical screen and resolution and specify an image-safe area of the screen that will always be visible to most viewers. The most important information can be located here. This is not an ideal solution, since users with very small screens will still have to scroll, and those with large screens will be viewing the page through a window that is smaller than the actual screen size.

Processes

- Do not display unsolicited windows or graphics.
- Provide printing options.
- Warn of time-outs.

https://hemanthrajhemu.github.io

Unwanted pop-ups on screens are annoying and distracting, especially when a person is focusing on doing something.

Many people prefer to read from a paper document, not a screen. A paper is more convenient, can be annotated, and can be perused almost anywhere. Also, people may not always trust a Web site's persistence, being unsure if the information can again be found or whether it may still be available later. So, always provide a printing option on a Web page.

Usually because of security reasons, a page may be designed to "time out" automatically. A time out may be frustrating to a user who is either reading or making entries slowly. Inform a user if a page will time out and permit, and act on, a request for additional time. It can be very exasperating to lose information already entered.

Downloading

- Minimize downloading time.
  - — Ensure a page is built up on the screen so that header and navigation information appear first.
  - — Only include videos, animation, sound, and video if they are needed.
  - — Make downloading of additional media, such as sound and video, optional.
  - — If a page is going to download slowly, advise the user and indicate file size.

Slow-to-load pages can also be aggravating, especially to people using slower dial-up connections or connecting from mobile devices. Slow-to-download designs with large and unnecessary graphics or animations may discourage people from using a site at all. Minimizing the number of bytes per page is the best way to facilitate fast page loading. If a page will probably download slowly, warn the user and provide the file size.

Web Applications

- Provide Web applications as necessary.

In general, as described earlier, a Web application lets you do and save something. A Web page mostly provides information. The Web is a type of Graphical User Interface. Consequently, it uses the same collection of screen controls as GUI systems do. In building a Web application, all the principles for selecting, presenting and organizing GUI controls should be followed.

Accessibility

- Design for accessibility.

A Web page or application can be useful only if it can be accessed. This is commonly refereed to as *accessibility*. The broad definition of accessibility states that all users must be able to easily access all aspects of the Web. The guidelines presented in this book are dedicated to that purpose. The narrower definition of accessibility requires that people

https://hemanthrajhemu.github.io

with disabilities also be able to access all aspects of the Web. Accessibility issues and guidelines for people with disabilities are discussed in Step 10.

Assistance

- Provide assistance to users.

People using a Web site will occasionally need assistance. This is especially true for inexperienced or first-time users. Koyani et al. (2004) report that one Web site found that 36 per cent of the people visiting it were first-time users. They suggest that what should be included on a Web site with many inexperienced users is a link that allows new users to access more information about the content of the site and how to navigate around it. This will balance the needs of both experienced and inexperienced users. The link concept is fully described in Step 4.

The Web Structure

Historically, people have mostly interacted with documents, books, GUI applications, etc., in a linear way. A book, for example, is generally read from beginning to end, a GUI application completed from start to finish. The Web, however, is a non-linear entity of a huge number of screens and/or pages that are linked together in a very complex way. Systems of this kind are referred to as *hypertext* systems. The result is that a Web site, or the Web, may be navigated in a seemingly endless number of ways.

The Web, then, is an extremely flexible entity. But flexibility often leads to confusion for the user, especially for sites possessing dozens, or hundreds, of pages. The complexity makes it difficult, or impossible, for the user to develop a mental model of the site structure. Getting lost within the pages is easy, and disorientation often occurs. To address these problems begins with developing as clear and meaningful a site structure as possible. A Web site must

- Organize content naturally for the audience.
  - — Reflect the natural organization of the information to be presented.
  - — Reflect the structure of the tasks people want to carry out.

Natural Organization. Certain organizations and entities, through their evolution, have created a standard structure for presenting their products, services, or themselves. A library, for example, has its books arranged within well-established and well-known groupings; a store organizes its shelves in an established pattern. Structuring a Web site following naturally evolved schemes will simplify the users' task in understanding, and navigating, the site. Ma and Salvendy (2003) have found that well-organized information hierarchies can be as effective and satisfying for users as search engines.

People's Tasks. A site's structure should always support what the users may want to do at the site. This will be determined when site requirements are defined. Put first what people need most. Unfortunately, it is difficult to anticipate every user requirement. So, navigating a site should be as flexible as possible.

https://hemanthrajhemu.github.io

Page Perusal

Focusing on the page's content, some studies show that the user's eyes are first drawn to the page's text, particularly headings, captions, summaries, and notes. Individual words and phrases are read for meaning and relevance. At this point the page's graphics are generally ignored. Studies find that the most frequent method used in perusing a page is scanning or skimming, concentrating less on detail and word-for-word reading. Design features that enhance scanning are critically important in page design. As the scanning continues, the user's attention becomes progressively more detailed; page components begin to take on meaning. Shallow reading is combined with selective depth. Rarely are articles read fully. If a page does not appear to be relevant to a user's needs or goals quickly, it will be removed.

Scanning Guidelines

- Organize content in a logical, systematic and obvious way.
- Highlight and emphasize important information.
- Use clear, well-located headings and subheadings.
- Use short phrases and sentences.
- Use small readable paragraphs.
- Include bulleted and numbered lists.
- Array information in tables.

To aid in finding information, a Web page must be structured to facilitate scanning. Studies report that about 80 percent of viewers scan a new page when it first appears. When they find something interesting, then they read. Only 16 percent read word-by-word. People also spend about 12 percent of their time trying to find desired information. People that scan generally read only headings, not full text prose. Useful information buried in dense text is often missed (Koyani et al., 2004). Difficult-to-scan pages may cause a viewer to give up and move on, preferring to spend what is often limited time elsewhere.

Organization. Pages should be organized to guide a person's eye through the page in a systematic way. Efficient eye movement, and scanning direction, can be made obvious through the alignment and columnization of page elements, and effective use of white space. Order information page information to be consistent with the viewers expected scanning path.

Highlight. Emphasize important information, issues, and topics through use of varying font sizes and boldness. Links can serve as emphasis entities since the underlining makes them stand out.

Headings and subheadings. Distinctive and obvious headings are often targets for page scanning. Well-written headings can be important cues in helping people classify and organize page information, understand page structure, and maintain orientation. Headings should be written clearly, conceptually relating to the information or functions that follow. Scanning is particularly important for older people because they stop scanning and start reading more quickly. Poorly written

https://hemanthrajhemu.github.io

headings may cause them to start reading in places that do not provide the information being sought (Koyani et al., 2004). The same reading reaction, of course, may result from poorly written headings for users of all ages.

Phrases, sentences, and paragraphs. Phrases, sentences, and paragraphs must also be concise and well-written for easy scanning.

Bulleted lists and tables. Lists draw attention to important points that may be entirely missed if buried within a paragraph. Lists also lend themselves to linking. One study reported that incorporating lists accounted for 47 percent of the improvement in a Web site. Also, present information in tables for easier scanning and comparing.

Writing Guidelines

- Provide a meaningful title.
- Provide meaningful headings and subheadings.
- Concisely write the text.
- Keep the language simple and avoid jargon.
- Write short paragraphs containing only one idea.
- Use the inverted pyramid style of writing.
- Provide concise summaries.

Title and headings. Provide a meaningful page title. The purpose and message of a page should be identifiable from its title. There is also evidence that poor titles make identifying page content much more difficult when using the browser's Back and History buttons (Cockburn et al., 2002). Provide meaningful headings and subheadings. Most users read headings; not all read further. Headings help users determine whether to continue or ignore large chunks of information. Headings should be simple, not clever or cute. Subheadings break up larger blocks of text, making it easier to find the portions of interest. Typeface variations should be used to establish a heading hierarchy.

Conciseness. Users prefer written text, says research, because words reduce one's cognitive load, resulting in faster and more efficient information processing. To aid scanning, Web page text should be very succinct, containing fewer words than conventional writing. Half the word count or less is recommended. Create narrow text columns that minimize the eye's horizontal scan. Write short paragraphs, restricting each to only one idea. People tend to skip additional important points in a paragraph as they skim, or if they are not caught by the first few words in the paragraph.

Simple language. Use simple language that the user understands. Avoid jargon, words that the user may not understand.

Inverted pyramid. Use the inverted pyramid style of writing. This style starts with the conclusion and follows with expanded detail, either within the paragraph or at the end of a link. The conclusion first allows the user to quickly determine whether it is worthwhile to continue reading.

https://hemanthrajhemu.github.io

Summaries. Provide concise content summaries to give the user a snapshot of the text content before reading. Link to full-length treatments where relevant and useful information can be presented.

Writing text is discussed in more detail in Step 8.

Presentation Guidelines

- Highlight:
  - — Key information-carrying words or phrases.
  - — Important concepts.
- Call attention to new or changed content.
- Provide comparison tools.

Highlighting. Important information-carrying words, phrases, or concepts should be highlighted. Keywords, words that differentiate a page from others, or words that symbolize what a page is all about are candidates for highlighting. About three times as many words as writing for print can usually be effectively highlighted on a Web page. Possible highlighting techniques include boldness, increased brightness, or a different color. The standard link designation, an underline, is considered a highlighting technique. Do not highlight entire sentences or long phrases. The eye picks up only two to three words at a time, and the screen may become too busy.

New or changed content. Call attention to new or changed content through highlighting or a *What's New* heading. Make it easy for people to see what has been added and when it was added. Fogg et al. (2001) found that updating a Web site frequently makes the site more credible.

Comparison Tools. People can make faster and better decisions when they can compare items together. Decision-making can be aided through comparison tools, a facility that presents selected items together on one screen. When a site offers too many choices, not the right information, and no comparison mechanism, decisions are not as good (Browne and Pitts, 2004; Jedetski et al. 2002).

Web Page Length

- Minimize page length.
  - — Generally, use shorter pages for
    - Homepages.
    - Navigation pages.
    - Pages needing to be quickly browsed or read online and content quickly found.
    - Situations where a page will be loading over slow modems and all pages are not needed.
  - — Generally, use longer pages for
    - Content pages where uninterrupted reading is desirable.
    - Content pages in which an entire concept must be understood without interruption:

https://hemanthrajhemu.github.io

- To match the structure of a paper counterpart.
- To make pages more convenient to download.
- To make pages more convenient to print.
- To simplify page maintenance.

— Generally, restrict to two or three screens of information.

— Place critical or important information at the very top so it is always viewable when the page is opened.

- Locate within the top 4 inches of page.

The answer to the question "What should be the maximum length of a Web page?" has no precise answer. Maximum length recommendations in the literature vary, running from "short" to "1 to 2" to "3 to 5" screens, to even more. One serious problem in establishing an exact page size is that the multitudes of Web viewers use a variety of different browsers and monitors. Another problem is that font types and sizes can vary, impacting how much of a page is visible at any one time. What is viewable on one person's screen, then, may not be entirely visible on another. Another problem is that the viewable depth of a Web screen is somewhat less than the typical business information screen. Very few Web users have screens large enough that enable them to view an entire letter-size page all at once. A business information system, conversely, usually contains standardized applications, consistent-size screens, and standard monitors.

Arguments *for shorter pages* are that longer pages:

- Tax the user's memory, as related information is more scattered and not always visible.
- Can lead to a lost sense of context as navigation buttons and major links disappear from view.
- Display more content and a broader range of navigation links making it more difficult for users to find and then decide upon what path to follow.
- Require excessive page scrolling, which may become cumbersome and inefficient.
- Are less conducive to the "chunking" information organization scheme commonly employed in Web sites.

Arguments *for longer pages* are that they:

- Resemble the familiar structure of paper documents.
- Facilitate uninterrupted reading, especially on content pages.
- Require less "clicks" for navigating through a Web site.
- Are easier to download and print for later reading.
- Are easier to maintain because they possess fewer category navigation links to other pages.

Ultimately, page length decisions must be made depending upon the use and goals of the Web site and its pages. In Web page design, what is generally agreed upon is the following:

Minimize page length. Given the scarcity of Web page screen real estate, and its screen size variability, it is not practical to confine a page to the least common

https://hemanthrajhemu.github.io

denominator display screen size. To fulfill the "Amount of Information" guidelines just described will often require a page length to exceed a "screenfull." Determining an optimum page length will require balancing these factors.

Generally, pages should be long enough to adequately convey the necessary information, but not so long that excessive scrolling becomes a problem. Candidates for short pages are homepages where a site's identity and goals must be quickly identified, and navigation pages where all navigation alternatives must be clearly displayed and quickly found. Pages needing to be quickly browsed or read online, and content quickly found should also be short. If pages download slowly and all pages are not always needed, also provide a contents page containing many navigation links to short pages.

Longer pages work best when an entire concept must be understood without interruption. In these cases, present the entire concept in one page with internal links to subtopics. The size of a page may also be governed by the size of a paper or document counterpart. Longer documents will require longer pages. Convenience in downloading or maintaining may also require longer pages. Fewer Web files are easier to maintain. To print all or most of the page content to read offline, use one long page or prepare a version that uses one page.

Restrict a page to two or three screens. This is a very general recommendation that will ultimately be governed by the factors just discussed.

Critical or important information at the very top. Critical or important information should be placed where it will be immediately visible when the page is displayed. In Web page design, this is referred to as "above the fold," a term borrowed from newspaper page layout. Above the fold is about the top 4 inches of a page. "Below the fold" will usually require scrolling to access and may not always be seen.

Web page size will also vary depending upon the user's task and motivation, the page's purpose, and the Web page provider's objectives. If the user is interested and motivated, a longer page may be warranted, reducing the user's effort in reloading pages. If the user is being "sold" a product, advertising practices should be employed, including keeping pages short. If the page's purpose is strictly navigation, then one screen is most practical so that all navigation choices are visible. A factor to be determined in establishing page length, then, is the user's reading "rhythm" (Levine, 1995). A number of the above recommendations are based Koyani et al. (2004).

Fluid versus Fixed Page Layouts

Layouts of a Web page may either be *fluid* or *fixed* (sometimes referred to as *frozen*). A fluid layout is the most traditional, the contents of a Web page filling the entire window, expanding and contracting with the size of the window. The fluid layout is simplest to implement. Common fixed layouts are *centered* or *left-justified*. The centered layout is centered within the boundaries of a display, requiring users to only focus on a narrower layout in the middle of the display no matter what screen resolution is being used. The left-justified layout is positioned against the left border of the display and is usually slightly wider than the centered layout.

https://hemanthrajhemu.github.io

Both types of fixed layouts reduce visual scanning requirements horizontally but leave a considerable amount of white space on the screen and may require more vertical scrolling. Fixed layouts are difficult to use on big monitors if windows are not resized. People with high-resolution monitors who enlarge their font size cannot also enlarge the window size to make the text more readable. On small monitors, fixed layouts may require the undesirable horizontal scrolling. The right-most part of a page may also be cut off when printing a fixed-width page.

Bernard and Larson (2001c) compared these three layout styles and found no differences in search time, accuracy in finding information, or in search efficiency. However, study participants felt that the fluid style was best suited to reading and finding information. It was also the preferred layout style.

Homepage

A site's homepage is probably a Web site's most important page. The homepage gives people a first impression of a site, and first impressions can create a positive or negative feeling, or a feeling somewhere in between. A negative first impression can lead to rejection, a positive first impression create an urge for further exploration. One study has reported that when people are asked to find high quality sites, about one-half the time they only look at the homepage (Koyani et al., 2004).

A homepage must be easily identified as a homepage. People now expect to find certain elements displayed on, and actions available from, the homepage. What follows is a brief summary of important homepage characteristics. A more detailed discussion can be found in Step 13.

Homepage Purpose

- To tell the viewer what the site does.
- To tell the viewer where they are.

In many cases a Web site's purpose is easily inferable. In other cases it may not be obvious. To avoid misunderstandings, assume that potential users know nothing about the site and what is presented within it. Explicitly describe a site's purpose using brief text or a tagline, a phrase or short sentence located just below the Web page owner's logo or name. Also describe who the site is intended for or of interest to. Describe the kinds of information available. The homepage must convince users to stay by telling them what they will find within the site. People usually will not read lengthy text, however, to determine a site's purpose. Annoying Flash graphics may also chase people away.

A homepage is also a concrete and visual anchor point; a safe haven to return to when one is confused or decides to do something else.

Content

- Show all major options.
- Limit prose text.

https://hemanthrajhemu.github.io

All of a Web site's major and important options should be presented on the homepage. People should never be required to go to a second or third level to uncover important content or links. Minimizing prose text will make the page much easier to scan.

Size

- Limit to one screen.

Limit the homepage to one screen whenever possible. All important elements should be viewable without scrolling so they will not be missed. Scrolling is time-consuming, people may not be aware that scrolling is required, or people may simply conclude that what they can see is not important so what is hidden must not be important, too. Elements of less importance and those not vital to site use may be included below the visible area, however.

Elements to Include

- Name or logo of Web site owner.
- Web site name.
- Brief description of Web site.
- Summary of the key informational content.
- Site overview or map.
  — Navigation links to most (if not all) of the site or major sections.
- Summary of the latest news or promotions.
- Search facility.
- Announcements of changes to Web site.

Most Web sites include most, or all, of the following elements.

Owner name or logo. Include the name and/or logo of the organization that owns the site. This is usually placed in the upper-left-hand corner. If necessary, include a descriptive tagline.

Web site name. Provide a Web site name. It is usually located at the top of the page. It should also be in the browser window title bar.

Brief description. A short introduction to the site.

Summary of the key content. A summarization of a Web site's major and important options. This is most valuable for new users.

Site overview or map. To aid understanding of site navigation.

Navigation links. Links to most (if not all) of the site's major sections.

Summary of the latest news or promotions. Call attention to content that has been recently changed or updated. This is most valuable for repeat visitors.

Search facility. Search is an important part of any large Web site.

https://hemanthrajhemu.github.io

Announce changes. Communicate any planned major Web site changes ahead of time. Do not surprise people. They may not know what to do on the changed site. Tell users what has changed and when, also assuring them that previously available information is still available. Also advise people of important changes at other locations in the site (Koyani et al., 2004).

Organization and Layout

- Place critical or important information at the very top so it is always viewable when the page is opened.
  — Locate it within the top 4 inches of the page.
- Position remaining elements according to importance.
- Reduce graphic complexity and textual density toward the page bottom.

Placing critical or important information at the page top where people first look will make it easier to find. All important navigation options should also be at the top. The general principle is to position these elements above the "fold," a term carried over from newspaper parlance, meaning that information that can be read on a folded paper displayed on a rack.

Remaining elements can then be positioned according to importance. Graphic complexity and textual density can be reduced toward the bottom of the page.

Access

- Enable access to the homepage from any other page.

A homepage is a concrete and visual anchor point: a place go back to when a person tries to recover from a problem, is confused, or desires to start something new. Consequently, create an easy and obvious way to quickly return to the homepage from any page in the Web site. To do this, provide a link labeled *Home* near the top of every Web page.

MAXIM **When the visual design clarifies the functional intent, the interface becomes intuitive.**

Browsing and Searching

A person, in looking for a particular item of information on the Web, can implement "find" strategies commonly called *browse* and *search*. As mentioned previously, browsing is non-specific surfing. People wander around a Web site, or Web sites, at their own pace, following discovered links, scanning headings, and using other presented page cues to try and locate what they are interested in. Search is a much more structured find mechanism. In search, one or more keywords are entered into a search field, sometimes with restrictive parameters, and then the user is presented with a collection of descriptions and links that might possibly contain the information of interest.

https://hemanthrajhemu.github.io

Essentially, browsing is an exercise of a person's powers of recognition. As page items are scanned, memories are triggered causing relevant, or possibly relevant, items to be pursued and explored. Search, on the other hand, is partially an exercise in recall. To start the search requires that keywords be dug from memory and entered into the search field. The result of the search will reflect the relevance of the keyword to what is actually being sought.

The task of browsing versus searching has been the subject of several recent research studies. The previously described Ma and Salvendy (2003) study, for example, found that well-organized and clearly presented information hierarchies can be as effective and satisfying for users as search engines. Katz and Byrne (2004) found that on sites with clear labels and a large number of navigation options, people tend to browse rather than search. Search was utilized less than ten percent of the time. Searching was no faster than browsing in this context. Presenting many navigation options with a good site organization and well-written labels is important.

One form of browsing involves following links. Spool et al. (2001b) reports that Web site *browsers* following links, compared to *searchers*, people utilizing a search facility, exhibited these differences:

- About sixty percent of browsers continued to browse a site, while only twenty percent of searchers did so.
- Browsers were three times more likely to find related (valuable) content.
- Browsers who started with links on the homepage looked at almost ten times more content pages.

Browsing resulted in a person staying longer, digging deeper, and finding more related content within a site. If this is desirable, browsing should be encouraged. Searching is still a valuable tool in Web site navigation and should never be discounted, however.

Bailey (2003a) surveyed the literature concerning browsing using links and searching and additionally reported the following:

- Linking tends to be more effective than searching to find content (Ojakaar and Spool, 2001).
- Linking through to a target may require fewer clicks than searching (Spool, 2002).
- When looking for something specific or unique, users prefer to use the search function (Spool, 2002).
- Preference for using links versus searching appears to depend on the Web site and the task being performed (Spool and Ojakaar, 2001).

Teevan et al. (2004) explored how and when people decide to search versus browse. The researchers were surprised to find that the participants, all with advanced computer experience, used key-word searches in only 39 percent of their searches. Based upon this research, these researchers suggest that people exhibit two search strategies: teleporting and orienteering.

Teleporting occurs when a person jumps directly to the sought information, such as through a search engine. *Orienteering* consists of narrowing the search space through a

https://hemanthrajhemu.github.io

series of steps, such as through following links. Based upon a person's past and current contextual information, the target is eventually honed-in upon. Most often the participants took one initial large step to get in the vicinity of the information source, then took a series of smaller steps, refining their search as they proceeded toward their destination. The researchers observed that

- Study participants choose not to teleport, even when it appeared viable.
- Keyword searching tended not to be used.
- Occasionally, when keyword search was used, it was a tactic with orienteering.

Teeven et al. propose that orienteering

- Cognitively is less demanding, allowing people to rely upon experience rather than requiring discrete articulation of the searched-for item at search initiation. Consequently, the search space is reduced.
- Provides a person a greater sense of control and location.
- Permits small, incremental steps providing additional context for interpreting the results.

Straub and Valdes (in Straub, 2005) describe the results of a pilot study comparing search and browse. In locating a specific piece of information, participants had a higher success rate with search (75 percent) than browse (69 percent). For participants where browse failed to find the information, a follow-up search had an 88 percent success rate. Despite the fact that search yielded better results than browse, study participants began a follow-up or additional information location task with browse 91 percent of the time.

In conclusion, while search seems to aid locating specific and unique items, browsing works better for items that are not specific and unique, and people seem to prefer to browse. Browsing requires less effort, provides more and better context, locates desired content faster, and allows viewing of substantially more content. Navigation links and a search function should be included on all key Web pages.

Browsing Guidelines

- Facilitate scanning.
- Provide multiple layers of structure.
  — Make navigation easy.
  — Freely use links.
- Respect the user's desire to leave.
  — Upon returning, help the users reorient themselves.

Scanning. The scanning guidelines recently presented also facilitate browsing behavior. The organization of information and the writing and presentation of content are critical to the process.

Structure. Provide multiple layers of structure with high-level summaries so users can decide if they want to browse deeper or simply move on. Freely use links to

https://hemanthrajhemu.github.io

encourage browsing. Imagine a multilevel store without floor layout signs (links) to department locations. Navigation through the store would be very difficult. It must be easy for the user to concentrate on content, not on how to get around and the frustrations resulting from being continually lost.

Desire to leave. Respect the user's desire to leave, even though staying might be desirable. (Imagine this multilevel store with only one exit on a single floor, an exit whose location is unknown to the shopper. Now imagine this shopper getting very annoyed and never returning.) Should the user return (the Web site design will not have chased them away permanently, of course), provide guidance to help in reorientation. (Perhaps they also reentered on a different floor from which they last exited.) Never assume that the user will remember the entire previous browsing session. Provide signposts, meaningful page titles, headings, and summaries. Place keywords at the start of all page titles. This will help users pick out pages if they are minimized in the window task bar.

Searching

A search engine or facility is a popular component of Web sites. Fallows and Rainie (2004) report that the use of search engines ranks second only to e-mail as the most popular activity on line. Straub (2005) says that the average Internet user performed 33 searches in June of 2004.

As has been discussed, people usually search on the Web when they have a specific goal or need for which they seek an answer. Their focus may be directed toward something specific, a fact, document, or product; toward gaining an understanding of some more general topic; or the search may be directed toward collecting multiple pieces of information (not necessarily looking for one particular piece), or to evaluate multiple products or answers in order to make a decision. The searching strategy a person employs may, based upon knowledge possessed, involve focused browsing, reviewing site maps and indexes, and reviewing and following site links. A person's strategy may also involve using a search engine.

Currently, as the recent research also suggests, the design of a Web site is the most effective searching tool, not a search facility itself. Experience and many studies have shown that using a search engine can actually reduce a user's chance of success, and that people prefer to browse. Browsing and following navigation links are more likely to lead to the desired result. Most users turn to a search facility only when all else fails. So, the first tool in facilitating user search is a well-designed site, one that is well organized, is easy to scan, and possesses clear navigation links, including a site map and links. But sites must also include a search facility.

Problems with Search Facilities

Many sites seem to deal with their search facilities engines in a haphazard way. A survey has found that 71 percent of users are frustrated by Web searches, and 46 percent find them nerve-racking. Why? The answer can be gleamed from three search components: the user, the search formulation, and the presented search results.

https://hemanthrajhemu.github.io

Not understanding the user. With so much variation among users there can never be a single ideal search interface. Too often, however, the level of searching expertise of the user is not determined. The nature of every possible query or the type of information being searched for is not anticipated. Not considered, as well, are some basic human traits. Few people read instructions, preferring to "try it." Instructions may not even be seen or may be simply ignored. People do not remember things very well. The existence of a search engine on another page, and/or the page's location is easily forgotten.

Difficulties in formulating the search. One study found that the average Web shopper does not know how to use a search function. Another study found that the users did not know what to type in or how to format the query. These problems can occur because of the diversity in user search needs. They may also occur due to numerous interface differences that create inconsistencies in layout and operation. Finally, too often the user is asked to think like the programming behind the search engine, a fruitless task. If the wrong search parameters are chosen, no results will appear.

Many users make many mistakes. People make mistakes in spelling. Fowler and Stanwick (2004) report a study carried out by the British Broadcasting System on users of the search engine BBCi Search. One in 12 search terms were misspelled, translating into over 30,000 search queries with misspellings on a particular day. Lots of people misspell words; 3 percent of all queries in another study contained misspellings, most off by only one letter. Misspelling may be caused by unfamiliarity with technical terms or the English language, by spelling difficulties, or simply by one's finger straying in the keying process. If the search engine does not accept misspelled variations, the chance of a search success is very slim.

Overly literal search engines. Overly literal search engines cannot handle typing mistakes, plurals, hyphens, and other variants of query terms (Nielsen, 2004).

Page titles with low search engine visibility. A page title is the food for a search engine in formulating a search response. Irrelevant titles may not be captured. Titles that are not descriptive or clear may not be understood as relevant by the user. Page titles should begin with descriptive information-carrying words that tell the user exactly what will be found on a page (Nielsen, 2004).

Difficulties in presenting meaningful results. Users are often frustrated with presented results because they do not know what the results mean, why they were presented, or what their relevance is to their query. If the chosen search parameters are too narrowly defined, no results or irrelevant results may appear. If too broadly defined, the user is inundated with results and the right answer is buried within. The user must then sift through all the useless data to try and determine where to go. Users often cannot tell that a "No Matches Found" result occurred because of a simple keying typographical error. The wrong key was pressed. Result descriptions consisting of a few words often make no sense, as do URLs displayed in the results for novice users. Displayed search accuracy ratings are seldom relevant to any one particular user. Goal-oriented users don't care how relevant results are for someone else; their only concern is the information they need.

https://hemanthrajhemu.github.io

Search Facility Guidelines

Search services on the Web will be judged on how well they enable the user to easily find what is needed in the galaxy of information space. These services will also be judged on how well the user confusion and frustration existing in the process is reduced. At this moment, as the browsing versus searching research indicates, human beings are better search instruments than machines. The purpose of a search facility, then, is to bring back information, not data. Specifically, answers to questions, the shortest, clearest, possible, are what really matter. Other studies of a person's search behavior reveals that most searches are simple. They generally involve, as summarized by Fowler and Stanwick (2004)

- Using only two terms per query.
- Running only two queries per session.
- Using Boolean operators or advanced search options rarely.

Other researchers have found that search persistence is weak. People become less successful with each search attempt. Spool and Schroeder (2001) reports that the more times users try a single search without success, the less likely it is that they will find what they are seeking. If users search a topic three or more times, they won't find it at all. Nielsen (2001) found that more than 50 percent of users quit searching e-commerce sites after one failed try. For both studies, the success rates for the first search ranged from 55 to 51 percent, the second search 38 to 32 percent, and the third 0 to 18 percent. To design complex search facilities for success on the first search attempt is very important, according to these studies.

The following sections address guidelines for a search facility. It begins with "Know Your Search User" and continues with "Express the Search," "Launch the Search," "Progressive Search Refinement," "Presenting Meaningful Results" and "Remember the Search." Guidelines for making pages "locatable" conclude the discussion.

Know Your Search User

- Identify the level of expertise of the user.
- Identify the terms commonly used by people in searching.
- Anticipate:
  — The nature of every possible query.
  — The kind of information desired.
  — The type of information being searched.
  — How much information will result from the search.
- Plan for the user's switching purposes during the search process.
- Plan for flexibility in the search process.

In designing search features, knowing your search users is just as important as it is in every other aspect of interface design. The more that is known about users, what they are looking for, and how they may search, the better the search facility can be designed to help them.

https://hemanthrajhemu.github.io

Expertise. Identify the level of the users' expertise in computer usage and in the function or application being designed. Do they need a simple or high-powered interface? A natural language or constrained and guided selection using check boxes and list boxes? Are they comfortable with Boolean operators? Will a help function be necessary?

Terms. Identify the terms commonly used by people in searching. People often use the same words when searching. Make information relevant to these common terms easy to find. Common designer terms and user terms may be substantially different.

Anticipate. Find out what the users' questions may be so every possible result can be included in the engine. Find out what kind of information is desired, a taste, a summary, or extensive detail. Present the proper amount in the results. Determine the type of information being searched, structured fields or full text, navigation pages or destination pages, so the searching function can be properly structured. Estimate how much information will result from the search. Will users be overwhelmed by the results? Should the search be qualified in more ways?

Switching purposes. Plan for the users' switching purposes during the search process. Searching may be a temporary interlude during a lengthy browsing process, or vice versa. What else might the users want to do coincidental with the search? Will it be easy to switch back and forth?

Flexibility. Searching strategies may vary. A person may want to search narrowly at one time, broadly the next. Can different user searching strategies be easily handled? Can changing the strategies also be easily handled?

Ultimately, the sophistication and complexity of the search operation must be appropriate for the users, the type of application, and the workflow.

Express the Search

- What:
  - — Design to allow simple searches.
  - — Design to search the entire site.
  - — Design for the Web site's information and the user's needs.
  - — Integrate searching and browsing.
- Where:
  - — Make the search facility prominent on the homepage.
  - — Include a search facility on every page.
- How:
  - — Permit users to specify the extent of searches
    - Within a section.
    - Across a site.
    - Within specified sources.
    - Globally.
  - — Provide methods of specifying the search, including
    - Templates.
    - Keywords.

https://hemanthrajhemu.github.io

- Variants.
- Phrases.
- Partial matches.
- Synonyms.
- For large sites include an internal glossary of terms and a thesaurus.

— Provide a spell checker.

— Provide search controls, including
- A text box large enough to enter a minimum of 20 characters.
- Structured controls.
- Check boxes.
- List boxes or drop-down list boxes
- A command button.
  - Labeled: *Search*.
  - Located to right of search text box.

— Provide separate interfaces for simple and advanced search.
- Place "Advanced Search" link under text search box.

— Provide guidance and assistance.
- Present clear instructions.
- Offer online help.
- Offer a search wizard.

What. The search function must be easy to use. As research has shown, most users tend to employ simple search strategies. Few search terms are used and even fewer search features employed. On many Web sites, only a simple search facility is needed, an advanced search facility not being necessary. It is also important that the probability of being successful on the initial search be high.

Design to search the entire site. People tend to believe that a search will be performed over an entire Web site. If only a portion of the site will be searched, clearly inform users of that fact. Structure the searching function to the Web site, the information it contains, and the user's anticipated needs. Don't use generic applications. A well-structured and organized content is a key factor in achieving good search results. Searching and browsing should also be closely integrated. Upon arriving a user may not know whether he or she wants to browse or search.

Where. Make the search option prominent on the homepage. No matter how well one supports navigation, a user may still want to search immediately after accessing a site. Include a search facility on every page, as well. Where a user is when a decision is made to search can never be predicted. Also, permitting a search from any Web site point facilitates a new user, or a user who wanders into the Web site at that point. These users do not have to first learn the conceptual organization of the site before they find something of interest to them.

How. Studies show that users perform better and have higher subjective satisfaction when they can view and control a search (Koenemann and Belkin, 1996). Search options should be clear. Permit users to specify and control the extent of their

https://hemanthrajhemu.github.io

searches, either confining them to within a section, across a site, within specified sources (such as libraries, educational institutions, particular kinds of businesses or industries, a particular language or geographical area, and so on), or globally across the Web. Specifying extent will also help the user maintain orientation. For Web site facilities, the initial target should be the Web site's contents.

Offer different ways to search, including by parameters such as *keywords*, *phrases*, *synonyms*, and *variants*. Do assist in identifying keywords in large sites and provide an internal glossary of terms and a thesaurus. Variants allow relaxation of or qualify search constraints, such as allowing case insensitivity, phonetic variations, abbreviations, multiple formats (such as date and time) or synonyms.

Templates also facilitate the use of search engines. A template consists of predefined keywords that provide guidance in the selection of search terms. People who are highly experienced on a certain topic can provide keywords. The keywords can then be selected from the list, or, alternatively, give the user ideas on how to formulate their own inquiry. Organize templates as a hierarchy of keywords to restrict the user's initial search sets. Fang and Salvendy (1999) found that people using templates found 70 percent more of target Web sites.

Always make every effort to catch user errors and work around them. Providing a spell checker will reduce also typing and spelling errors that prevent matches from being found.

Search Controls. Provide search controls, including a text box, structured controls, and a command button. The search text box should be large enough to enter a minimum of 20 characters. To constrain searches or designate variants use structured controls, including check boxes, list boxes, or drop-down list boxes. Default to the most likely selection. (Use of either list boxes or drop-down list boxes will be determined by screen space constraints; see Step 7.) Provide a command button labeled "Search" to the right of search text box. For a grouping of search controls, position the search command button at the end of the field completion sequence. (See Step 7 for detailed command button size and location guidelines.)

Separate interfaces for simple and advanced search. A simple search will normally consist of a text box for entering keywords and phrases. If an advanced search is necessary and included, place an "Advanced Search" link under the search box.

Guidance and assistance. Also provide guidance and assistance. Present clear instructions and examples and offer online help and a search wizard. This is most important for inexperienced Web users. Guidance and assistance are addressed in more detail in Step 9.

Launch the Search

- Permit search activation by clicking on the command button or pressing the Return key.
- In search refinement, permit changes to a parameter to automatically produce a new set of results.

https://hemanthrajhemu.github.io

Activate the search with an explicit action — clicking the command button or pressing the return key. This will indicate that the parameter specification process is completed. In search refinement, permit changes to a parameter to automatically produce a new set of results. This will speed the refinement process.

Progressive Search Refinement

- Allow the user to control the size of the result set by providing a simple mechanism to
  - — First perform a rapid rough search that reports only
    - The number of items in the result set, or
    - A preliminary list of topical matches.
  - — Then perform a refinement phase to narrow the search and retrieve the desired result set.

To help narrow the field for large searches, and avoid overwhelming the user with hundreds of items to search through, support progressive search refinement. A progressive search consists of two phases. The first is a quick search using specified criteria. For the second phase, two alternatives are possible. The first alternative is to simply report back the number of matches found based upon the search. For example, a message might say, "A search on 'Automobiles' found 977 matches." If the number is too large, the user can continue to refine the search until a manageable number is found. If too small, the search criteria can be relaxed. The second alternative is to present a list of topical matches. The user can then select the relevant topic and continue further refinement. For example, the search may be for "automobiles." Returned topics may include such items as "New," "Used," "Purchase," or "Lease." Designating "New" might then result in "American," "European," or "Japanese."

Selection of variations and changing of parameters should be easy and convenient to do. Also, provide messages with suggestions to guide users in the refinement process. Suggesting possible similar words, or word spellings, would be helpful. So would guidance concerning whether the search involves looking for whole words such as "tire," or should include partial words such as "entire" or "tired." While the search parameters may allow specification of such criteria, the user may have neglected to set them properly.

Present Meaningful Results

- Goal:
  - — Provide exactly the information or answer the user is looking for.
  - — Present it in a language and format that is easy to understand and use.
- Criteria summary:
  - — Present a summary of the search criteria with the search results.
- Explanatory message:
  - — Provide a meaningful message to explain search outcomes.
  - — Indicate how many items compose the search result set.

https://hemanthrajhemu.github.io

- Results presentation:
  — Present a textual listing that is
    - Concise.
    - Ten to 50 returns in length.
    - Arrayed in order of relevance.
    - Clear.
    - Easily scannable.
- Permit the user to
  — Modify the result set sequencing.
  — Cluster the result set by an attribute or value.
- For multipage listings, make obvious the link to the next search result page. For results with only one item, immediately present the result page.
- Present polite and useful messages.

Goal. In presenting meaningful results, the objective is to provide exactly the information or answer the user is looking for. Presenting information or data in great quantities will test the user's patience and likely "hide the tree in the forest." Presenting irrelevant results often leads to abandonment of the search. To speed the search results review process, present information in a language and format that is easy to understand and use.

Criteria summary. Present a summary of the search criteria with the search results. Never assume that the user will remember what the search parameters were.

Explanatory messages. Provide meaningful messages explaining search outcomes that aid progressive refinement. Include how many items compose the search result set.

Results set presentation. Result listings should be concise, arrayed in order of relevance, clear, and easily scannable. A *concise* listing is one that displays the least possible amount of return descriptive information for determining that a match meets the user's search needs. Presenting 10 to 50 search returns per results page optimizes both performance and preference (Bernard et al., 2002a). This study did find, however, that a listing of 50 returns resulted in faster scanning and information location, and was preferred by users. Eysenbach & Kohler (2002) report that users typically peruse only the first page of search results, 93 percent of the links visited being within the first 10 results. Joachims et al. (2005) found that 42 percent of the users selected the top item in a relevance order listing (see below). When the number one and two items were (without telling the participants) swapped in position (the second item in the most relevant listing now being placed on top), the new top item was still selected 34 percent of the time. Nielsen (2005d) concludes this may have occurred because 1) search engines are good at judging relevancy and almost always place the best on top, 2) users assume the search engine places the best on top, or 3) people tend to select the top item because it is first on the list. He suggests the answer is a combination of all three factors.

https://hemanthrajhemu.github.io

Present the results in the most useful way possible. A list that *is relevance-ordered* places exact or best matches first and follows with those less close. A relevance-ordered list may also be sorted by a criterion reflecting the users' need — alphabetically, chronologically, by date, and so forth.

A *clear* listing provides enough information for the user to determine whether to proceed. Textual listings are usually much clearer than graphical listings. Consider graphical listings, such as maps, however, if the information will be more clearly presented in a graphical format. While being concise, a one to three word description, in most cases, will not provide the necessary clarity. URLs should not be presented because they tend to confuse more than help. Page upload dates should be presented only if they are of relevance to the searching user. A news site date, for example, is usually important to the user. For most other kinds of sites, including the date would not be relevant.

Present an easily *scannable* results listing. A vertical array with highlighted keywords is best. Allow the user to modify the result set by changing its sequencing; from alphabetical to chronological, for example. Allow the user to cluster the result set by an attribute or value, or by presented topics. For *multiple-page* listings, the link to the next search page should be clearly presented at the end of the listing currently being displayed. This link should not be displayed when no more listing pages follow. For results that find only one item to link to, immediately present the item page instead of presenting a one-item listing.

Messages. If no items are found to present in a listing, provide an informational message telling so. If search parameters are incorrect, politely provide advice telling how to correct the problem.

Remember the Search

- Save the search entries.
- Save the search results automatically.
- Let the user save the results.

There are three kinds of search histories that can be saved (Fowler and Stanwick, 2004).

Search entry. With search entry, the search terms are saved, not the results of the search. This is a form of auto-complete and is most useful when people tend to jump back and forth between working and searching.

Search results. The search result lists are saved automatically. This method is most helpful when people may want to review their earlier search results. The search term will have to be reentered to get the newer information that is available since the saved list.

User saved. Found articles are saved by the user in a binder. This is useful when people need to hold on to something for a long period of time.

https://hemanthrajhemu.github.io

Destination Pages

- Describe how the page relates to the search query.
  — Provide page summary.
  — Highlight keywords.

When linked to a page through a search facility, the user should know why the page was found. Start the page with a summary of its contents. Also highlight the words in the page that were keywords used in the search.

Locatability

- Provide text-based content.
- Repeat keywords frequently throughout the text.
- Provide a page title
  — That possesses meaningful keywords.
  — Whose first word is its most important descriptor.
  — That makes sense when viewed completely out of context.
  — That is different from other page titles.
  — That is written in mixed-case, headline style, with no highlighting.

A Web site must be easily found by a search facility. To ensure a Web site's *locatability*, use the following guidelines.

Text content and keywords. Text-based content is easily accessible to search engines. Provide text-based content as much as possible. Include as keywords on the page all possible query terms that can be used to search for the topic presented on the page. Repeat the keywords frequently throughout the text. Do not add keywords only peripherally related to the page's contents. Consider using a professional indexer to create keywords for full-text searches.

Titles. Page titles must be carefully designed to provide useful information. They should contain as many keywords as possible. While a title may contain 60 characters, ensure that the first 40 characters adequately describe the page topic. Titles are often truncated in navigation menus and by search facilities. The title's first word should be its most important descriptor. This word is most easily noticed in the scanning of long lists. Never begin a title with a generic term such as "Welcome" or "Page," or with an article such as "The." Give different titles to different pages. If page titles addressing the same topic must begin with the same word, end the title with words that explain the differences between them. A title must also make sense when viewed completely out of context, with no supporting content, or arrayed in a listing with other titles. Write titles in mixed case using the headline style. Do not use highlighting for keywords. A single keyword might be emphasized by putting it in uppercase, but be conservative in this regard. Never use uppercase for the first word in a title because its position is sufficient emphasis.

https://hemanthrajhemu.github.io

Intranet Design Guidelines

- Provide a single homepage containing at least
  - — A directory hierarchy.
  - — A search facility.
  - — Current news.
- Homepage Layout
  - — Provide a
    - Top horizontal bar.
    - Left column.
    - Middle area.
    - Right column.
- Present a visual style that is
  - — Different.
  - — Distinguishing.
  - — Unified.
- Orient the intranet Web site toward tasks.
- Include many options and features.
- Develop a strong navigational system.

Intranets are internal closed systems that use the capabilities of the Internet. In recent years intranets have increased greatly in usability, but still lag behind their sibling because of a lack of resources to address usable design. Lack of usability for an intranet translates into a significant loss of employee productivity. Nielsen (2006a) reports that IBM estimates it has saved $194 million per year because of an application redesign, and each employee saves 72 minutes per month.

Nielsen (2006a) also describes the following intranet trends:

- More consistency in look and feel.
- More task-based information architecture.
- Migration of Web trends to the intranet.
- Increased mobile access.

Design consistency is improving, as is the use of design templates. More task-based information architectures are appearing rather than architectures based upon a company's business units. Migration of Web concepts to the intranet is occurring because many employees use the Internet at work and at home. Finally, while mobile intranet use is not yet widespread, intranet applications are beginning to appear.

Nielsen (1997b, 1999a, and 2005c) suggests the following intranet design guidelines:

Homepage content. Provide a single homepage containing at least a directory hierarchy, a search facility, and a current news segment. The directory will provide a structured overview of the site's content. The search facility will provide a means of quickly accessing the site's index. Unlike a generic facility, this local facility can present information about the importance of the item or topic of interest to the organization. The news segment can include information about the company and things that are of interest to employees.

https://hemanthrajhemu.github.io

Homepage layout. Some standardization of intranet homepage layouts is beginning to appear. These components are a *top horizontal bar* that is typically used for a logo, global navigation, and a search box. It is 100 pixels tall with a colored background. A *left column* typically is used for a navigation bar with detailed navigation and/or a contextual list of options for the current intranet subsite. It is 200-250 pixels wide with a colored background. A *middle area* featuring one or two columns contains a list of news headlines, photos, illustrations, and boxes with "portlets" to the most important features and applications. It is 400-600 pixels wide and contains a white background. Finally, a *right column* typically is used for a set of stacked boxes, some with colored contents or pictures. It contains a white background and is 200-250 pixels wide.

Visual style. Because the Internet and an intranet are different information spaces, a complementary but distinguishing look will quickly inform the users should they wander outside of the closed internal net to the public site. The style should also be unified and consistent throughout its entire structure.

Task-oriented. An intranet will be more task-oriented and less promotional than a regular Web site.

Options and features. Because employees will frequently use the site, it will be understood and learned faster (if unified in design). More options and features can exist because feelings of intimidation and being overwhelmed are much less likely to occur.

Navigational system. A stronger navigational system will be necessary because the intranet will encompass a much larger amount of information. Movement between servers may be necessary.

Extranet Design Guidelines

- To distinguish the extranet from the Internet, provide a subtle difference in
  - — Visual style.
  - — Navigation.
- Provide links to the public Internet site.

An extranet is part of an organization's intranet that may be accessed from the Internet. Because it is a mixture of the Internet and the intranet, its design should reflect this. Because its users will access it from the Internet, its visual style and navigation should be similar to the Internet site to indicate companionship, but subtly different to connote its independence. Links to locations on the public Internet site may be included (Nielsen, 1997b).

Small Screens

Small displays are becoming commonplace on devices such as the personal digital assistant (PDA) and the cell phone. Guidelines are evolving, but as technology quickly advances, it is difficult to hit a moving target. Some research has been accomplished. Its results, and some broad guidelines, are presented as follows:

https://hemanthrajhemu.github.io

- Prose text.
  - Use Rapid Serial Visual Presentation (RSVP) for presenting prose text.
    - There are no performance or preference differences between 10- and 3-line displays.
    - Use 20-point text presented at speeds of about 250 words per minute.
- Other guidelines.
  - Use choice selection, not typing.
  - Provide consistency.
  - Provide stability.
  - Provide feedback.
  - Be forgiving.
  - Provide obvious clickable graphics.

Prose text. Bernard et al. (2001e) studied Rapid Serial Visual Presentation for reading prose text on small displays and found RSVP yielded an acceptable level of reading performance. They also found no performance or preference differences between 10-line and 3-line presentations. Russell et al. (2001) found the optimum was 20-point type presented at speeds of 250 words per minute.

Other guidelines. Weiss (2002) recommends the following. Use choice selection instead of typing whenever possible. Keying text on a small device can be difficult. Provide consistency in terminology within and between applications. While consistency between desktop applications and small screens is desirable, occasional abbreviations may be necessary on the small screens. Maintain abbreviation consistency throughout, however. Provide stability. If a connectivity problem occurs, enable a person to pick up where he or she left off. Previously entered information should not have to be reentered. Provide feedback regarding what the application is doing, perhaps through an assigned information key. Be forgiving and tolerant of user errors. Also provide an undo function. Finally, provide obvious clickable graphics. Use high-contrast images for clickable graphics. Non-clickable graphics should be subdued.

Weblogs

Weblogs are a type of Web site. Therefore the guidelines scattered throughout this book should be followed. Some important Weblog guidelines follow. This discussion is based mostly on Nielsen's (2005f) top ten Weblog design mistakes.

- Provide an author biography.
- Provide an author photo.
- Provide for easy scanning.
- Provide meaningful links.
- Link to past important content.
- Categorize postings.
- Publish regularly.

https://hemanthrajhemu.github.io

- Avoid mixing topics.
- Remember you leave an audit trail.

Author biography. A writing that is signed has more credence, and is more trustworthy than one that is anonymous. Readers want to know something about the author, including credentials, experiences, and reasons for the comments.

Author photo. A photo offers a more personable impression of the author. It enhances credibility and connects the virtual and physical worlds. People relate more easily to someone they have seen. It also makes the author recognizable.

Easy scanning. Like all Web sites, headlines, important points, and text should be easily scannable. An article's objective must be quickly understandable. If not, the entire posting may not even be read. Headlines should be descriptive of the content for representation in search engines, news feeds, and other external environments.

Meaningful links. Write link text to indicate (or include in its immediately surrounding text) where the link leads and what will be found at the destination. People generally do not like voyaging to the unknown.

Past important content. Highlight the most important past postings and link directly to them. Otherwise they may never be found. Also link to past postings in the newer postings to provide background and context. Never assume the reader has been there since the beginning.

Categorize postings. Categorize postings by topics so people can easily get a list of all postings on a particular topic. Be conservative in posting to categories, ensuring that all postings are relevant to that particular topic.

Publish regularly. Provide a regular publishing schedule so readers can anticipate when and how often updates will occur. Without a regular schedule, readers may be lost.

Avoid mixing topics. Focus your content on a topic. People often visit to read an entry that interests them. If their topic of interest appears only sporadically among a wide range of other topics, they are unlikely to return because of the effort that must be expended.

Audit trail. Postings may last forever, leaving a long audit trail. Think twice before posting something that might eventually come back to haunt you. Never, unintentionally, burn any bridges.

Statistical Graphics

A *statistical graphic* is data presented in a graphical format. A well-designed statistical graphic, also referred to as a chart or graph, consists of complex ideas communicated with clarity, precision, and efficiency. It gives its viewer the greatest number of ideas, in the shortest time, and in the smallest space, and with least possible clutter. It will also induce the viewer to think of substance, not techniques or methodology. It will provide coherence to large amounts of information by tying them together in a meaningful way, and it will encourage data comparisons of its different pieces by the eye. In general, a statistical graphic should possess the following qualities:

https://hemanthrajhemu.github.io

- The objective and use of the graph should be obvious and apparent.
- The graph type should be recognizable.
- The graph type should help users understand the data more easily.
- The data should be formatted and presented correctly.
- The data should be formatted and presented for the using audience.
- The graph should avoid distortions by telling the truth about the data.

Much of this material on statistical graphics is based upon Tufte (1983), Smith and Mosier (1986), and Fowler and Stanwick (1995 and 2004).

Use of Statistical Graphics

- Graphs are used for
  — Simple comparisons of data.
  — Presentation of changes over time.
  — Statistical analysis.
  — Illustration of proportions.
- Reserve for material that is rich, complex, or difficult.
  — Less than or equal to 3 numbers — use a sentence.
  — For 4 to 20 numbers — use a table.
  — More than 20 numbers — use a graph.

A few data sets may be compared using *bar charts* while changes over time can be shown using *line graphs*. Statistical analysis can be accomplished using *histograms* and *scatterplots*. Proportional graphs illustrate differences in size number, or value while not requiring a scale. Graph types include *surface charts* and *pie charts*.

Statistical graphics should be reserved for large sets of data with real variability. The power of graphics should not be wasted on simple linear changes or situations in which one or two numbers would summarize the result better. Tufte (1983) says that tables usually outperform graphics on small data sets of 20 or fewer numbers, or when data sets are noncomparative or highly labeled. Tables are also better if the data must be studied or very specific information must be retrieved (Coll et al., 1994). Wainer (1997) suggests that for 3 or less numbers a sentence is the best descriptor; for 4 to 20 numbers use a table; and for more than 20 numbers use a graph.

Components of a Statistical Graphic

Most statistical graphics have at least two axes, two scales, an area to present the data, a title, and sometimes a legend or key, as illustrated in Figure 3.71. Pie charts are the exception to this general rule. Guidelines for graph components include the following.

https://hemanthrajhemu.github.io

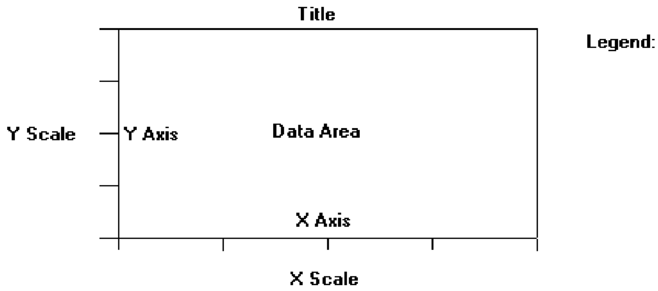


Figure 3.71: Components of a statistical graphic.

Data Presentation

- Emphasize the data.
- Minimize the nondata elements.
- Minimize redundant data.
- Show data variation, not design variation.
- Provide the proper context for data interpretation.
- Restrict the number of information-carrying dimensions depicted to the number of data dimensions being illustrated.
- Employ data in multiple ways, whenever possible.
- Maximize data density.
- Employ simple data-coding schemes.
- Avoid unnecessary embellishment of
  — Grids.
  — Vibration.
  — Ornamentation.
- Fill the graph's available area with data.

The most important part of a graphics display, as with an alphanumeric display, is the data itself.

Emphasize the data, minimize the nondata elements. A person's attention should be drawn to the measured quantities. The largest share of the graphic's "ink" should present data. Nondata elements — such as elaborate grid lines, gratuitous decoration, and extensive, detailed, and wordy labels — draw attention to themselves and mask the data. So, nondata elements should be minimized, or eliminated entirely.

Redundant data. Information that depicts the same value over and over should also be minimized or eliminated. Redundancy, on occasion, can be useful, however. It may aid in providing context and order, facilitating comparisons, and creating an aesthetic balance. Use redundancy only if necessary.

https://hemanthrajhemu.github.io

Data variation. Show data variation, not design variation. Each part of a graphic generates visual expectations about its other parts. The expectancies created in one part should be fulfilled in other parts so the viewer does not confuse changes in design with changes in data. Scales should move in regular intervals; proportions should be consistent for all design elements. If the viewer confuses changes in design with changes in data, ambiguity and deception result.

Proper context. Provide the proper context for data interpretation. Graphics often lie by omission. Data for making comparisons or establishing trends must always be included to provide a proper reference point. "Thin" data must be viewed with suspicion. The graphic in Figure 3.72, for example, might have a number of possible interpretations, as illustrated in Figure 3.73. All important questions must be foreseen and answered by the graphic.

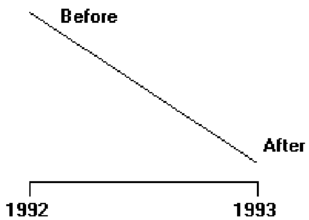


Figure 3.72: A change between 1992 and 1993 without proper context for interpretation.

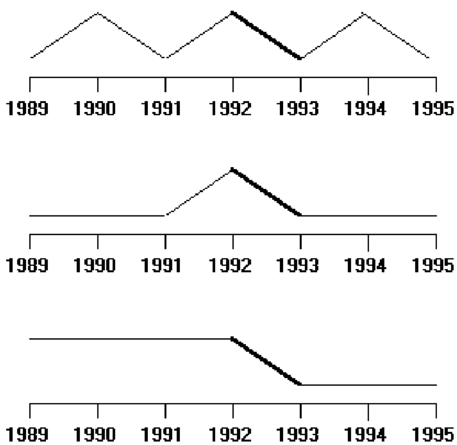


Figure 3.73: Changes between 1992 and 1993 with proper contexts for interpretation.

https://hemanthrajhemu.github.io

Restrict information-carrying dimensions. Restrict the number of information-carrying dimensions depicted to the number of data dimensions being illustrated. Displaying one-dimensional data in a multidimensional format is perceptually ambiguous. With multidimensional data, changes in the physical area of the surface of the graphic do not produce an appropriate proportional change in the perceived area. Examples of multidimensional formats used to display one-dimensional data would be different-sized human bodies to indicate populations or different-sized automobiles to indicate the number of cars. Often the impression on the viewer is that the change is actually much greater than it really is. This problem can be avoided if the number of information-carrying dimensions on the graphic is restricted to the number of data dimensions being illustrated.

Employ data in multiple ways. Whenever possible, employ data in multiple ways. Parts of a graph can be designed to serve more than one purpose. A piece of data may at the same time convey information and also perform a design function usually left to a nondata element. Some examples are

- A grid to aid readability of a bar chart, instead of being inscribed on the graphic background, may be positioned within the bars themselves, as illustrated in Figure 3.74.
- The size of what is being measured can be conveyed through the size of the graphical element, the intensity through color or level of shading.

Graphics can be designed to have multiple viewing depths. The top level provides an overall view, each succeeding level an ever-increasing closer view. They may be also designed to have different viewing angles or lines of sight.

The danger in employing data in multiple ways is that it can generate a graphical puzzle. A sign of a puzzle is that the graphic, instead of being interpreted visually, must be interpreted verbally. Symptoms of a puzzle are frequent references to a legend to interpret what is presented and extensive memorization of design rules before one can comprehend what is presented. By contrast, a well-designed multiple-function graphic permits a quick and implicit transition of the visual to the verbal.

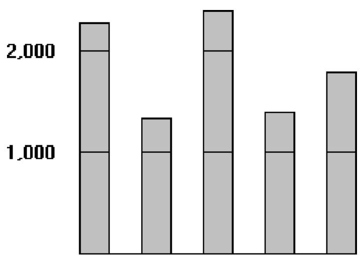


Figure 3.74: A piece of data (line in the bars) performing a nondata function.

https://hemanthrajhemu.github.io

Maximize data density. In graphics, more is better than less—the greater amount of information displayed, the larger the number of visual comparisons that can be made, improving comprehension. This is true because the eye can detect large amounts of information in a small space. Simple things belong in a table or in the text.

Data density of a graphic can be maximized in two ways: enlarging the data matrix or shrinking the graphic. Enlarging the data matrix involves displaying as much information as possible. If the graphic becomes overcrowded, techniques such as averaging, clustering, smoothing, or providing summaries can reduce the number of elements to be displayed. Shrinking the graphic means reducing it in size, but screen resolution may impose limitations on how much shrinking can be performed.

If visual differentiation in the types of data being displayed is necessary, use simple coding methods in the areas being depicted. Elaborate schemes or patterns can be eye-straining and can actually impede the flow of information from the graphic to the viewer. Some possible coding alternatives include

- Varying shades or densities.
- Labeling with words.
- Varying colors.

Avoid unnecessary embellishment. All pieces of a graphic must tell the viewer something new. An unnecessary embellishment is "chartjunk." It does not add anything new to the graphic's meaning. It is decoration or noise that hinders assimilation of the message being communicated. Nondata elements and redundant data are forms of chartjunk. Three other kinds are vibration, heavy grids, and ornamentation.

A grid carries no information, contributes noise, and focuses attention away from the data. An excessively heavy grid can even mask the data. Grids should be suppressed or eliminated so they do not compete with the data. When a grid serves as an aid in reading or extrapolating, it should, of course, be included. Its tendency to overwhelm can be reduced by constructing it with delicate lines or muted colors and placing it behind the data.

The eye is never absolutely still; it produces continuous slight tremors that aid visual acuity. The result is that, when small patterns, lines, boxes, or dots are viewed, they shimmer or vibrate. This vibration can be distracting; examples are illustrated in Figure 3.75. Although eye-catching, vibrations can also strain the eye. Simple data-coding schemes, such as using shades of color, are much more effective.

When the graphic is overwhelmed by decoration, it is very ineffective. Ornamentation can take many forms: extensive use of color when it is not necessary; creating multidimensional graphics when single dimensional will do; pointless use of vibrating patterns; or forcing data into a graphic when a table

https://hemanthrajhemu.github.io

would work much better. Ornamentation is more effective as a piece of art hanging on a wall. It is a symptom of "See what I can do with my computer" rather than an effort to provide the user with the data in the most comprehensible way possible. The best graphic display is the simplest graphic display.

Fill the display area. For ease of interpretation and efficiency, the graphic's data should fill up the entire display area within the axes. If it does not, the scale or the graphic is too large.

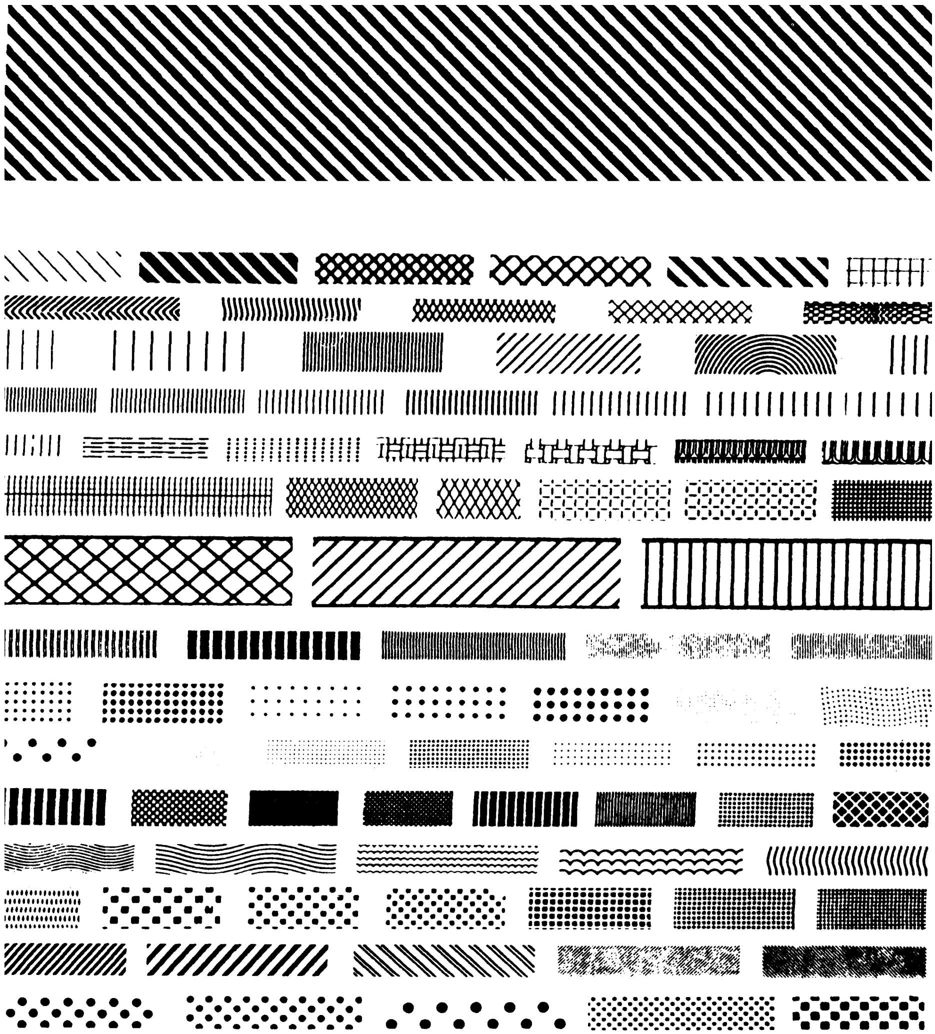


Figure 3.75: Examples of patterns creating vibrations.

https://hemanthrajhemu.github.io

Axes

- Values on an axis should increase as they move away from the origin.
- Use the horizontal axis (X) to show time or cause of an event (the independent variable).
- Use the vertical axis (Y) to show a caused effect (the dependent variable).
- Provide additional axes when appropriate. For example, to show
  — Two scales for the same data.
  — Two or more related graphs within the same area and frame.
- Assure the area enclosed by the axes is the proper size for the displayed data.

Values on an axis should increase as they move away from the origin point. If the numeric values displayed are positive, the origin point will be the lower-left point of the graphic. If the data includes negative values and the axes must extend in both directions from the zero point, position the origin in the center of the graph.

Use the horizontal axis (X) to show time or cause of an event (the independent variable). Use the vertical axis (Y) to show a caused effect (the dependent variable). When the X-axis plots time intervals, the labeled points should represent the end of each time interval. The X-axis may also be called the *abscissa* or *category* axis, the Y-axis the *ordinal* or *value* axis. If the graphic possesses three dimensions, the third axis is called the Z-axis, reflecting the graph's plane. The axes can be reversed, especially for bar graphs.

Additional axes can also be displayed, if appropriate. For example, temperature in Celsius can be displayed in a top X-axis, temperature in Fahrenheit in a bottom X-axis. Speeds or distances in kilometers and miles can be displayed simultaneously in a similar manner. Putting two datasets with a close relationship within the same frame is also possible. The Y-axes may be used to show the different but related scales.

The area enclosed by the axes should be of the proper size for the data. It should not be too large or too small.

Scales and Scaling

- Place ticks to marks scales on the outside edge of each axis.
- Employ a linear scale.
- Mark scales at standard or customary intervals.
- Create relatively square grid cells.
- Start a numeric scale at zero (0).
- Keep the number of digits in a scale to a minimum.
- For scales containing decimals, put zeros in front of the decimal.
- If data points are on the zero (0) line, offset the scale.
- Display only a single scale on each axis.
- For large data matrices, consider displaying duplicate axes.
- Provide aids for scale interpretation.
- Provide scaling consistency across two or more related graphics.
- Clearly label each axis in a left-to-right reading orientation.

https://hemanthrajhemu.github.io

A *scale* is a set of measurement points or markers. There are three major types of scales: category, quantitative, and sequence. A *category* scale, also called *qualitative* or *nominal*, consists of an unordered or unsequenced series of words or numbers. A *quantitative* scale, also referred to as *value*, *interval*, *numeric*, or *amount*, consists of sequential numbers with uniform spacing. A *sequence* scale consists of words or numbers in a customary and standard order, such as the months of the year.

Scaling is the positioning of data in relation to these points or markers. Choose an appropriate scale for both graph axes. If the scale is too expanded, the effect may be exaggerated, if too small, the effect may be underreported. Standard scaling practices are these:

Tick marks. Place ticks to mark scales on the outside edge of each axis. Placing them outside will prevent a tick from interfering with data located near the axis.

Linear scale. Employ a linear scale. Most people are more familiar with linear scales than with logarithmic or other nonlinear scales. These latter kinds are often interpreted inaccurately. If necessary or desirable, two X or Y scales can be used, one displaying a linear scale, the other a logarithmic scale.

Scale markings. Mark scales at standard or customary intervals to aid comprehension. Familiar standard intervals are 1, 2, 5, 10, and multiples of 10. Familiar customary intervals include the days of the week and the months of the year. Construct scales with tick marks at these intervals. To aid visual comprehension, it may be necessary to provide intermediate marks as well. Intermediate marks should be consistent with the scale intervals shown.

Grid cells. Create squarish grid cells by using the same distance between ticks on both axes. Distorting the axes will distort the data as well.

Start a numeric scale at zero. Using zero as the starting point on a scale aids visual comparisons because zero is an expected starting point. If a zero point is omitted because of the nature of the data, this omission should be clearly indicated in the graphic.

Minimization of scale digits. Keep the number of digits in a scale to a minimum. Smaller numbers aid understanding. Round off all numbers to two digits or less.

Scales containing decimals. Place zeros in front of decimal numbers so the decimal point is not missed.

Data points on zero line. If a substantial number of data points are on the zero (0) line, offset the scale to enhance readability. Otherwise, they may not be seen.

Single scale on each axis. Display only a single scale on each axis. Avoid multiple scales associated with a single axis. For all but the most experienced people, multiple scales can be confusing and can lead to interpretation errors. Meanings can also be greatly distorted. If multiscale graphs must be used, permit the user to select any data curve individually and have the computer highlight its corresponding scale.

Duplicate axes for large matrices. For large data matrices, consider displaying duplicate axes. The readability of large data matrices is improved if the X-axis scale appears at the top as well as the bottom of the graph, and the Y-axis scale at the right as well as the left side.

https://hemanthrajhemu.github.io

Scale interpretation. Provide aids for scale interpretation. When reading accuracy is extremely critical, provide computer aids for interpretation, such as the following:

- Displaying a fine grid upon request.
- Vertical and horizontal rules that the user can move to the intersection point.
- Letting the user point at a data item and having the computer then provide the exact values.

Consistency. Provide scaling consistency across two or more related graphics. If comparisons must be made between multiple graphs or charts, use the same scale for each. Data sets scaled differently lead to interpretation errors.

Labeling. Each scale axis should be clearly labeled in a conventional left-to-right reading orientation. A complete description of the values, with measurement units, should be provided. Do not, however, overwhelm the graph with tick labels. Default to a reasonable number of tick marks and labels.

Proportion

- Provide accurate proportion of the displayed surfaces to the data they represent.
- Provide proper proportion by
  — Conforming to the shape of the data.
  — Making the width greater than the height.

The displayed surfaces on graphics should be directly proportional to the numeric qualities they represent. Failure to display the correct proportions can create false impressions of magnitudes of differences in sizes or changes. This kind of graphical distortion can be eliminated through clear, detailed, and thorough labeling, a topic to be addressed shortly.

Provide proper proportion. When the relative proportions of a graphic are in balance, it looks better. Graphics should tend toward the horizontal, assuming a greater length than height. There are a number of reasons for this recommendation. First, people prefer this shape. Second, it is easier to read words arrayed left-to-right. Third, many graphics plot cause and effect relationships, with effect on the vertical axis and cause on the horizontal. An elongated horizontal axis helps describe the causal variable in more detail. If, however, the data being displayed suggests a shape either square or higher than wide, conform to the shape suggested by the data.

Lines

- Data lines should be the heaviest.
- Axes lines should be of medium weight.
  — Extend the lines entirely around the graphic.
- Grid lines should be very thin or absent.
  — Let the user turn the grid on or off.

https://hemanthrajhemu.github.io

The most important part of the graphic is the data. Emphasize the data by making the data lines the heaviest. Of secondary importance are the axes lines. Display them in a medium thickness. Axes lines should be extended entirely around the graphic to create a rectangle (or box). This will define the graphic area and help focus attention on the data itself.

Grid lines should be avoided if at all possible, unless absolutely needed for accurate data interpretation. When included, grids should be in the background. Let the user turn the grid on or off as needed.

Labeling

A graph also needs a label for its title, the axes, its key or legend, and its data. First, however, some general labeling guidelines follow.

- Employ clear, detailed and thorough labeling.
- Maintain a left-to-right reading orientation.
- Integrate the labeling with the drawing.
  — Do not curve letters to match the shape of curved lines.
- Use only one typeface, font, and weight.
  — For emphasis, use different type sizes.
- Do not separate labeling from the data through ruled lines.

Clear and detailed. Employ clear, detailed, and thorough labeling. Words should be fully spelled out. Follow standard capitalization schemes, using both upper and lowercase, with lowercase used for textual information. Use the simplest and shortest word forms possible. If an abbreviation is necessary, use standard word abbreviations.

Left-to-right. Maintain a left-to-right reading orientation. Display all labels horizontally. Avoid words that are organized vertically or words that run in different directions. Whereas non-horizontal words on hard-copy graphics can easily be read by turning the paper, this screen capability is not available, nor yet easy to accomplish.

Integrate. Integrate the labeling with the drawing. Explanations on graphics help the viewer and should be incorporated as much as possible. Words are data, and they can occupy space freed up by nondata elements or redundant data. Integrating words and captions with the graphic eliminates the need for a legend and the eye movements back and forth required to read it. Also, incorporate messages to explain the data, and label interesting or important points. Never curve letters to match the shape of curved lines. This is terribly distracting. Run all text horizontally.

One typeface. Use only one typeface, font, and weight. Using the same type style for graphics and text aids the visual integration of the two. If text needs to be emphasized, use different type sizes.

Separation. Do not separate labeling from the data through ruled lines. Again, this creates visual noise and impairs proper associations.

https://hemanthrajhemu.github.io

Title

- Create a short, simple, clear, and distinctive title describing the purpose of the graphic.
- Position the title above, centered, or left-aligned to the rectangle formed by the extended axes.
- Spell it out fully, using a mixed-case or uppercase font.

A title should be brief and descriptive of the graphic. A title may be centered or flush left to the rectangle formed by the extended axes. Marcus (1992) feels that left-aligning it yields a stronger composition. Titles should be spelled out fully, and may be displayed larger, bolder, and in mixed- or uppercase font.

Axis Labels

- Center the X-axis label below its scale.
- Center the Y-axis label above the scale or adjacent to the scale.
- State units of measurement.
- Provide information about the source of the data.

Center axis labels. Center the X-axis label below its scale. Center the Y-axis label above the scale or position adjacent to the scale. Avoid turning the label sideways whenever possible. If it must be positioned sideways, center it facing the scale. A facing position will make it more difficult to read, however. Never stack words on the Y-axis in a vertical format because the shape of the words will be destroyed and they will be difficult to read.

Units of measurement. Include the units of measurement in the axes labels. Show the name of the variable first and the unit of measurement immediately after or below the name.

Data source. Provide information about the source of the data. This information can be placed, in small type, below the X-axis label or in a caption. It can also be made available through online help.

Key or Legend

- Incorporate a key or legend for complicated graphs.

If a graph is complicated and contains a lot of data and a lot of visual coding, use a key or legend. Although a key or legend is not as easy to comprehend as integrated labeling, its use is better than cluttering up a graph with too much integrated labeling, or presenting illegible labels. A key or legend should contain the name of the data series and the symbols or colors used to show the displayed data.

https://hemanthrajhemu.github.io

Data Labels

- Permit data labels to be turned on or off.

Data labels contain the names or values for individual data points. If they are always on, they can easily clutter up the graph. If they are always off, the viewer must look back and forth between the legend and the graph. Permit the user to turn labels on or off as needed.

Aiding Interpretation of Numbers

- Display a grid on request.
- Permit the viewer to click on a data point to display actual values.
- Show numeric values automatically for each point or bar.
- Permit the viewer to zoom in on an area of the graphic.
- Permit the user to change the scale values.
- Permit toggling between a graphic and a table.

Computer graphics, unlike paper graphics, can be easily manipulated. Fowler and Stanwick (1995) suggest that the interpretation of numbers in graphical displays can be aided by permitting the aforementioned actions.

Types of Statistical Graphics

Statistical graphics take many forms. There are curve and line graphs, surface charts, scatterplots, bar charts, histograms, segmented or stacked bars, and pie charts.

Curve and Line Graphs

- Display data curves or lines that must be compared in a single graph.
- Display no more than four or five curves in a single graph.
- Identify each curve or line with an adjacent label whenever possible.
- If a legend must be included, order the legend to match the spatial ordering of the lines.
- For tightly packed curves or lines, provide data differentiation with a line-coding technique, such as different colors or different line composition types.
- Highlight curves or lines representing important or critical data.
- When comparing actual to projected data
  — Use solid curves or lines for actual data.
  — Use broken curves or lines for projected data.
- Display a reference index if the displayed data must be compared to a standard or critical value.
- Display differences between two data sets as a curve or line itself.

https://hemanthrajhemu.github.io

Curve and line graphs can be used to show relationships between sets of data defined by two continuous variables. They are especially useful showing data changes over time, being superior to other graphic methods for speed and accuracy in determining data trends. With a curve, a smoothed line summarizes the data relations. With a line, straight line segments connect the data plots. A line graph is illustrated in Figure 3.76. This kind of graph implies a continuous function. If the data point elements are discrete, it is better to use a bar graph.

Single graph. If several curves must be compared, display them in one combined graph to facilitate their comparison.

Four or five maximum. Display no more than four or five curves in a single graph. As more curves or lines are added to a graph, visually discriminating among them becomes more difficult. The maximum number of lines presented should be limited to four or five. If one particular curve or line must be compared to several others, consider multiple graphs where the line of interest is compared separately with each other line.

Label identification. Identify each curve or line with an adjacent label whenever possible. A label is preferable to a separate legend. If direct labeling is impossible due to the tightness of the lines, a legend may be the only alternative. If a legend is used, visually differentiate the lines (colors, line types, and so on), and include the coding scheme in the legend.

Legend. If a legend must be included, order the legend to match the spatial ordering of the lines. If legends are to be used on a series of graphs, however, maintain one consistent order for the legends on all graphs.

Tightly packed curves or lines. For tightly packed curves or lines, provide data differentiation through a line coding technique. Common coding techniques include different colors and line types. Do not exceed the maximum number of alternatives for the method selected, as shown in Table 11.1 in Step 11. If color-coding is used, choose colors on the basis of the considerations described in Step 12. Line width and dot size coding should be avoided because of their similarity to grids and scatterpoints. If a series of related graphs are line coded, be consistent in the selection of techniques for corresponding data.

Important or critical data. Highlight curves or lines representing important or critical data. If one curve or line in a multiple-line graph is of particular significance, highlight that curve (high intensity, different color, and so on) to call attention to it. The coding scheme selected should be different from that used for spatial differentiation. Use solid curves or lines for actual data; use broken curves or lines for projected data.

Comparing actual and projected data. When a curve or line must be compared to a standard or critical value, display a reference curve or line reflecting that value.

Data differences. If the difference between two sets of data must be determined, display the difference itself as a curve or line. This is preferable to requiring the user to visually compare the two values and calculate the difference between them. If the difference between the related curves is of interest, consider a band chart where both lines and curves are displayed and the area between them coded through use of a texture, shading, or color.

https://hemanthrajhemu.github.io

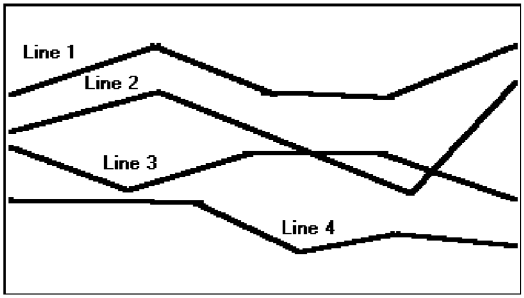


Figure 3.76: A line graph.

Surface Charts

- Order the data categories so that
  — The least variable is at the bottom, and the most variable at the top.
  — The largest is at the bottom and the smallest at the top.
- Use different texture or shading coding schemes to differentiate the areas below each curve or line.
- Incorporate labels within the bands of data.

If the data being depicted by a curve or line represents all the parts of a whole, consider developing a *surface* or *area chart*, as illustrated in Figure 3.77. In this kind of graph, the curves or lines are stacked above one another to indicate individual amounts or aggregated amounts. Each boundary height is determined by the height of the line below it, and the area between each line or curve is differently coded, usually by textures or shading. A surface chart is similar to a segmented bar chart.

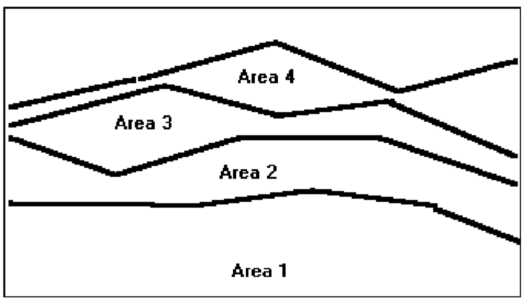


Figure 3.77: A surface chart.

https://hemanthrajhemu.github.io

Ordering. In ordering the data categories, place the least variable at the bottom and the most variable at the top. Irregularities in the bottom curve or line will affect those above it. This makes it difficult for a viewer to determine whether the irregularity in the upper curves reflect real data differences or is the result of this style of graph. Displaying the least variable data at the bottom will minimize this effect. Alternately, place the widest area at the bottom and the narrowest at the top. Wide bands look like they belong at the bottom, narrow at the top. If the data itself implies that some logical organization must be followed, and the resulting organization creates confusing distortions in the curves, this kind of graph should not be used.

Coding schemes. Use different texture or shading coding schemes. Ensure that the coding scheme chosen for each area is visually distinguishable from all the others. Place darker shades or colors toward the bottom.

Labels. Labels with a left-to-right reading orientation should be included within textured or shaded bands, if possible. Legends showing individual percentages, or cumulative percentages, should only be incorporated where space constraints exist within the bands.

Scatterplots

- Limit use to two-dimensional displays of data.
- Maintain consistent scale size intervals.
- Provide distinguishable, equal-sized plot points.
- If there is more than one set of data on the plot, use different symbols for each data set's points.
- Visually distinguish points of particular significance through a highlighting technique.

Scatterplots can be used to show relationships among individual data points in a two-dimensional array. A point is displayed on the plot where the X-axis and Y-axis variables intersect, as illustrated in Figure 3.78. Correlations and trends on scatterplots can be indicated by the superimposition of curves (thus combining the scatterplot with another kind of graphic display).

Two dimensions. Limit scatterplots to two dimensions. Three-dimensional scatterplots, while possible, do not yield clear, unambiguous displays.

Consistent intervals. Maintain consistent scale size intervals. Inconsistent spacing size between scale ticks on the two axes will distort the displayed data.

Distinguishable plots. Construct the plot points of distinguishable, equal-sized circles, squares, rectangles, or diamonds. These symbols may be filled in or empty. Color may also be used to designate the points. Keep in mind that, when using color, different colors can look different in size, and some people using the graphic may be colorblind.

https://hemanthrajhemu.github.io

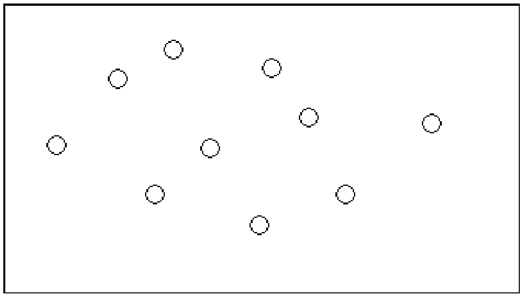


Figure 3.78: A scatterplot.

Multiple data sets. If there is more than one set of data on the plot, use different symbols for each data set's points. Choose distinguishable symbols from those described above.

Significant points. Visually distinguish significant points. Points of particular significance on scatterplots can be made distinctive through highlighting techniques such as the use of high intensity, different colors, or different shapes.

Bar Charts

- Orient bars consistently, either horizontally or vertically.
- Use vertical bars when the item being counted is of greatest interest.
- Use horizontal bars
  - — When the data labels are long.
  - — To highlight the information rather than the count.
- Use a meaningful organizing principle.
  - — If none exists, arrange the bars so that the length of bars is in ascending or descending order.
- Make the spacing between bars equal to one-half the width of the bars or less.
  - — If groupings of bars are presented, leave space between the groupings only.
- If different kinds of bars must be easily distinguished, provide differentiation through a coding technique.
  - — If possible, use a pattern or color that reinforces the data.
- Highlight bars representing important or critical data.
- Provide a consistent ordering for related groups of bars.
- Display a reference index if displayed data must be compared to a standard or critical value.
- Identify each bar with an adjacent label.
  - — Place labels below, or to the left of, the baseline.
- When a great many pieces of data must be compared, consider using histograms or step charts.

https://hemanthrajhemu.github.io

Bar charts can be used to show a few differences between separate entities or to show differences in a variable at a few discrete intervals. They are useful for comparing or ranking a small number of values — no more than 10 or 12. A bar chart consists of a series of bars extending from a common origin or baseline, as illustrated in Figure 3.79, or they may extend between separately plotted high and low points, as illustrated in Figure 3.80, having only one axis. Bar charts may be arrayed horizontally or vertically. Vertical bar charts are sometimes called column charts.

Consistent orientation. While bars may be oriented either horizontally or vertically, a consistent orientation should be maintained for bars displaying similar information. In general, frequency counts are best displayed in vertical bars. Also, use vertical bars when the values or count being displayed are of greatest interest. Use horizontally arrayed bars for time durations. Also use this orientation when the data labels are long and room is needed to present them, and when the information categories must be highlighted, rather than the count.

Meaningful organization. Use a meaningful organizing principle, such as volumes, dates, or alphabetical. If no meaningful principle exists, arrange the bars so that the length of bars are in ascending or descending order. If the information is being compared to a baseline or other comparative data, place the baseline bar to the far left or at the top.

Bar spacing. Space the bars for ease of visual comparison. Comparison of bars should be accomplishable without eye movement. Generally, the spacing between bars should be one-half or less of the bar width. If many bars are to be displayed, the alternating pattern of bright and dark bands that results can be visually disturbing to some viewers. In this case it is better to completely eliminate the spacing between bars. (The graph is then called a *histogram*.) If groupings of bars are presented, leave the space between the groupings.

Differentiation. If different kinds of bars must be easily distinguished, provide differentiation through a coding technique such as the use of color, texture, or shading. If possible, use a meaningful pattern or color that reinforces the differences.

Important or critical data. Highlight important or critical data. If one bar represents data of unusual significance, call attention to that bar through a different coding technique. Related groups of bars should be ordered in a consistent manner.

Related bar ordering. Provide a consistent ordering for related groups of bars.

Reference index. When bars must be compared to some standard or critical value, display a reference line to aid that comparison.

Labeling. A label associated with each bar, in left-to-right reading orientation, is preferable to a separate legend. Place labels below, or to the left of, the baseline. If the labels on a horizontal bar chart are short, left-align them. If they are long, right-align them to the axis. If groups of bars are repeated, it is only necessary to label one group rather than all bars in all groups.

Histograms or step charts. When a great many pieces of data must be compared, consider histograms or step charts. These are bar charts without spaces between each of the bars, as illustrated in Figure 3.81. The area of a bar in a histogram reflects the amount of the value; so all bars should be of equal width.

https://hemanthrajhemu.github.io

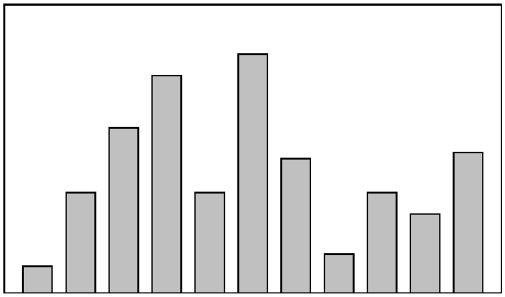


Figure 3.79: A bar chart with a common origin point.

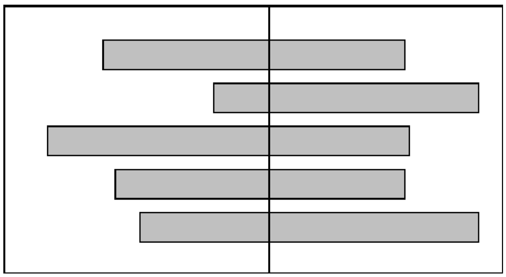


Figure 3.80: A bar chart with separately plotted high and low points.

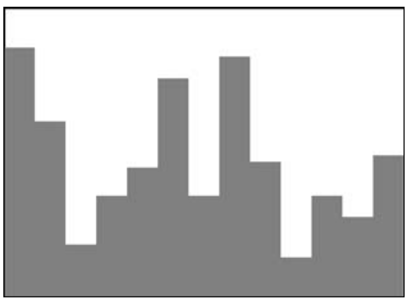


Figure 3.81: A histogram or step chart.

Segmented or Stacked Bars

- Order the data categories in the same sequence.
- Order the data categories so that
  — The least variable is at the bottom.
  — The most variable is at the top.
- Limit the number of segments to those that are large enough to be seen and labeled.
- Use different texture or coding schemes to differentiate the areas within each bar.
- Clearly associate labels with bars or segments.
  — Place segment labels to the right on a vertical chart or above on a horizontal chart.

https://hemanthrajhemu.github.io

If both the total measure of a value and its component portions are of interest, consider using *segmented* or *stacked bars*. These bars are similar to bar charts except that the bar is segmented into two or more pieces reflecting the component values, as illustrated in Figure 3.82. In this way they are similar to surface graphs and pie charts. Design guidelines are similar to stacked bars, except for the following:

Data category ordering. To provide consistency, order the data categories in the same sequence for all bars. Order data categories to show least variable at bottom and most variable at top. Irregularities in the bottom segment will affect those above it. This can make it difficult for a person to determine whether the irregularity in the upper segments reflects real data differences or is the result of this style of graph. Displaying least variable data at the bottom will minimize this effect. Also consider displaying the least variable values at the bottom, as is done with surface charts, unless the data itself dictates that some other logical organization must be followed.

Large segments. Limit the number of segments to those that are big enough to be seen and labeled. If small segment components exist, group them into an "other" category.

Coding schemes. Use different texture or shading coding schemes. Ensure that the coding scheme chosen for each segment is visually distinguishable from all others. Place darker shades or colors toward the bottom or toward the left.

Labeling. Associate labels with bars and segments. Labels, with a left-to-right reading orientation, are preferable to legends. Do not place labels within segments, as they most often will not fit. Legends should only be used if space does not allow labels.

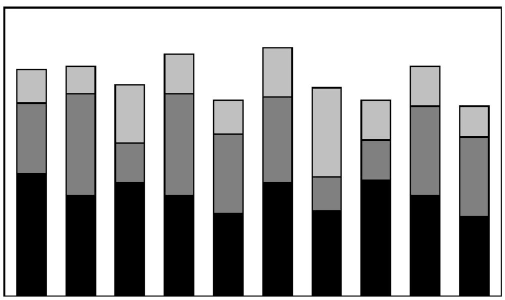


Figure 3.82: A segmented, or stacked, bar graph.

https://hemanthrajhemu.github.io

Pie Charts

- Pie charts should be used with caution.
- If pie charts are used,
  - — They must add up to 100 percent.
  - — Use five segments or fewer.
  - — Each segment should take up at least 5 percent (18 degrees) of the circle.
  - — Place the largest segment starting at 12:00.
  - — Directly label each segment in the normal reading orientation.
    - If leaders for labels in small segments are necessary, orient them in as few angles as possible.
  - — Include numbers with segment labels to indicate percentages of absolute values.
  - — Texture- or color-coding selected for segments should not emphasize one segment over another (unless it is intended).
  - — Highlight segments requiring particular emphasis through a contrasting display technique or by "exploding" it.
  - — Never tilt a pie.

Pie charts, a circle broken up into pie-shaped pieces, can be used to show an apportionment of a total into its component parts, as illustrated in Figure 3.83. Bar graphs, however, usually permit more accurate estimates of proportions. Experts caution against the use of pie charts because

- They provide no means of absolute measurement.
- They cannot represent totals greater than 100 percent.
- They can only represent a fixed point in time.
- Human estimation of relationships is more accurate with linear than with angular representations.

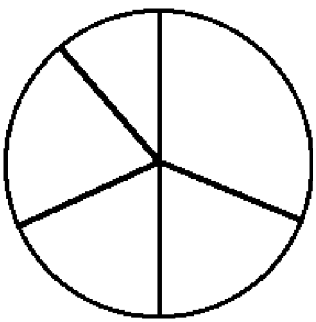


Figure 3.83: A pie chart.

https://hemanthrajhemu.github.io

If pie charts are used, the guidelines below should be followed.

Total 100 percent. The parts must add up to 100 percent. To convert from percentages to degrees, multiply the percentage by 3.6.

Five segments or fewer. To minimize confusion, restrict pies to five segments or fewer. This permits adequate differentiation of its pieces and accurate labeling.

Minimum five percent. Avoid very small segments. Segments should take up at least 5 percent (18 degrees) of the circle in order to provide adequate segment differentiation. If small portions exist, combine the pieces into an "other" category and list them in a caption or note.

Start at 12:00. Start with the largest wedge at 12:00 (or a quarter hour) and order from largest to smallest in a clockwise order.

Labeling. To provide maximum association of the label with data, and for reading clarity, use a left-to-right reading orientation. If it is impossible to include the label within the segment, it may be placed outside and tied to the segment with a leader line. Place the labels in one or two columns (one on each side). If multiple outside labels and leader lines are necessary, orient the lines in as few angles as possible.

Numbers with segment labels. Include numbers with the segment labels to indicate percentages or absolute values. Only by including numbers with segment labels can numeric values be accurately established. Alternately, make actual percentages available when requested.

Segment coding. The kinds of textures or colors selected for segments should not emphasize one segment over another, unless this emphasis is intended.

Highlighting. Highlight segments requiring emphasis. Use a contrasting display technique or *explode* segments requiring emphasis. Exploding is accomplished by slightly displacing a segment from the remainder of the pie.

Tilting. Never tilt a pie. Distortion will occur with tilting. Small wedges at the front will look larger than they actually are.

Choosing a Graph Type

- Determine what kind of information is most important for the viewer to extract.
- Choose the type of graph best suited for presenting that kind of information.

The types of graphics just described have rarely been experimentally studied to determine their most effective use. Some studies addressing this issue, however, are those of Hollands and Spence (1992) and Simkin and Hastie (1987). These researchers collected data on three tasks: (1) determining a proportion of a whole where the proportion *was* a part of the whole (proportion), (2) determining a proportion of a whole where the proportion *was not* a part of the whole (comparison), and (3) determining a change over time (change). The results are summarized in Table 3.4.

https://hemanthrajhemu.github.io

Table 3.4: Tasks and Best Types of Graphs

	PROPORTION			
	WITH SCALE	WITHOUT SCALE	COMPARISON	CHANGE
Best	Line Graphs	Segmented Bars	Bar Graphs	Line Graphs
	Bar Graphs	Pie Charts	Segmented Bars	Bar Graphs
	Segmented Bars			
	Pie Charts			
Poorest	—	Bar Graphs	Pie Charts	Segmented Bars
	Line Graphs	Pie Charts		

Source: Hollands and Spence (1992) and Simkin and Hastie (1987).

In estimating proportion, if a scale is not included on the graph, pie charts and segmented bars were found to be best. If a scale is included, line graphs and bar graphs are also usable, both actually having a slight edge in speed over pie charts and bar graphs. In the comparison task, bar graphs and segmented bars were superior to pie charts. In estimating change over time, line graphs and bar graphs were both very effective, and pie charts and segmented bars the poorest.

In choosing a graph to display information, the kind of information important to the viewer must always be determined first. This will point to the kind of graphic most effective for the task.

Flow Charts

- Displayed steps should be designed to
  — Follow some logical order.
  — Minimize path link.
- Orient the chart following common flowchart reading conventions such as left-to-right and top-to-bottom.
- Follow common flowchart coding conventions to distinguish elements.
- Use arrows in conventional ways to indicate directional relationships.
- Highlight elements requiring particular attention through a contrasting display technique.
- Require only one decision at each step.
- Be consistent in all option ordering and wording.

If the data to be displayed flows in a complex, yet sequential, process, consider using a *flowchart* to schematically represent it. Flowcharts can also be used to aid problem solving in which a solution can be reached by answering a series of questions. They are not useful when trade-offs must be made.

https://hemanthrajhemu.github.io

Order of steps. One logical ordering scheme is to follow a sequence of operations or processes from start to finish. Other potential ordering schemes include placing the most important decisions first or the decisions that can be made with the most certainty. If no logical order is apparent, order the flowchart to minimize the length of the path through it. If some decision paths are more likely to occur than others, minimize the length of the most likely path.

Orientation. Follow a left-to-right and top-to-bottom orientation.

Coding conventions. Follow existing shape coding conventions for the kinds of boxes being displayed. Adhere to standards and people's expectations.

Arrows. Use arrows to indicate directional relations and sequential links.

Highlighting. Contrasting display techniques, such as high intensity or color, should be used to call attention to relevant paths or elements. Color is particularly effective in this regard.

Only one decision at each step. Multiple decisions reduce flowchart size. However, requiring multiple decisions such as "Is A true and B false?" can be confusing. Require that only single decisions be made.

Consistency. Consistently order and word all choices. Consistency always aids learning.

Technological Considerations in Interface Design

Interface design is also affected, and constrained by, characteristics of the hardware being used and the interface's controlling software.

Graphical Systems

- Screen design must be compatible with the capabilities of the system, including
  — System power.
  — Screen size.
  — Screen resolution.
  — Display colors.
  — Other display features.
- Screen design must be compatible with the capabilities of the
  — System platform being used.
  — Development and implementation tools being used.
  — Platform style guide being used.

Graphical system design must be compatible with the system's power, screen size, screen resolution, and displayable colors, fonts and other features. Designs for Web systems must also take into consideration the characteristics of the browsers being used and the bandwidth of the communication medium. The design must also be compatible

https://hemanthrajhemu.github.io

with the system platform and any development and implementation tools being used. The design must also take into consideration any available platform style guide.

System Power

A slow processing speed and small memory may inhibit effective use of a system. Feedback and animation capabilities may be limited, reducing the system's usability. Slow responses can be error prone, grossly inefficient, and very aggravating. A slow screen refresh rate will increase the user's chances of perceiving screen flicker, which can be visually fatiguing. A system must be powerful enough to perform all necessary actions promptly, responsively, and meaningfully.

Screen Size

Through the years, the physical size of an available monitor's screen area has been gradually increasing. Current typical monitor sizes range from 13 to 21 inches (measured diagonally), with 17 inches being most common.

Many of today's screens are not large enough in size to take full advantage of windowing capabilities. As a result, many windows are still of Post-It dimensions. There is some evidence, from studies and personal observation, that many users of windowing systems expand their windows to cover a full screen. Either seeing all the contents of one window is preferable to seeing small parts of many windows, the operational complexity of multiple windows is not wanted, or visual noise is being eliminated. Whatever the case, these actions indicate a shortcoming in windowing systems, as they exist today.

Best monitor size was the focus of a study by Simmons and Manaham (1999). They compared monitors of 15, 17, 19, and 21 inches for search activities using Microsoft's Word and Excel, and for browsing the Web. The 21-inch monitor resulted in fastest task completion. Users, however, preferred using the 19-inch monitor. DiPierro et al., (2000) compared Web navigation performance using small, medium, and large screens. No performance difference was found between small and medium screens, but the large screen elicited a 26 percent faster performance.

Clearly, these studies point out that bigger screens may be better. One typical-sized screen today hardly approximates one standard-sized piece of paper. Comparisons between information contained on separate pieces of paper are still difficult using current monitor sizes. The answer to this problem may lie in providing multiple monitors for doing one's work or performing other activities. Multiple monitors reflect more closely the "cluttered desk" metaphor, for better or worse.

Continually expanding monitor and screen size as a solution, though, may create other problems. A large display area will require longer control movements to reach all locations on the screen, and more head and eye movements. Even with today's relatively small screens, many activities in the peripheries of vision still go unnoticed. Larger screens will compound this problem. The effect on user's physical comfort, and the possibility of needing an expanded working area must be considered. Clearly, the screen size/multiple monitor usability trade-off must be studied further.

https://hemanthrajhemu.github.io

Screen Resolution

Screen resolution is the horizontal and vertical height of a screen in pixels. It is a function of the monitor's capabilities and its video card. Most common display resolutions currently are 800×600 (pixels width and height), 1024×768, and 1280×1024. Higher resolutions are also available. Poor screen resolution may deter effective use of a graphical system by not permitting sharp and realistic drawings and shapes. Window structure and icon design may be severely affected.

In a study mentioned previously, Ziefle (1998) evaluated reading performance, using both hard-copy paper documents and monitors at different resolutions. In the first study, she compared paper printed at 255 dots per inch (dpi) and monitors whose resolutions were 832×600 pixels (60 dpi) and 1664× 1200 pixels (120 dpi). A 19-inch monitor showing black characters on a light background was used, and reading speeds and proofreading accuracy were compared. Ziefle found no difference in performance between the monitors with different resolutions. In the second Ziefle study, participants performed a continuous visual search task. Performance was compared using screens of lower resolution, 720× 540 (62 dpi), and higher, 1024×768 (89 dpi). She found that the higher resolution screen was searched significantly faster. Using the lower resolution, she found that after 30 minutes the participants began to search more slowly, made more errors, and had more and longer eye fixations.

In conclusion, looking just at the differences between the monitors of different resolutions, both studies found advantages for the higher resolution screens in terms of both preference and performance. Adequate screen resolution, then, is a necessity to achieve meaningful representations. Higher resolution screens also appear to be better for one's eyes.

Colors

The color palette must be of a variety large enough to permit establishment of a family of discriminable colors. The colors used must be accurately and clearly presented in all situations. The contextual effect of colors must also be considered, because hues may change based on factors such as size and one color's location in relation to other colors. Color is thoroughly discussed in Step 12.

Other Display Features

A wide range screen attributes or properties are available to aid the screen design process. Included are such techniques as higher brightness, reverse polarity, different font sizes and styles, underlining, blinking, line rules and boxes, color, and white space. Before beginning design, the designer must be aware of what capabilities exist, how they may be most effectively used, and what their limitations are. All of these techniques are described in other sections in this text.

The design must be compatible with the system platform and any development and implementation tools being used. The design may also take into consideration any available platform style guide. Finally, the design must effectively utilize the various available display features or attributes.

https://hemanthrajhemu.github.io

Platform Compatibility

The design must be compatible with the windowing platform being used — Apple Computer's Macintosh, Microsoft Windows, or any other.

Development and Implementation Tool Compatibility

More that half of software code is now devoted to user interface design. To use a very old cliché, the tail is now beginning to wag the dog. Available tools include toolkits, interface builders, and user interface management systems.

A *toolkit* is a library of controls or widgets such as menus, buttons, and scroll bars. Toolkits have a programmatic interface and must be used by programmers. They are usually for a specific windowing platform. Examples of toolkits include those for Motif, OpenLook, and the Macintosh.

An *interface builder* is a graphical tool that helps a programmer create dialog boxes, menus, and other controls. It provides a palette to select and position controls, and to set properties. Interface builders are limited to use in laying out the static parts of the interface. They cannot handle the parts of the interface that involve graphical objects moving around. A *user interface management system* (UIMS) extends the features of a builder by also providing assistance with creating and managing the insides of windows. Examples include HyperCard and Visual Basic.

MAXIM **Software should be seen and not heard.**

Style Guide Compatibility

A thrust for commonality in graphical system application design has emerged as providers have finally come to realize that design consistency is a virtue that has been ignored too long. To achieve this consistency in interface design, most providers have developed style guidelines for system developers. These guidelines specify the appearance and behavior of the user interface. They describe the windows, menus, and various controls available, including what they look like and how they work. They also provide some guidance on when to use the various components.

Examples of industry-produced guidelines include Apple's *Macintosh Human Interface Guidelines*, IBM's *System Application Architecture Common User Access (SAA CUA)* and Microsoft's *The Windows Interface Guidelines for Software Design*. Product style guides vary in their ability to control compliance with the guidelines they present. Some present strict requirements, leading to excellent consistency across applications; others provide little guideline compliance control. The design should comply with the relevant platform style guide.

Web Systems

- Understand the current level of Web technology.
- Design for system configuration used by most users.
- Refrain from haphazard use of leading-edge technology.

https://hemanthrajhemu.github.io

The Web is truly a Web, a Web of users whose only consistency is inconsistency in the variety of the technologies they possess. Old PCs with few features must coexist with new PCs possessing the latest technological advances. Monitors with small screens must coexist with large screens. High-resolution displays must coexist with those of lower resolution. High-speed information transmission must coexist with low speed. New browsers that contain and support many different and desirable features must coexist with old browsers that support less. To make matters worse for the designer, users can reconfigure their own PCs, further changing some of their characteristics.

The designer must be capable of handling these various demands while creating usable Web pages accessible through different browsers, operating systems, and computer platforms. To do this requires having an awareness of system configurations that satisfy the needs of the majority of users, and then designing for these users. To utilize the Web's richest features, however, the designer must understand the current level of technology and apply it in a meaningful and usable way, especially for those users at the high end of the technological spectrum. The temptation, though, to apply technology simply for technology's sake must be resisted. The goal in design is to satisfy the user's need or want, not the designer's. The following sections address technological considerations affecting Web site design.

Browsers

- Compatibility:
  - — Design for the most common browsers.
  - — Account for browser differences.
- Operating systems:
  - — Design for popular operating systems.
- Connection speed:
  - — Design for the most commonly used connection speeds.
- Monitor size and resolution:
  - — Design within the boundaries of an image-safe area for all browsers.
  - — Design for commonly used screen resolutions.
- Fonts:
  - — Use fonts that can be displayed on a variety of browsers.
- Colors:
  - — Use colors that succeed on a variety of browsers and platforms.
    - Use a palette of 216 colors.
- Versions:
  - — Create multiple versions that support multiple browsers.
    - Always provide a text-only version.
    - Make use of browser sniffers.

The pressure for lowest-common-denominator design is often outweighed by the designer's desire to create larger displays and employ the latest display and browser features. The needs of all users must be considered in design. A year 2006 user technology profile is shown in the next section.

https://hemanthrajhemu.github.io

Compatibility. The entire Web page content should be accessible from the browsers of at least 95 percent of all users, presenting content as consistently and predictable as possible. In general, use browser defaults as much as possible, designing for what everyone can see. Do not assume, however, that all users will have the same browser features, and will have set the same defaults. Some people may use larger fonts, override fonts, or use fewer colors. So, never assume that the designed page will look exactly the same to users as it does to the designer. Test the design on all browsers and all versions of the each. Specify on the Web site what assumptions were made about the browser settings.

Operating Systems. Provide a design that will work well with the most common operating systems. Test the design on all popular operating systems.

Connection speed. Design for the connection speeds used by most users. Currently, the typical Web home user is dialing in at 56 Kbps (or slower) through a regular telephone line. High-speed access is increasing in popularity for home users, however. The typical business user possesses a high-speed connection to the Web. At the other end of the connection speed scale are users with low-cost devices, users wanting low-bandwidth wireless access, users with small personal display devices, and users in developing countries with poor communication infrastructures. All must be supported in Web page design. Failure to consider these modem speed limitations will lead to long download times and can result in user frustration. The biggest influence on download speed is the number and size of graphics on Web pages.

Monitor Size and Resolution. Designed page content should always be restricted to the boundaries of an "image-safe" area horizontally, and perhaps vertically, depending upon whether vertical scrolling is determined as necessary to see the page's entire content. Exceeding the horizontal safe area will require horizontal scrolling to see the page's entire width. Because some information will not always be visible, content usability and interpretation will be severely degraded, and the user inconvenienced. Exceeding the safe area length will require vertical scrolling to see the entire page, and could push important information out of the user's view. A display's safe area will be dependent upon the monitor's size and the resolution at which it is set. It must be established based upon the anticipated or known range of monitor sizes and their set resolutions. Current typical monitor sizes range from 13 to 21 inches (measured diagonally) with 17 inches being most common.

Today, about 90 percent of users have their screen resolution set to 800×600, 1024×768, or 1280×1024 pixels. Designing for these settings will accommodate the majority of users.

Fonts. Not all browsers provide the same typographic operations. Different default font types and sizes may exist, depending on the type of browser, browser version, and operating system the browser runs on. If a page is designed using a font the user does not have installed, the browser displays its default fonts. Many older browsers support only two fonts, Times New Roman and Courier. Newer browsers support more fonts but they must be installed on the machine doing the browsing. Default fonts may include Times New Roman, Arial, Helvetica, and Verdana. Type displayed on a Windows browser may look 2 to 3 points larger than that on a Macintosh.

https://hemanthrajhemu.github.io

It is best to stick with the default fonts installed on the user's computers. All available options must be known before beginning the design process. Then ensure that the text is readable and legible in all usage environments.

Color. The color palette must be of a variety large enough to permit establishment of a family of discriminable colors. The colors used must be accurately and clearly presented in all situations, but be aware that colors may appear slightly differently on different monitors, and all users may not default their palettes to high color settings. Use colors that will succeed on a variety of platforms and monitors. Design using a browser-safe, cross-platform palette of 216 colors. It is sometimes referred to as a "Web safe" color palette.

Versions. To provide universal access to a Web site, provide multiple versions that support multiple browsers. To limit the site to one browser may deny access to, and alienate, users who do not have the proper one. Make use of browser "sniffers," programs on the server that detect the user's browser type and determine which version should then be downloaded.

Always provide a text-only version of the Web site. This will be necessary as long as users with small displays and low bandwidths exist. Vision-impaired users with readers will also require a text-only version, as will users with text-only browsers, and those who turn off image display.

Other Web Considerations

- Downloading:
  - — Provide fast page download times, no more than 8 to 10 seconds per page.
    - Minimize the use of design techniques that cause longer download times.
    - Long pages.
    - Large chunky headings.
    - Numerous or large graphics and images.
    - Animation.
    - Excessive amount of color.
    - Excess use of frames.
  - — Provide enough information to the user so that whether or not to request a download can be determined, including
    - Program or document description.
    - Type of download.
    - Size of download.
    - Download version.
    - Estimated loading time.
    - Special operating requirements.
- Currency:
  - — Keep Web site information current.

https://hemanthrajhemu.github.io

- Page printing:
  — Provide a means to print
    - Groups of related pages.
    - Individual pages.
    - Sections of pages.
- Maintainability:
  — Ensure easy Web site maintainability.

Downloading. Slow download speeds are an ongoing complaint of Web users. Download times of 8 to 10 seconds per page should not be exceeded, even for bandwidths of 28.8 Kbps. In general, keep graphics and page size as small as possible. Specifically, use text instead of graphics whenever possible. When graphics are desirable, keep the graphic as small as possible. Also, repeatedly use a graphic so it may be stored in the browser's cache.

The cache is a temporary storage area for Web pages and images. There are two types: memory and hard drive. Once a graphic is downloaded, it is placed in the cache and remains there for a prescribed period of time. Download times, then, are longest when a site is visited and downloaded for the first time. After the first download, the graphic is in the cache. It is retrieved from there, reducing the time it takes to display the page.

In addition to graphics, other design elements that slow download times include large and chunky headings, the use of many colors, animation, excessive use of frames, and the use of Java and JavaScript. Limit the use of these elements that require a long time to download. Create graphics that load quickly. Limit an individual image to 5KB, the images on a page to 20KB. Use Graphics Interchange Format (GIF) files because they are smaller than Joint Photographic Experts Group (JPEG) files. GIFs and JPEGs are described in more detail in Step 11.

Also, because few monitors display images at greater than 72 dpi, restrict the resolution of downloaded images to 72 dpi. Using a higher dpi ratio will not produce a better image, but will increase file size, causing longer download times. All images should also contain alternate text (alt text). This will give the user something to read while the image downloads, and is necessary for text-only viewers.

Always provide enough information to let the users know whether it is worth their time and trouble to download something of apparent interest described on a Web page. Provide a program or document description, type, size, version, estimated loading time, and any special operating requirements, including such things as hardware needed, the required operating system, special software needed, and memory requirements.

Currency. Update the Web site regularly to keep information current. The nature of the Web implies timeliness. Outdated information casts doubts on a Web site's credibility. Currency means trustworthiness to many users.

https://hemanthrajhemu.github.io

Page Printing. Some people prefer to read hard copy, especially anything longer than half a page. Make printing easy for users, including the capability to print sections, pages, or groups of related pages with minimal effort. Since most low-end printers print at 300 dpi, pages may be printed at this resolution. This higher resolution will result in a longer printing time, however.

Maintainability. Provide easy Web site maintainability to sustain its currency. Change must be easily accommodated as the Web site grows, evolves, and matures. Web site maintenance is, in reality, Web site enhancement. Remove outdated information and expired links, link old pages to those newly created. Properly designed, modular system pages covering specific topics can be updated quickly without needing to change and reformat large amounts of information.

The User Technology Profile Circa 2006

While a great variety does exist in the technological tools people use to interact with the Internet, the dominance of certain operating systems and browsers is quite evident. Following is a summarization of several characteristics as of December 2006 (from `www.thecounter.com/stats`). Statistics from December 2001 are included for comparison purposes.

BROWSERS

December 2006		December 2001	
Internet Explorer 6.x	70%	Internet Explorer 5	68%
Internet Explorer 7.x	12%	Internet Explorer 6	19%
Firefox	11%	Internet Explorer 4	5%
Safari	3%	Netscape 4.x	4%
Internet Explorer 5.x	1%	Other	4%
Opera	1%		
Unknown/Other	2%		

OPERATING SYSTEMS

Windows XP	82%	Windows 98	68%
Windows 2000	7%	Windows 2000	15%
Mac	4%	Windows 95	6%
Windows 98	3%	Windows NT	4%
Unknown/Other	4%	Mac	1%
		Unknown/Other	6%

https://hemanthrajhemu.github.io

SCREEN RESOLUTION IN PIXELS

December 2006		December 2001	
1024×768	54%	800×600	53%
1280×1024	21%	1024×768	33%
800×600	14%	640×480	4%
1152×864	3%	1280×1024	3%
Unknown/Other	5%	1152×864	2%
		Unknown/Other	5%

COLOR IN BITS

32	84%	16	54%
16	11%	32	30%
24	2%	24	10%
Unknown/Other	3%	8	4%
		Unknown/Other	2%

Assuming that these four elements are independent, the probability of a person having on his or her computer elements found in each listing above is .80. This means that designing for the above technological characteristics will satisfy the needs of four out of five Web users. Bailey (2001) suggests that when it is impossible to design for all users, because of cost, schedule, or personnel considerations, designing for the above computer characteristics will at least satisfy the needs of most users.

Examples of Screens

What follows are examples of poor and proper design. The problems of the poorly designed screens will be described in the discussion to follow. Redesigned versions of these poorly designed screens will be presented at the end of Step 8. Examples of other, properly designed, screens will also be presented as models of good design.

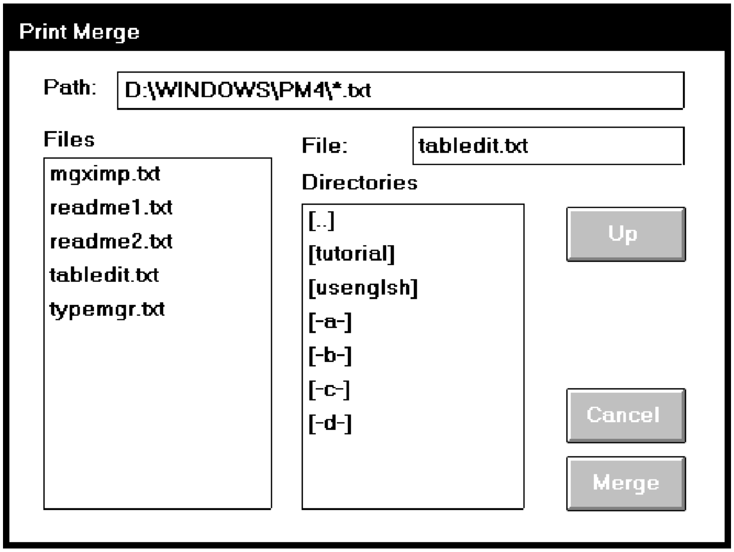
Example 1

Here are three information entry/modification dialog boxes from a popular drawing program, PRINT MERGE, PAGE SETUP, and EXPORT. Analyze them for problems, including inconsistencies between them.

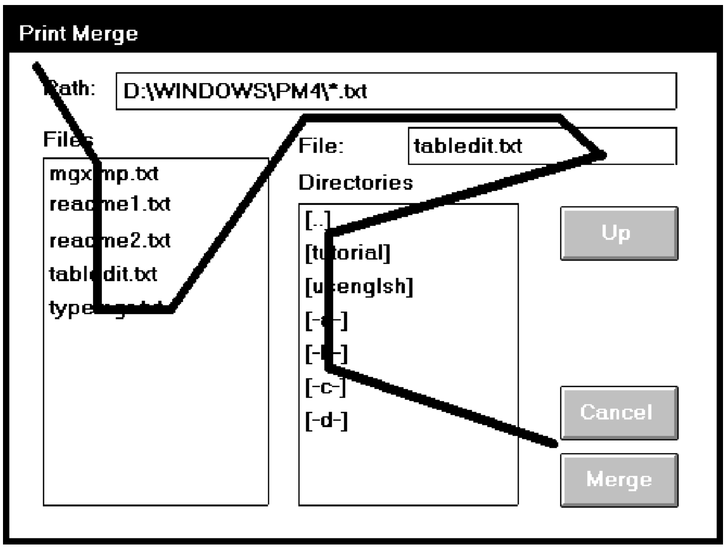
Screen 1.1

The controls on the PRINT MERGE screen are very poorly aligned. The File text box is located quite far from its associated list box. What does the Up button do? It is actually related to the Directories list box. This is certainly not clear. Look at the required sequence of eye movements through this screen, as illustrated by the line drawn between successive controls. This is very inefficient.

https://hemanthrajhemu.github.io

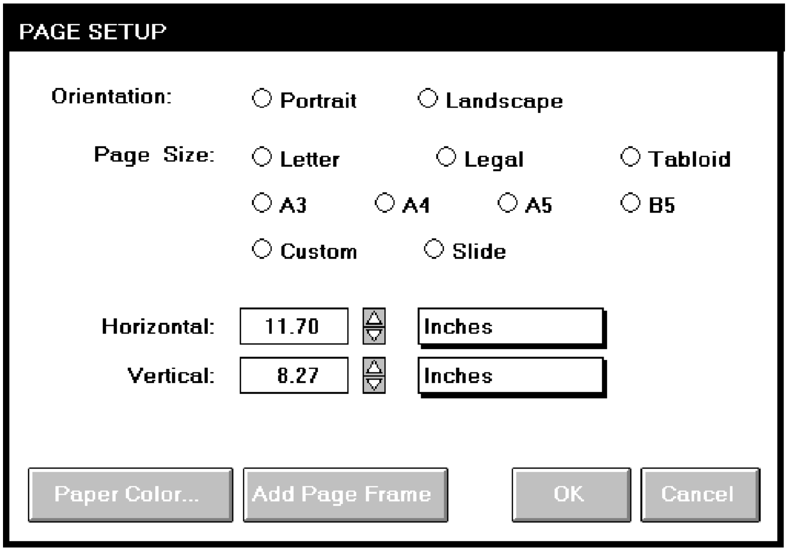


Screen 1.1A

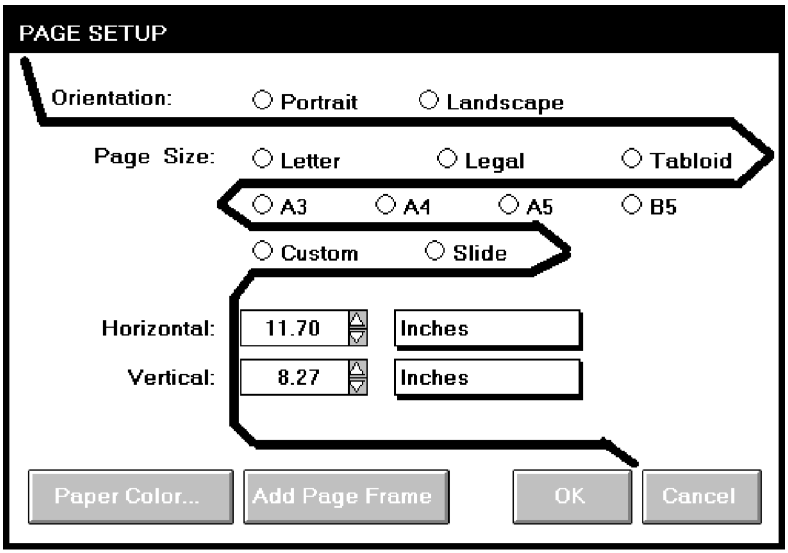


Screen 1.1B

https://hemanthrajhemu.github.io



Screen 1.2A



Screen 1.2B

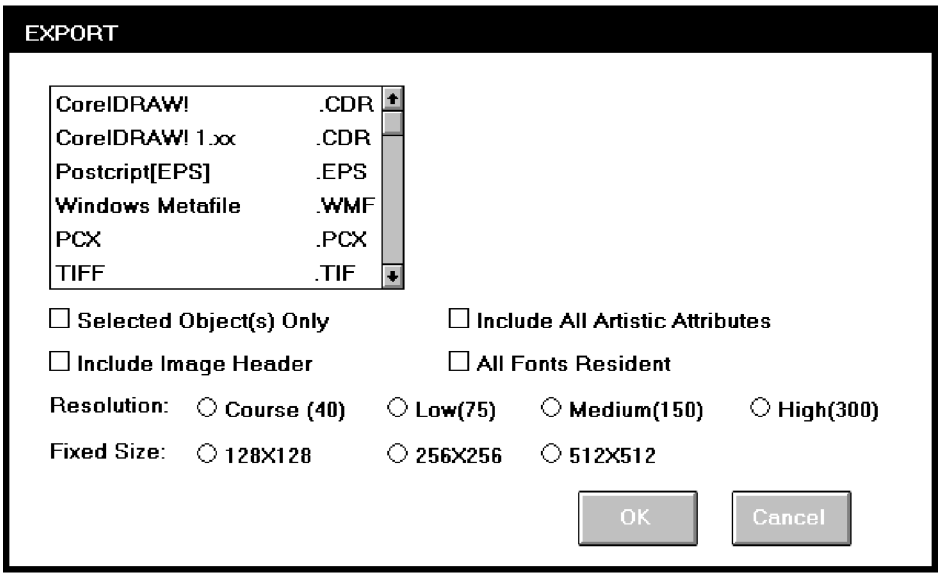
https://hemanthrajhemu.github.io

Screen 1.2

The controls on the PAGE SETUP screen are very poorly grouped. Are the nine radio buttons beginning at the Page Size caption one or three groupings? The horizontal orientation of the radio buttons necessitates a less efficient horizontal scanning and makes visual comparison of the alternatives more difficult. Why is the Orientation caption not right-aligned with the other captions? What are the controls inscribed with Inches? They are actually nonstandard controls but this is not clear until one discovers how to operate them (clicking on the rectangular bar changes the value [inches] now displayed). Nonstandard controls increase learning requirements and add to the complexity of the interface. Again, look at the required sequence of eye movements through this screen, as illustrated by the line drawn between successive controls. This is very inefficient.

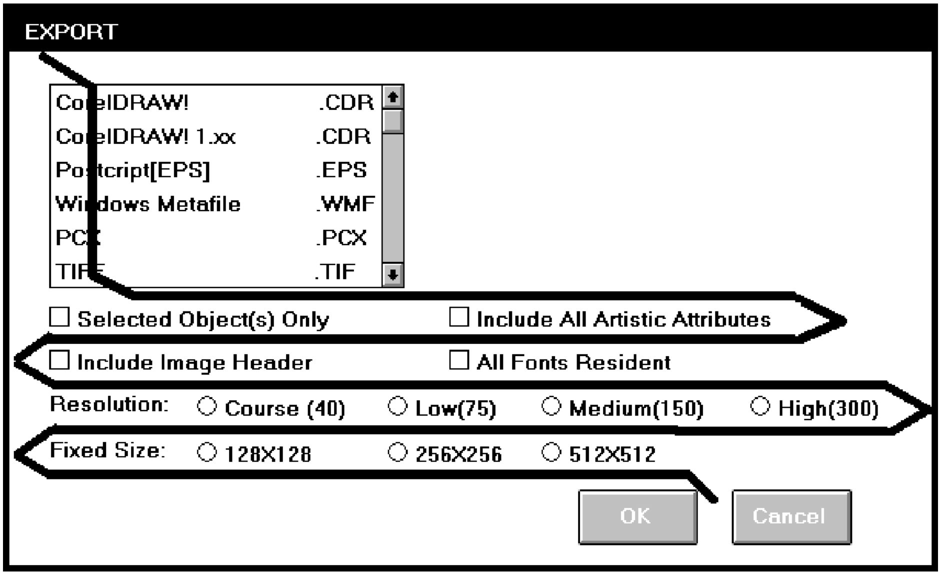
Screen 1.3

The check boxes and radio buttons on the EXPORT screen are again very poorly grouped. Their horizontal orientation necessitates a less efficient horizontal scanning and makes visual comparison of the alternatives more difficult. Can the check boxes be grouped? The list box has no caption with it. Screen balance is also poor, with the large open area in the upper-right part of the screen. Again, look at the required eye scan through this screen.



Screen 1.3A

https://hemanthrajhemu.github.io



Screen 1.3B

Now, look at the inconsistencies between these three screens. The Print Merge title is in mixed case; PAGE SETUP and EXPORT are capitalized. The Print Merge command buttons are on the right side; the PAGE SETUP and EXPORT ones are on the bottom. Print Merge uses the action being accomplished as the button label (Merge); the others use the standard OK. This will certainly cause user confusion. Print Merge and EXPORT appear to use left-aligned captions, PAGE SETUP right-alignment.

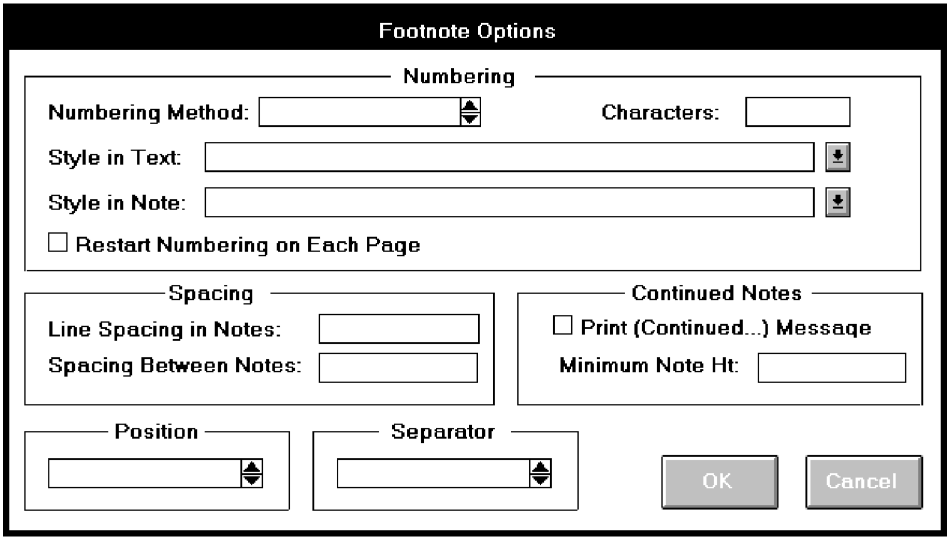
Example 2

Here are two information entry/modification dialog boxes from a popular word-processing program, Footnote Options and Location of Files. Some design enhancements are immediately obvious. The command buttons are consistently positioned in the lower-right corner and group boxes or borders are included to visually strengthen information groupings.

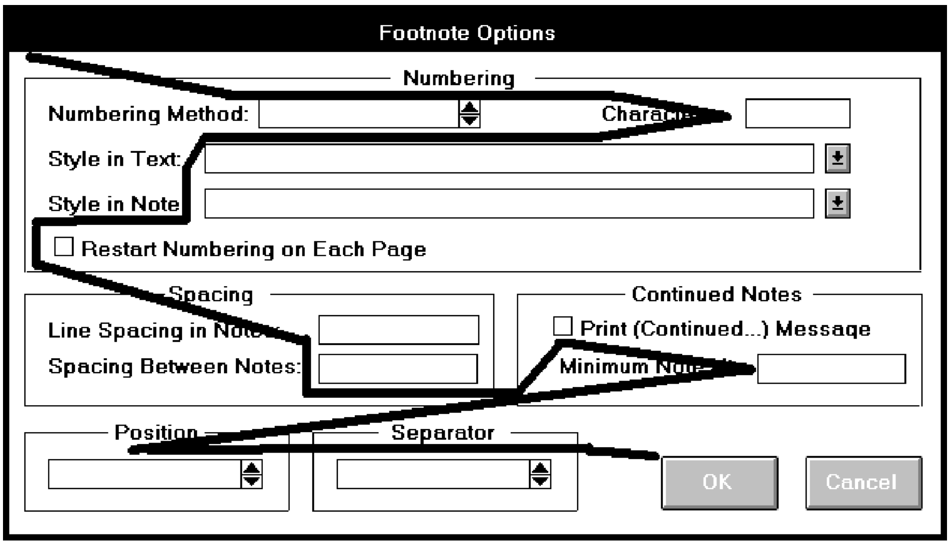
Screen 2.1

This screen's main problem is poor alignment of controls. The standalone check boxes tend to get lost. The centered headings in the group boxes are not a severe problem, but they do somewhat compete with the control captions for the screen viewer's attention. The Position and Separator control borders are placed around single elements. This is not recommended, but it does help to provide screen balance, which is quite good. The sequentiality of this screen, as illustrated by one's required eye scan, is quite poor, as illustrated.

https://hemanthrajhemu.github.io

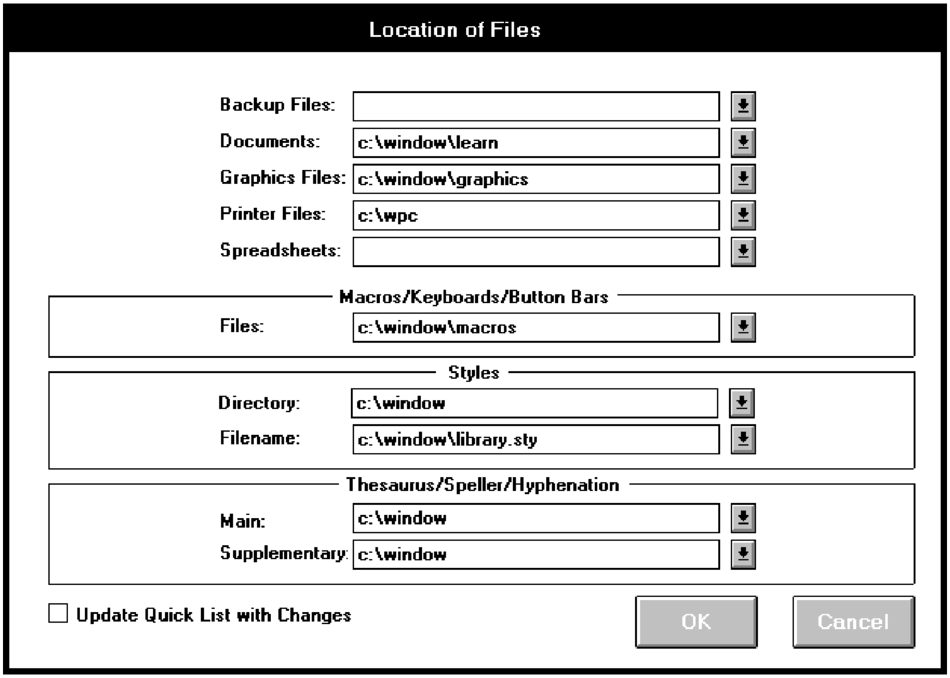


Screen 2.1A

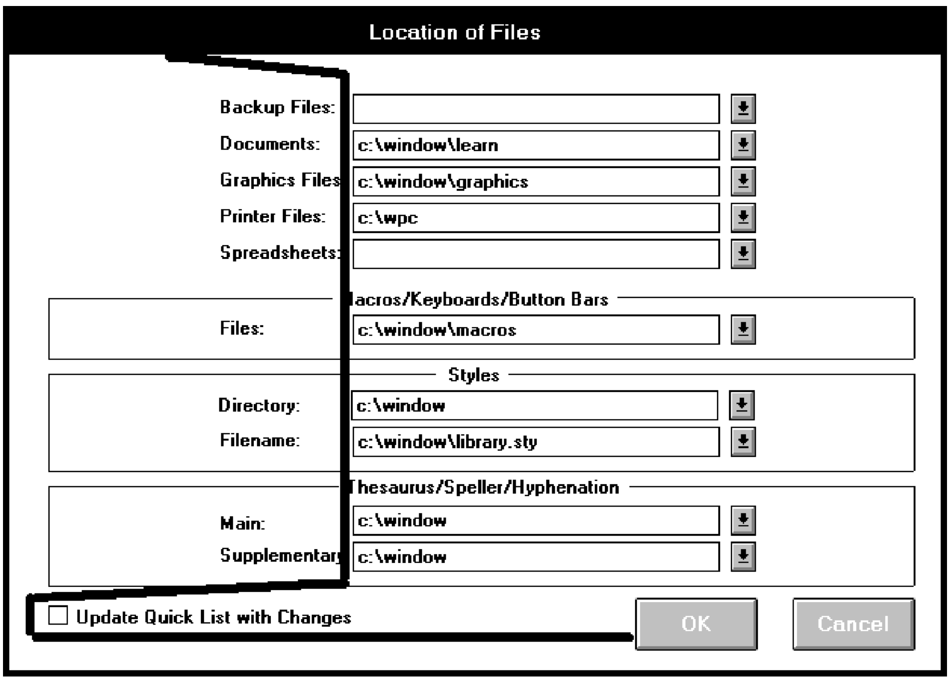


Screen 2.1B

https://hemanthrajhemu.github.io



Screen 2.2A



Screen 2.2B

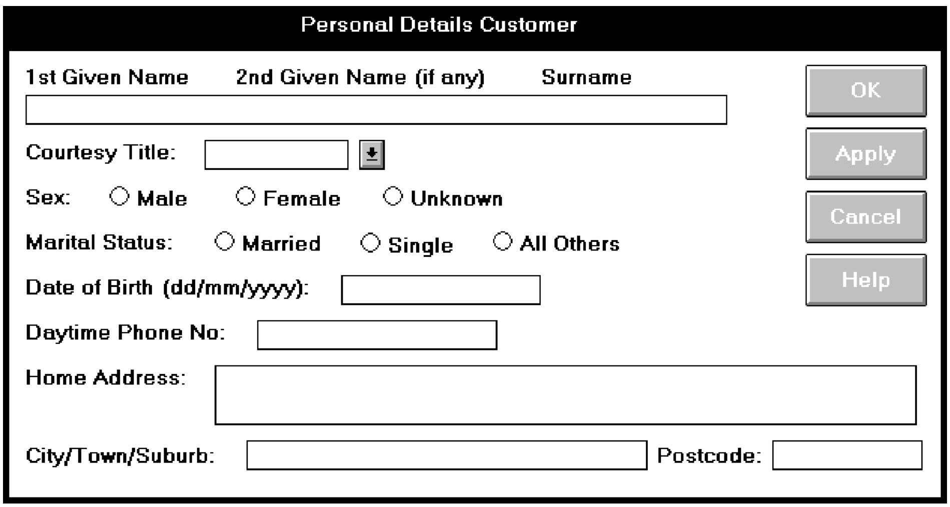
https://hemanthrajhemu.github.io

Screen 2.2

Note the improved alignment of this screen's controls. It is excellent, except for the single check box in the lower-left corner. Again, this check box can get lost here. This screen does have two problems. First, the headings, in mixed case and centered in the group boxes, visually compete with the information within the boxes for the viewer's attention. It would be much better if they were positioned away from the important screen information and capitalized to set them off visually from the captions and data. Second, for completeness and closure, a group box around the top five controls should also be included.

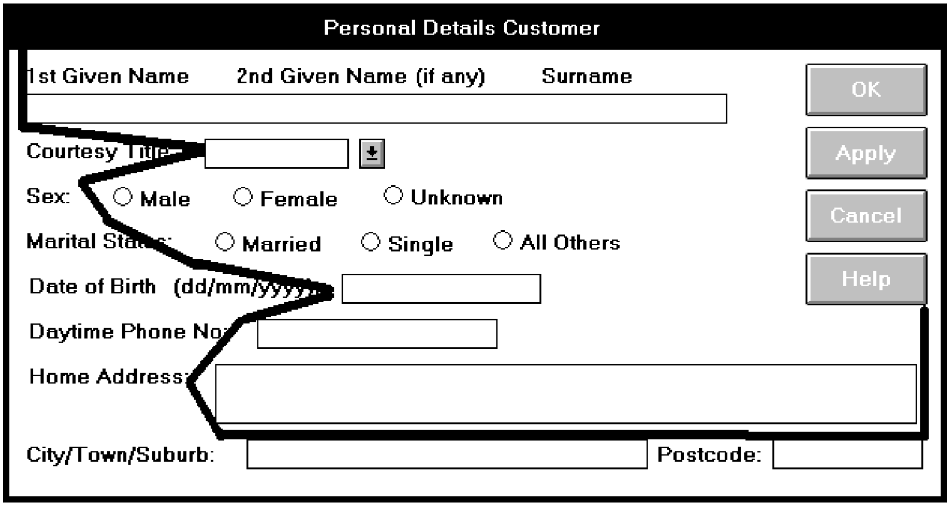
Example 3

This is an information entry/modification screen from a banking system. The major problem is very poor alignment of the controls, as illustrated by the eye scan requirements. There are some other problems. There are no groupings. Does the name control have a caption? Are the labels above the Name text box intended as captions? Are they intended as prompts? This is not clear. Is the prompt (dd/mm/yyyy) in the proper location? In its current position, it is set up as an aid to a novice or casual user of the system. For an expert user of the system, who does not need the prompt, it is positioned where it is visual noise. For the expert, it should be to the right side of the text box.



Screen 3.1A

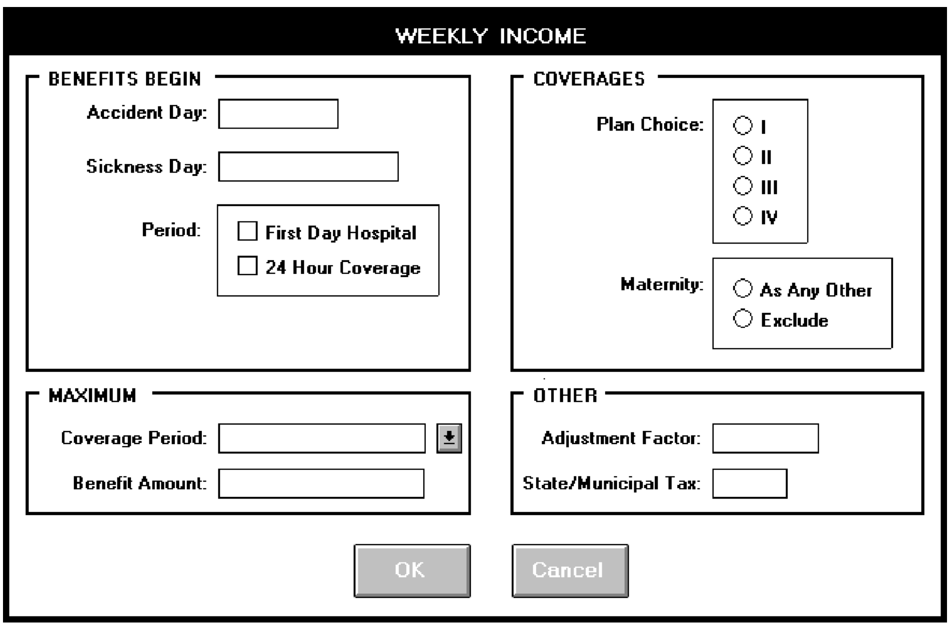
https://hemanthrajhemu.github.io



Screen 3.1B

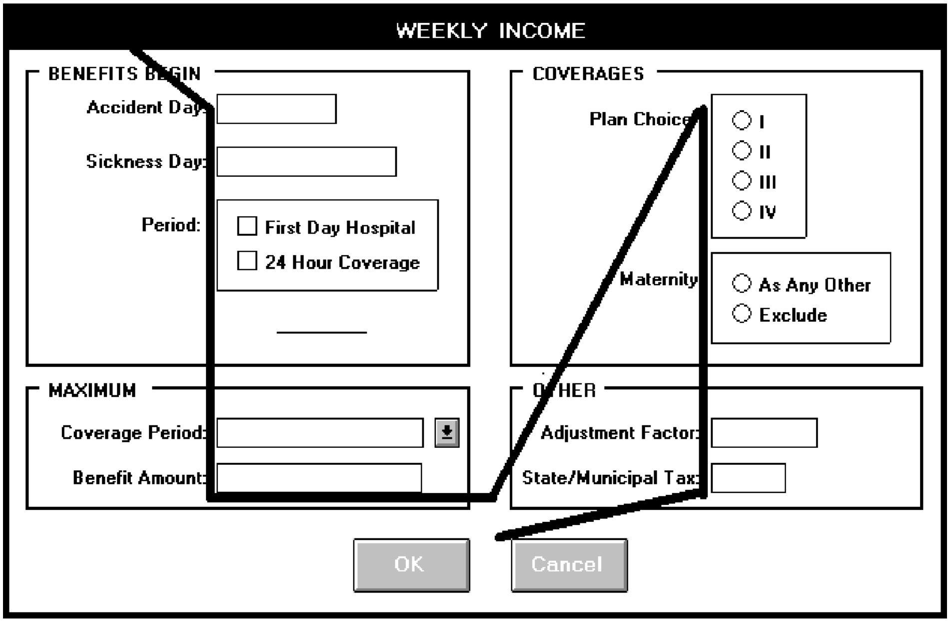
Example 4

This is a properly designed information entry/modification screen. It contains groupings reinforced by group boxes, alignment of controls, right-alignment of all captions (in this case) located consistently to the left of each control, capitalized section headings aligned to the upper-right of the group boxes, and command buttons centered at the bottom. Scanning of the columns of information is simplified by the control alignment. Headings do not compete with control captions for the viewer's attention. The screen also possesses balance.



Screen

https://hemanthrajhemu.github.io

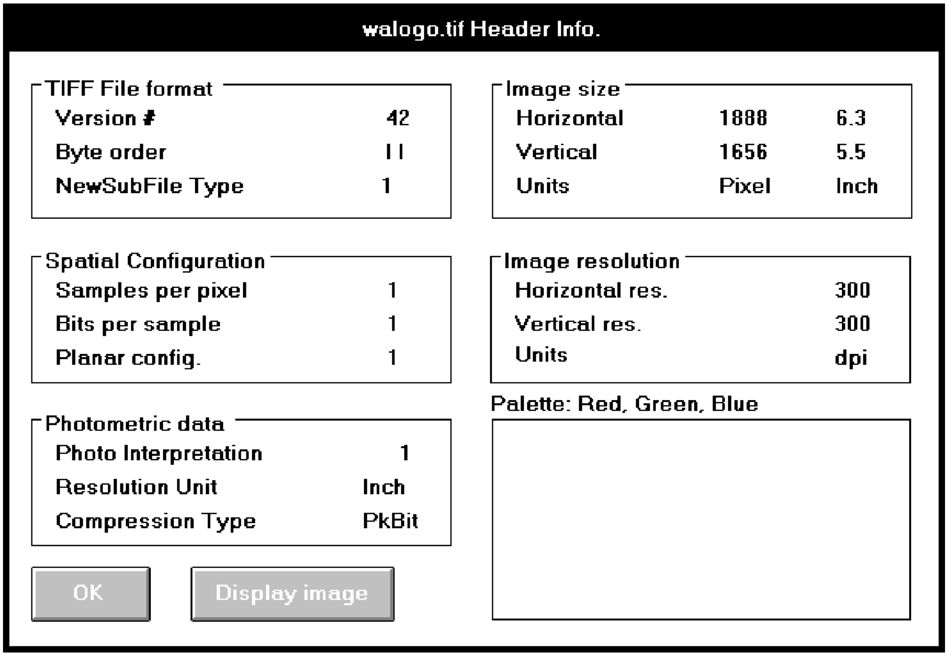


Screen 4.1B

Example 5

This is a read-only/display screen from the drawing program. It possesses good balance and nice groupings reinforced through use of group boxes. The problems are as follows. There are no colons with the captions. Other screens from this system have colons included with captions, violating the viewer's mental model of a caption possessing a colon. Which caption style is used, headline (all significant word initial caps) or sentence (capitalization of the first word only)? If you examine the screen you'll find many instances of both styles. The section headings are mixed case just like the control captions, competing with each other visually for the viewer's attention. The numeric data fields are not properly right-aligned although fairly good top-to-bottom scanning does exist. Units in the Image Size/Resolution groupings are at the bottom of listing. They would more appropriately be placed at the top because they appear to be column headings. In the Image Resolution grouping the use of "res" is redundant, since it already appears in the section heading.

https://hemanthrajhemu.github.io

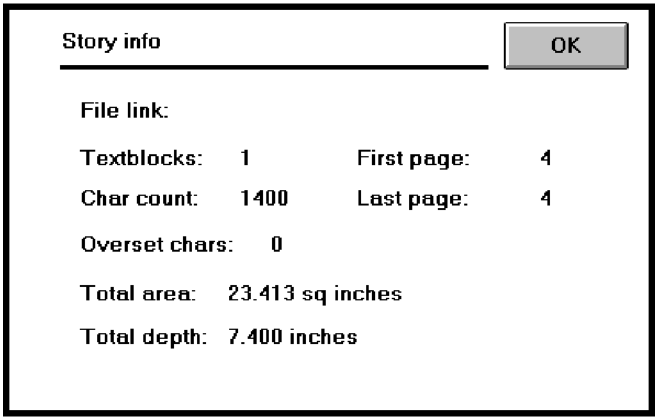


Screen 5.1

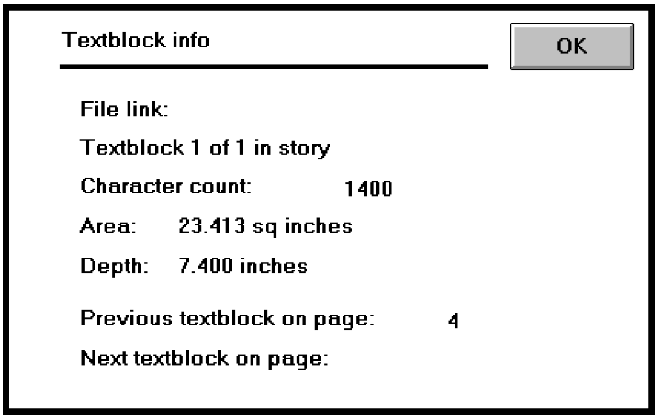
Example 6

Here is a pair of similar read-only/display screens from another system. Both are poorly aligned, the data getting buried among the captions. See the illustration of how Textblock Info will be scanned. Why abbreviate Info; there is plenty of room to spell it out. The groupings are not very strong, either. Notice the inconsistencies in captions, Character count versus Char count, Area versus Total area, and Depth versus Total depth. Again, there is room to spell them out fully. Otherwise, the viewer has to establish two mental models for the same element. Why establish this learning requirement?

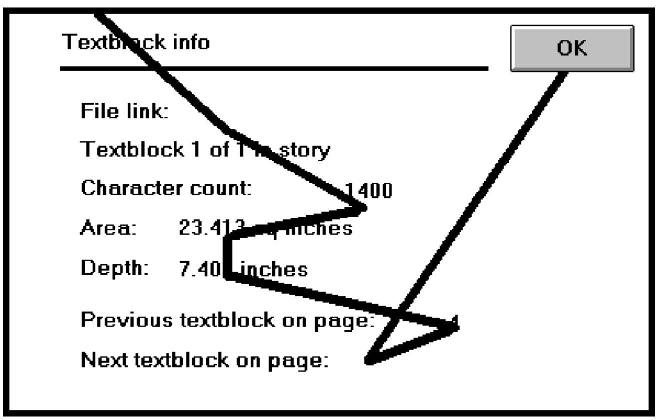
https://hemanthrajhemu.github.io



Screen 6.1A



Screen 6.1B



Screen 6.1C

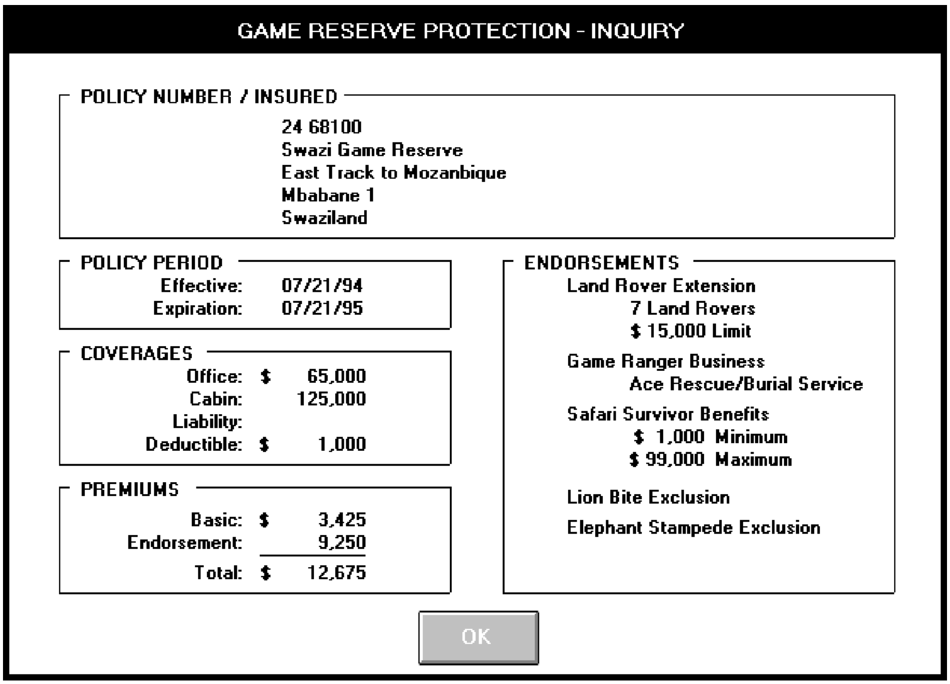
https://hemanthrajhemu.github.io

Example 7

Here are two properly designed read-only/inquiry screens, both possessing good alignment, grouping, and balance. The data is displayed on the screen background for ease of identification and comprehension.

Screen 7.1

From an insurance system, this screen summarizes some of the most important kinds of information a policyholder would want to know about an insurance policy. Captions are omitted in the POLICY NUMBER/INSURED section at the top. Contextually, this information is self-explanatory. The information presented in the ENDORSEMENTS section reflects the principle of displaying something only if it is present or applicable. If one or more of these endorsements were not included with another similar policy, they would not be displayed at all. In that case, the applicable endorsements would fill this section beginning at its top. The descriptive information included with the top three endorsements reflects the conversion of the more customary "caption: data" format into simple data statements. The data statements are self-explanatory; captions are not required.

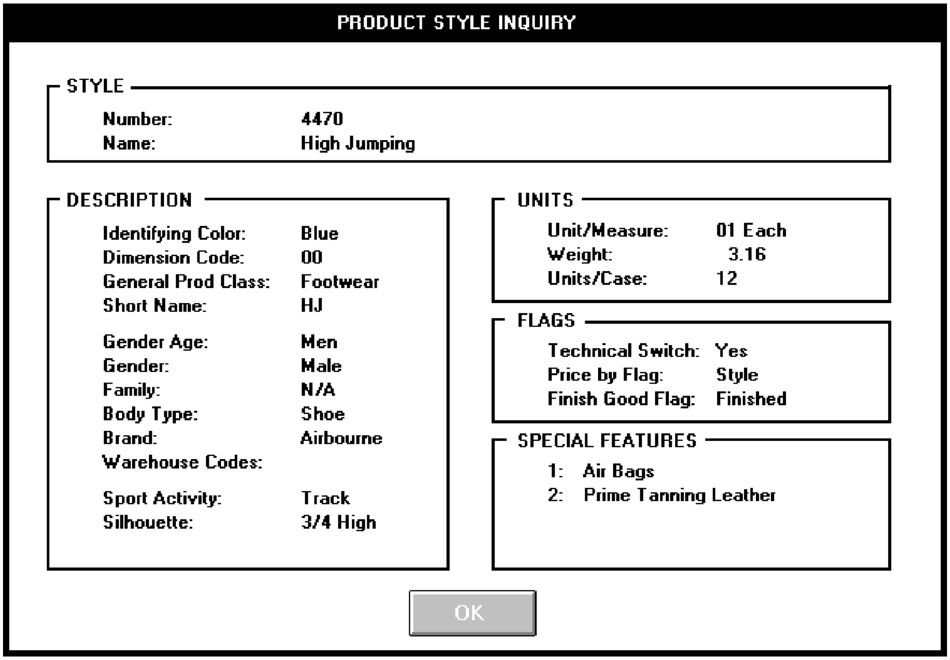


Screen 7.1

https://hemanthrajhemu.github.io

Screen 7.2

This screen from a shoe manufacturer is similar to the insurance screen in structure.


PRODUCT STYLE INQUIRY

STYLE

Number: 4470
Name: High Jumping

DESCRIPTION

Identifying Color: Blue
Dimension Code: 00
General Prod Class: Footwear
Short Name: HJ

Gender Age: Men
Gender: Male
Family: N/A
Body Type: Shoe
Brand: Airbourne
Warehouse Codes:

Sport Activity: Track
Silhouette: 3/4 High

UNITS

Unit/Measure: 01 Each
Weight: 3.16
Units/Case: 12

FLAGS

Technical Switch: Yes
Price by Flag: Style
Finish Good Flag: Finished

SPECIAL FEATURES

1: Air Bags
2: Prime Tanning Leather

OK

Screen 7.2

Step 3 Exercise

An exercise for Step 3 can be found on this book's companion Web site, www.wiley.com/college/galitz.

https://hemanthrajhemu.github.io

STEP

4

Develop System Menus and Navigation Schemes

A system contains large amounts of information and performs a variety of functions. Regardless of its purpose, the system must provide some means to tell people about the information it possesses or the things it can do. This is accomplished by displaying listings of the choices or alternatives the user has at appropriate points while using the system; or creating a string of listings that lead a person from a series of general descriptors through increasingly specific categories on following listings until the lowest level listing is reached. This lowest level listing provides the desired choices. These listings of choices are commonly called menus. Menus are a major form of navigation through a system and, if properly designed, assist the user in developing a mental model of the system. In this step, the following menu topics will be addressed:

- The structures of menus.
- The functions of menus.
- The content of menus.
- Formatting menus.
- Writing menus.
- Navigation using menus.
- Web site navigation and links.
- Web site navigation elements.
- Maintaining a sense of place in Web sites.
- Types of menus.

https://hemanthrajhemu.github.io

Menus are effective because they utilize the more powerful human capability of recognition rather than the weaker capability of recall. Working with menus reminds people of available options and information that they may not be aware of or have forgotten.

Menus are not without problems, however. New and inexperienced system users might find learning large systems difficult. Menu information must often be remembered and integrated across a series of screens. If each menu is viewed in isolation, relationships between menus may be difficult to grasp. Words and phrases with multiple meanings may also be wrongly interpreted because of the user's inability to see relationships between menus. Ambiguities, also, may not be correctly resolved if the user makes incorrect assumptions about menu structure. The frequent result is that people make mistakes and can get lost in the hierarchical structure.

Experienced system users, while finding menus helpful at first, may find them tedious as they learn the system. Continually having to step through a series of menus to achieve the desired objective can be time-consuming and frustrating. Therefore, the design of menu systems must consider the conflicting needs of both inexperienced and experienced users.

Graphical and Web systems are heavily menu-oriented. They vary in form and are applied in diverse ways. In graphical systems they are used to designate commands, properties that apply to an object, documents, and windows. When selected, a graphical menu item may lead to another menu, cause a window to be displayed, or directly cause an action to be performed. To accomplish these goals, a graphical system presents a variety of menu styles to choose from. Included are entities commonly called menu bars, and menus called pull-downs, pop-ups, cascades, tear-offs, and iconic. In Web site design, common menus include textual links to other pages, command buttons, and both graphical and textual toolbars.

In this step, graphical and Web system menus will be addressed. It will begin with a description of the kinds of menu structures available and their content, and then present a series of general menu design guidelines for formatting, phrasing, selecting choices, and navigating menus. Next, Web-specific navigation issues and guidelines will be discussed. Finally, specific types of graphical menus will be described, recommendations for proper usage given, and relevant specific guidelines summarized.

Structures of Menus

Menus vary in form from very simple to very complex. They may range from small dialog boxes requesting the user to choose between one of two alternatives, to hierarchical tree schemes with many branches and levels of depth. A menu's structure defines the amount of control given to the user in performing a task. The most common structures are the following.

Single Menus

In this simplest form of menu, a single screen or window is presented to seek the user's input or request an action to be performed, as illustrated in Figure 4.1. In using

https://hemanthrajhemu.github.io

the Internet, for example, at a point in the dialog people may be asked if they wish to "Stay Connected" or "Disconnect." In playing a game, choices presented may be "novice," "intermediate," or "expert." Single menus conceptually require choices from this single menu only, and no other menus will follow necessitating additional user choices. The user need only consider the immediate consequences of the item being chosen and need not be concerned with any other additional system menus. While other single menus may exist in the system and might be encountered later, these other menus are not perceived by the user as comprising a series of choices.

A single menu may be iterative if it requires data to be entered into it and this data input is subject to a validity check that fails. The menu will then be represented to the user with a message requesting reentry of valid data.

Sequential Linear Menus

Sequential linear menus are presented on a series of screens possessing only one path. The menu screens are presented in a preset order, and, generally, their objective is for specifying parameters or for entering data. The length of the path may be short or long, depending upon the nature of the information being collected. All the menus are important to the process at hand and must be answered in some manner by the user. Sequential linear menus are illustrated in Figure 4.2.

Sequential path menus have several shortcomings. A long sequence may become tedious as menu after menu is presented. The user may not remember an answer to a previous question, a question important to the currently presented choices. The user may also want to return to a previous menu to change an answer or look at an answer, an awkward process that must be allowed. Finally, the user may, conceptually, want to complete the menus in a different order than that in which they are being presented.

Simultaneous Menus

Instead of being presented on separate screens, all menu options are available simultaneously, as illustrated in Figure 4.3. The menu may be completed in the order desired by the user, choices being skipped and returned to later. All alternatives are visible for reminding of choices, comparing choices, and changing answers. The tedium associated with a long series of sequential menus is greatly reduced.

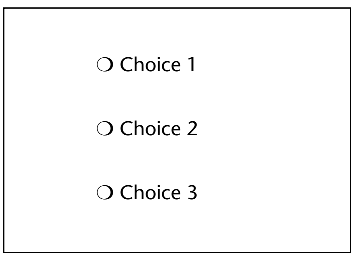


Figure 4.1: Single menu.

https://hemanthrajhemu.github.io

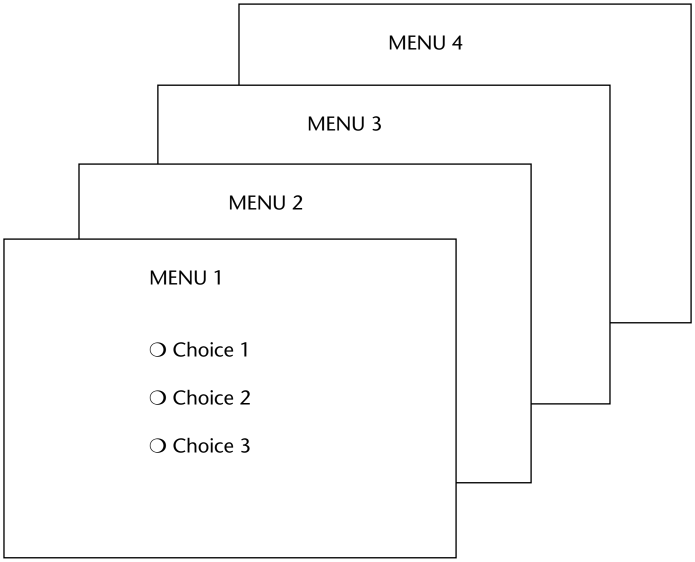


Figure 4.2: Sequential linear menus.

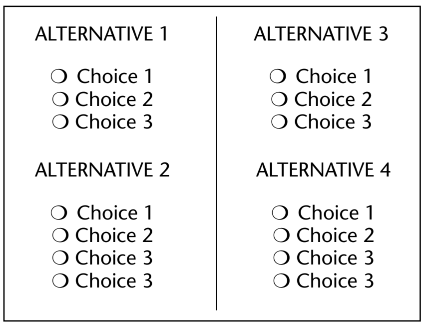


Figure 4.3: Simultaneous menus.

Problems with simultaneous menus are that for large collections of menu alternatives screen clutter can easily occur, and screen paging or scrolling may still be necessary to view all the choices. This type of menu must also clearly indicate menu choice relationships and dependencies, something better accomplished in a linear menu string, or a hierarchical menu, described next. Presenting many menu dependencies and relationships on a screen, especially if poorly indicated, can also be very confusing for a novice user.

Hierarchical or Sequential Menus

When many relationships exist between menu alternatives, and some menu options are only appropriate depending upon a previous menu selection, a hierarchical structure is the best solution. In Web site design, hierarchical menus are often referred to as

https://hemanthrajhemu.github.io

sequential menus. A hierarchical structure results in an increasing refinement of choice as menus are stepped through, for example, from options, to suboptions, from categories to subcategories, from pages to sections to subsections, and so on. A hierarchical structure can best be represented as an inverse tree, leading to more and more branches as one moves downward through it. Hierarchical structures are characterized by depth and breadth, depth being the number of choice levels one must traverse to reach the destination, breadth being the number of alternatives found at each level. Menu depth and breadth has been a well-researched topic and will be fully discussed in succeeding pages. Common examples of hierarchical design today are found in menu bars with their associated pull-downs, and in Web sites with their navigation links.

The order and structure of branching in a hierarchy is preset and the normal order of flow one-way: top down. A disadvantage of a hierarchical scheme is that the defined branching order may not fit the user's conception of the task flow. If users are not familiar with the hierarchical menu, or are unable to predict what suboptions lie below a particular choice, they may go down wrong paths and find it necessary to go back up the tree to change a choice, or perhaps even return to the top-level menu. It is important, then, that hierarchies be consistent with user expectations, and that choice uncertainties be reduced as much as possible. It must also be easy to back upward through the tree to facilitate exploration of the tree.

Hierarchical menus are illustrated in Figure 4.4. Note that the top level of the tree is considered level 0 with subsequent levels numbered sequentially beginning with number 1. Starting at the top, level 0, two selections, or mouse clicks, are required to reach level 2.

Connected Menus

Connected menus are networks of menus all interconnected in some manner. Movement through a structure of menus is not restricted to a hierarchical tree, but is permitted between most or all menus in the network. From the user's perspective there is no top-down traversal of the menu system, but an almost unhindered wandering between any two menus of interest. A connected menu system may be cyclical, with movement permitted in either direction between menus, or acyclical, with movement permitted in only one direction. These menus also vary in connectivity, the extent to which menus are linked by multiple paths. (In a hierarchical menu system, the ability to go back to a previous menu or to return to the top-level menu are also examples, although restricted, of connected menus.)

The biggest advantage of a connected menu network is that it gives the user full control over the navigation flow. Its disadvantage is its complexity, and its navigation may be daunting for an inexperienced user. An example connected menu structure is represented in Figure 4.5.

https://hemanthrajhemu.github.io

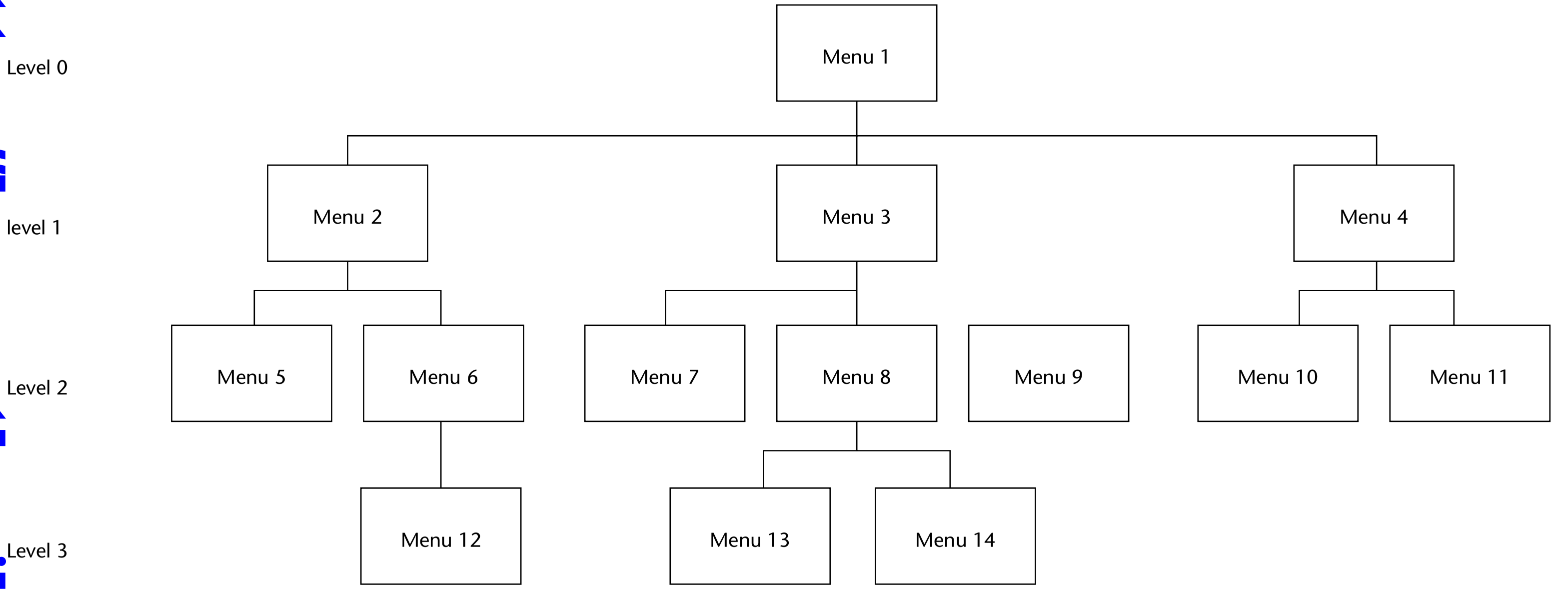


Figure 4.4: Hierarchical or sequential menus.

https://hemanthrajhemu.github.io

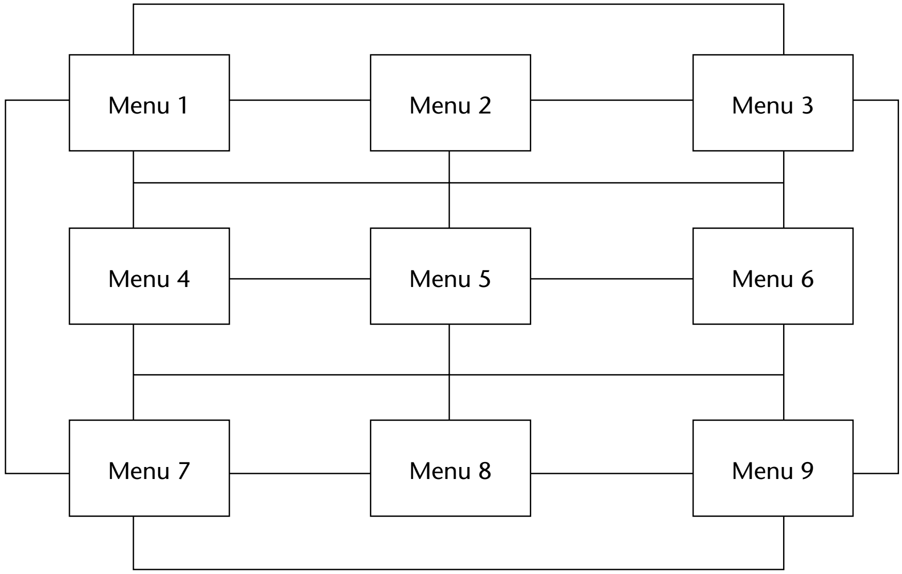


Figure 4.5: Connected menus.

Event-Trapping Menus

Event-trapping menus provide an ever-present background of control over the system's state and parameters while the user is working on a foreground task. They are, in essence, a set of simultaneous menus imposed on hierarchical menus. In a graphical system, for example, existing together are a simultaneous menu, the menu bar, and a hierarchy — the menu bar and its pull-downs. Event-trapping menus generally serve one of three functions; (1) They may immediately change some parameter in the current environment (bold a piece of text), (2) they may take the user out of the current environment to perform a function without leaving the current environment (perform a spell check), or (3) they may exit the current environment and allow the user to move to a totally new environment (Exit).

These menus can also change content based upon the system state, or an event, existing at that moment. A Paste option in a word-processing application, for example, will only function if there is something in a clipboard to paste. A Grid option on a pull-down, as another example, will toggle between a "Hide Grid" or "Show Grid" state, depending upon whether or not a grid is displayed on the screen at that moment. Event-trapping menus such as menu bars are constantly available to aid in establishing a sense of context, or where one is, while things may be changing in the foreground.

Functions of Menus

From the user's perspective, a menu can be used to perform several functions: to navigate to a new menu, to execute an action or procedure, to display information, or to input data or parameters.

https://hemanthrajhemu.github.io

Navigation to a New Menu

Each user selection causes another menu in a hierarchical menu tree to be displayed. The purpose of each selection is to steer the user toward an objective or goal. Selection errors may lead the user down wrong paths, and cost time and, perhaps, aggravation, but these errors are nondestructive and usually undoable.

Execute an Action or Procedure

A user selection directs the computer to implement an action or perform a procedure. The action may be something like opening or closing a file, copying text, or sending a message. In some cases, execution may only occur after a hierarchical menu tree is navigated. In other cases, actions may be performed as successive hierarchical menus are encountered and traversed. Selection errors may or may not have serious consequences, depending upon the nature of the action. Accidental selection of critical irreversible actions must be prevented in interface design.

Displaying Information

The main purpose of selecting a menu choice may simply be to display information. The user may be searching for specific information in a database or browsing the Web. The user's focus is primarily on the information desired and less on the selection function. In many cases, information retrieval may occur only after a hierarchical menu tree is navigated. The content material and the user's interests will determine the paths followed. Users may spend considerable time and effort understanding and processing uncovered information to evaluate subsequently displayed menu choices. Wrong turns in the process will again cost time and perhaps aggravation, but these errors are nondestructive and usually undoable.

Data or Parameter Input

Each selection specifies a piece of input data for the system or provides a parameter value. Data or values may be input on a single menu or spread over a hierarchy of menus. The user's focus is primarily on the information being provided and, again, less on the selection function. Selection errors can be easily corrected if detected by the system.

Content of Menus

A menu consists of four elements: its *context*, its *title*, its *choice descriptions*, and its *completion instructions*. These concepts are introduced here and will be expanded in detailed guidelines to follow on succeeding pages.

https://hemanthrajhemu.github.io

Menu Context

A menu's context provides information to keep the user oriented. This kind of information is critical in complex or hierarchical menu systems, where loss of position or disorientation can easily occur. Feedback is necessary that tells users where they are in a process, what their past choices were, and possibly how much farther they still have to navigate. Human memory being what it is, where one is and how one got there all too easily slip from consciousness.

Verbal linkage, spatial linkage, or both may be used to provide navigation feedback. *Verbal linkage* involves providing, on the current menu screen, a listing of choices made on previous menus that have led to this position. It also involves assuring the user that the displayed menu is the menu desired. Its title should mirror the option selected on the previous menu, and its content should reflect its title. *Spatial linkage* can be accomplished by graphic methods. Each succeeding menu screen can be displayed overlapping the previous menu screen so a succession of choices can be seen in a single view. A sense of progress and distance can then be easily ascertained.

Menu Title

A menu's title provides the context for the current set of choices. The title must reflect the choice selected on the previously displayed menu.

Choice Descriptions

Choice descriptions are the alternatives available to the user. These descriptions can range from a mnemonic, numeric, or alphabetized listing of choices, to single words or phrases, to full sentences, or more. The style chosen will reflect the experience of the user (novice or expert), the nature of the choices (well-learned alternatives or not), the nature of the selection mechanism (keyboard or mouse), and the nature of the system (business system application or Web page).

Completion Instructions

Completion instructions tell users how to indicate their choices. They may include the rationale for why the user is being asked to make this choice and the impact the choice will have on subsequent processes. Explicit instructions may be needed for first time or casual users of a system. Experienced users will find overly verbose instructions unnecessary. The needs of all system users, and the nature of the system, must again be considered in creating this kind of on-screen guidance.

Formatting of Menus

The human-computer interface has a rich history of experimental studies with menus, the results of which can and have been applied to graphical screen and Web page menu design and presentation. What follows is a series of guidelines for formatting menus.

https://hemanthrajhemu.github.io

Consistency

- Provide consistency with the user's expectations.
- Provide consistency in menu
  - — Formatting, including organization, presentation, and choice ordering.
  - — Phrasing, including titles, choice descriptions, and instructions.
  - — Choice selection methods.
  - — Navigation schemes.

Like all aspects of screen design, menu design consistency is an integral component of system usability. Menu formatting, phrasing, choice selection, and navigation must be consistent throughout a graphical system or Web site.

Display

- If continual or frequent references to menu options are necessary, permanently display the menu in an area of the screen that will not obscure other screen data.
- If only occasional references to menu options are necessary, the menu may be presented on demand.
  - — Critical options should be continuously displayed, however.

Whether to display a menu continually, or on demand, is determined by the menu's frequency of use. Always permanently display menus that are frequently referenced. This will provide memory support and immediate access to what is needed most. Occasionally needed menus may be presented on request via pop-ups or pull-downs. Critical options, however, should always be continuously displayed. Wright, Lickorish, and Milroy (1994) found superior performance for permanently displayed menus as opposed to menus that had to be retrieved through mouse clicks. They speculate that because retrieving a menu for display requires more actions, this may also impair people's memory for other tasks being performed.

Presentation

- Ensure that a menu and its choices are obvious to the user by presenting them with a unique and consistent structure, location, and/or display technique.
- Ensure that other system components do not possess the same visual qualities as menu choices.

A menu and its choices should be immediately recognizable by the user as being a menu of choices. This can be accomplished through giving the menu a distinctive and consistent structure and presenting it in a consistent screen or page location. Presentation techniques must, of course, be compatible with those used for other purposes on the remainder of the screen. A good way to set a menu off from the remainder

https://hemanthrajhemu.github.io

of the screen is to enclose it in a box or display it using a background that contrasts with the remainder of the screen. Techniques chosen should be consistent throughout the system. Web page navigation links, which may be scattered throughout a page, are displayed underlined and in a unique color to differentiate and identify them.

Ensure that other system elements do not possess qualities that allow users to confuse them with menu choices. In Web page design, for example, the underlining of any system component other than navigation links is not recommended because of the possibility that they may be confused with links.

Organization

- Provide a general or main menu.
- Display
  - — All relevant alternatives.
  - — Only relevant alternatives.
    - Delete or gray-out inactive choices.
- Match the menu structure to the structure of the task.
  - — Organization should reflect the most efficient sequence of steps to accomplish a person's most frequent or most likely goals.
- Minimize number of menu levels within limits of clarity.
  - — For Web sites, restrict it to two levels (requiring two mouse clicks) for fastest performance.
- Be conservative in the number of menu choices presented on a screen.
  - — Without logical groupings of elements, limit choices to 4 to 8.
  - — With logical groupings of elements, limit choices to 18 to 24.
- Provide decreasing direction menus, if sensible.
- Never require menus to be scrolled.

In organizing a menu, the goal is to simply and effectively reveal its structure, while also reducing the number of actions needed to locate the target item.

General menu. The top-level menu in a hierarchical menu scheme should be a general or main menu, consisting of basic system options. This will provide a consistent starting point for all system activities and a "home base" to which the user may always return.

Relevant alternatives. A menu should provide all relevant alternatives, and only relevant alternatives, at the point at which it is displayed. Including irrelevant choices on a menu screen increases learning requirements and has been found to interfere with performance. There are two exceptions to this rule, however. Alternatives that are conditionally inactive may be displayed along with the conditionally active choices, if the active choices can be visually highlighted in some manner (such as through bolding or reverse video), or the inactive choices can be visually subdued (perhaps as through graying them out). One study, however, found that completely eliminating inactive alternatives on a menu resulted in faster choice access time, when compared to leaving inactive alternatives on a

https://hemanthrajhemu.github.io

menu, but displayed in a subdued manner. This study concludes that eliminating conditionally inactive choices from a menu appears to be the best approach. Mayhew (1992) suggests that while deletion does provide an advantage to expert users of keyboard-driven menus, graying out seems to be advantageous to novices in systems using pointer-driven selection devices. She concludes that because menus are geared toward novices, graying appears to be the best overall choice. In general today's graphical systems follow the gray-out approach for inactive menu choices. Whatever method is chosen should be consistently followed throughout a system. Options to be implemented in the future may also be displayed if they can be visually marked in some way (through a display technique or some other annotation).

Matching menu structure to the tasks. Menus should be organized according to how people structure their tasks. They should reflect the most efficient sequence of steps to accomplish a person's most frequent or likely goals.

Minimize number of levels. The issue that must be addressed in creating a multilevel menu structure is determining how many items will be placed on one menu (its breadth) and how many levels it will consume (its depth). In general, the more choices contained on a menu (greater breadth), the less will be its depth; the fewer choices on a menu (less breadth), the greater will be its depth.

The advantages of a menu system with greater *breadth* and less depth are

- Fewer steps and shorter time to reach one's objective.
- Fewer opportunities to wander down wrong paths.
- Easier learning by allowing the user to see relationships of menu items.

A broad menu's disadvantages are

- A more crowded menu that may reduce the clarity of the wording of choices.
- Increased likelihood of confusing similar choices because they are seen together.

The advantages of greater *depth* are

- Less crowding on the menu.
- Fewer choices to be scanned.
- Easier hiding of inappropriate choices.
- Less likelihood of confusing similar choices because there is less likelihood that they will be seen together.

Greater depth disadvantages are

- More steps and longer time to reach one's objective.
- More difficulties in learning because relationships between elements cannot always be seen.
- More difficulties in predicting what lies below, resulting in increased likelihood of going down wrong paths or getting lost.
- Higher error rates.

In text-based and graphical systems, a good number of studies have looked at the breadth-depth issue. Included are studies by Kiger (1984) and Jacko and

https://hemanthrajhemu.github.io

Salvendy (1996). Some have concluded that breadth is preferable to depth in terms of either greater speed or fewer errors, that a low number of levels (two to three) and an intermediate number of choices (four to eight) results in faster, more accurate performance as opposed to fewer or greater numbers of levels and choices, and that four to eight choices per menu screen is best. Another study found that one level was easiest to learn, and a couple of studies have concluded that a menu could contain up to 64 items if it were organized into logical groups. The least desirable alternative in almost all cases was deep-level menus that simply presented the user with a binary choice (select one of two alternatives) on each menu. In an early study of hypertext, Snowberry et al. (1983) found that as hypertext depth increased, performance and preference declined and errors increased. The general conclusion of these studies:

- People found resources faster in, and understood better, broader, shallow menu structures than narrow deep ones.

In Web site design, studies have also looked at the breadth-depth issue (Zaphiris and Mtei, 1998; Larson and Czerwinski, 1998). Both found that a two-level (two mouse click) Web site was searched faster than those containing more levels. Zaphiris (2001) modeled user performance in menu search using the results of previous studies and concluded that a menu design of either extreme (very deep or very broad) undermines learnability and usability.

Straub (2003a) describes additional research that suggests a concave hierarchical menu structure is optimal for browsing (Norman and Chin, 1988; Bernard, 2002). A concave shape presents a broad initial selection screen followed by menus with small categories. The terminal menu then contains an option set that is again somewhat broad. Straub's broad conclusion from the research is this:

- **Too deep is too deep.** People have a more difficult time understanding, and consequently navigating, deep sites.
- **Too broad is too broad.** Extremely broad sites also present a challenge to efficient navigation.
- **Effective sub-grouping reduces perceived breadth.** Logical grouping improves performance for even the broadest structures.
- **Clear labels improve navigation accuracy.** Creating clear and distinct labels for navigation elements enhances performance (as described in Step 3.)

The conclusion that one might derive from these studies is this: Fewer levels of menus aid the decision-making process, but trying to put too many choices on a single menu also has a negative impact. The final solution is a compromise: Minimize the number of levels within limits of clarity. What is clarity? The studies seem to indicate that, if the choices to be displayed cannot be segmented into logical categories, then confine the number of alternatives displayed to four to eight per menu. (Straub suggests a maximum of 16 for links.) If logical categorization is possible, and meaningful, logical category names can be established, then a larger number of choices can be presented. The maximum number of alternatives will, however, be dependent upon the size of the words needed to describe the alternatives to the user. Wordy captions will greatly restrict the number of alternatives capable of being displayed. There is one exception to these

https://hemanthrajhemu.github.io

basic principles. Large, linearly ordered, well-learned listings, such as months of the year, or numbers, would be better presented in a one-level menu, rather than by breaking them into multiple levels.

Limit the number of choices. Be conservative in the number of menu choices presented on a screen. If the choices cannot be logically grouped, restrict the number to four to eight as just described. If the choices can be grouped, 18 to 24 can be displayed, with no more than 10 items within a group. Mayhew (1992) suggests that if the menu choices are complex and/or there are no groupings of items, choices presented should be restricted to *10 or fewer*. This recommendation is similar to the eight or fewer recommendations above. If the menu choices are not complex, items on the menu can be grouped, and the users are infrequent or casual, she recommends *20 or fewer* choices. If the menu choices are not complex, items on the menu can be grouped, and the users are frequent or expert, she suggests *21 or more* choices can be provided.

MAXIM **The best journey is the one with the fewest steps.**

Provide decreasing direction menus. In addition to breadth and depth, direction has been found to affect menu choice selection performance. In a multilevel menu, a *decreasing* direction structure presents successively fewer choices as each lower level is traversed. An *increasing* direction structure presents successively more choices as each lower level is traversed. Bishu and Zhan (1992), in a study of 16 and 32 item iconic menus, found that decreasing direction menus were significantly faster and more accurate than increasing menus. As just described, however, the research of Norman and Chin (1988) and Bernard (2002) suggests a concave hierarchical menu structure is optimal for browsing.

Scrolling. Never require menus to be scrolled. Keep all choices visible at all times.

Complexity

- Provide both simple and complex menus.
- Simple: a minimal set of actions and menus.
- Complex: a complete set of actions and menus.

Providing two sets of menus will more effectively satisfy the differing needs of the novice and expert user. The novice or casual user often only requires a minimal set of actions and menus to accomplish tasks. The expert may prefer a full set of options. Make selection, and changing, between simple and complex menus easy to accomplish, preferably through a menu bar choice. IBM's SAA CUA refers to these menus as *short* and *full.*

https://hemanthrajhemu.github.io

Item Arrangement

- Align alternatives or choices into single columns whenever possible.
  — Orient for top-to-bottom reading.
  — Left-justify descriptions.
- If a horizontal orientation of descriptions must be maintained
  — Organize for left-to-right reading.

For scanning ease, menu choices should be left-justified and aligned vertically into columns. Research has found that columnar menus and listings are searched much faster than horizontally-oriented menus. Do not array a menu in multiple columns.

When menus are included on other screens, space constraints often exist, and the menu must be arrayed horizontally. If a single-row (horizontal) orientation is necessary organize for left-to-right reading. If two or more rows are available for displaying choices, organize for top-to-bottom, then left-to-right reading to facilitate visual scanning.

Ordering

- Order lists of choices by their natural order, or
  — For lists associated with numbers, use numeric order.
  — For textual lists with a small number of options (seven or less), order by
    - Sequence of occurrence.
    - Frequency of occurrence.
    - Importance.
    - Semantic similarity.
- Use alphabetic order for
  — Long lists (eight or more options).
  — Short lists with no obvious pattern or frequency.
- Separate potentially destructive actions from frequently chosen items.
- Maintain a consistent ordering of options on all related menus.
  — For variable-length menus, maintain consistent relative positions.
  — For fixed-length menus, maintain consistent absolute positions.

Within information categories included on a menu, or in menus in which categories are not possible, options must be ordered in meaningful ways. When a menu contains multiple categories of information, the ordering of categories will follow these same principles. A meaningful ordering is necessary to

- Facilitate search for an item.
- Provide information about the structure and relationships among items.
- Provide compatibility with the user's mental model of the item structure.
- Enhance the user's ability to anticipate a choice's location.

https://hemanthrajhemu.github.io

When items are organized along some dimension or characteristic, the user can use that information to locate items faster. An alphabetized list, for example, provides an indication of approximately where in the listing an item beginning with a particular letter will be found. Understanding structure and relationships, item similarities and dissimilarities, can also aid in focusing attention on that which is relevant. Any incompatibility with the user's mental model will disrupt searching as the user tries to make sense of something that had been well understood, but now is being presented in a way that has not been well learned. Months of the year presented in alphabetic order, for example, would be very disrupting.

Experienced users often anticipate the location of a desired choice within a familiar menu. Hornof and Kieras (1999), in studying how items are selected from pull-down menus, found that people often make an initial eye and mouse-positioning movement toward the expected choice location before the pull-down even appeared on the screen. They also found that choices in the top three positions of the pull-down were selected faster than those in other positions. This may have been caused by users' ability to better predict a choice's location at the top, and/or the shorter mouse movement required from the menu bar to the pull-down. Observational studies also reveal that experienced users also anticipate the location of command buttons appearing within a window. While waiting for a window to appear upon which a command button will be immediately "clicked," the pointer is often positioned at the button's expected location before the window appears.

Another study, Byrne, John, Wehrle, and Crow (1999), studied how people search unfamiliar pull-down menus. They found that the search primarily flowed from menu top to bottom, and that the initial eye fixation was usually focused on the choice in the topmost menu position. Almost all recorded eye fixations were on one of the first three items.

Both of the studies point out the importance of presenting important menu items at the top of menu arrays, and providing consistency in menu organization schemes and menu locations. Common ordering schemes for menus, then, are the following:

Natural ordering. If items have a natural sequence, such as chapters in a book, days in a week, or months in the year, the ordering scheme should follow this natural sequence. The screen viewer will have learned these ordering schemes very well.

Numeric ordering. Use numeric ordering for choices associated with numbers, for example, type size, baud rate, or number of pixels.

Small number of options. For groupings with a small number of options (about seven or fewer), *sequence of use*, *frequency of use*, or *importance* are good ordering schemes. Also consider ordering by *semantic similarity*, along a semantic dimension such as impact, potency, or emphasis. Type style, for example, may be ordered by emphasis from least to most: regular, underlined, italicized, and bold.

Alphabetic order. For a large number of options, alphabetic ordering of alternatives is desirable. Alphabetic ordering is also recommended for small lists where no frequency or sequence pattern is obvious. It has been found that alphabetically ordered menus can be searched much faster than randomly ordered menus. One study, for example, found that an 18-item alphabetic menu was visually searched four times faster than a randomly organized menu. Search time was a function of

https://hemanthrajhemu.github.io

saccadic eye movements through the display. Search patterns were random, but fewer eye movements were required with the alphabetic arrangement. After 20 trials, however, only one eye movement was required for all conditions, and search time was the same. Another study has found that the longer the list, the greater the value of an alphabetic ordering scheme. As list length increased, the time to find items in longer random lists increased significantly faster than the time to find items in longer alphabetic lists. Learning of a randomly ordered menu will eventually take place, but this learning will be greatly aided by a meaningful choice-ordering scheme.

Separate destructive choices. Destructive menu choices, such as delete or clear, should be positioned as far away from frequently chosen choices as possible to minimize the chance of accidental selection.

Consistency between menus. Options found on more than one menu should be consistently positioned on all menus. If menus are of variable length, maintain relative positioning of all item options (for example, always place Exit at the bottom or end of the list). If menus are of fixed length, place options in the same physical position within the list.

Groupings

- Create groupings of items that are logical, distinctive, meaningful, and mutually exclusive.
- Categorize them in such a way as to
  — Maximize the similarity of items within a category.
  — Minimize the similarity of items across categories.
- Present no more than six or seven groupings on a screen.
- Order categorized groupings in a meaningful way.
- If meaningful categories cannot be developed and more than eight options must be displayed on a screen, create arbitrary visual groupings that
  — Consist of about four or five, but never more than seven, options.
  — Are of equal size.
- Separate groupings created through either
  — Wider spacing, or
  — A thin ruled line.
- Provide immediate access to critical or frequently chosen items.

Create groupings. Items displayed on menus should be logically grouped to aid learning and speed up the visual search process. Studies have demonstrated that logically categorized menus are easier to learn and result in faster and more accurate performance. Categorical organization may facilitate the transition from novice to expert user because information is visually represented in the way people think about it.

https://hemanthrajhemu.github.io

Categorizing. Groupings should also cover all the possibilities and contain items that are non-overlapping. While some collections of information will be easily partitioned into logical groups, others may be very difficult to partition. Some users may not understand the designer's organizational framework, and there may be differences among users based on experience. Thus, no perfect solution may exist for all, and extensive testing and refinement may be necessary to create the most natural and comprehensible solution.

Number. Limit the number of groupings on a screen to six or seven. The total number of items within all the groupings should not exceed about 18 to 24.

Ordering. Groupings of menu items may be ordered following the guidelines described in "Ordering" earlier in this step. Ordering alternatives include alphabetic, sequence of use, frequency of use, importance, and semantic similarity.

Arbitrary visual groupings. Uncategorized menus should be broken in arbitrary visual groupings through the use of space or lines. Groups should be as equal in size as possible and consist of about four or five options. Groupings should never exceed more than seven options.

Separation. Perceptually separate groupings by a leaving a wider spacing between groupings, or by inscribing line separators between groupings. Guidelines for displaying line separators follow.

Critical choices. Choices that are critical or frequently chosen should be accessible as quickly and through as few steps as possible. Place them on the highest-level menu, whenever possible.

Line Separators

- Separate vertically arrayed groupings with subtle solid lines.
- Separate vertically arrayed subgroupings with subtle dotted or dashed lines.
- For subgroupings within a category,
  — Left-justify the lines under the first letter of the columnized choice descriptions.
  — Right-justify the lines under the last character of the longest choice description.
- For independent groupings,
  — Extend the line to the left and right menu borders.

Inscribing subtle solid or dashed lines between groupings can reinforce groupings and subgroupings of vertically arrayed related choices. For breaking subgroupings within one category, the line or lines should only extend from the first character of the descriptions to the end of the longest description, as shown in Figure 4.6. Many graphical platforms always extend the line from menu border to border, as illustrated in Figure 4.7. This extended line results in too strong a visual separation between what are related menu parts. Visual separation should exist, but it should not be too overpowering.

https://hemanthrajhemu.github.io

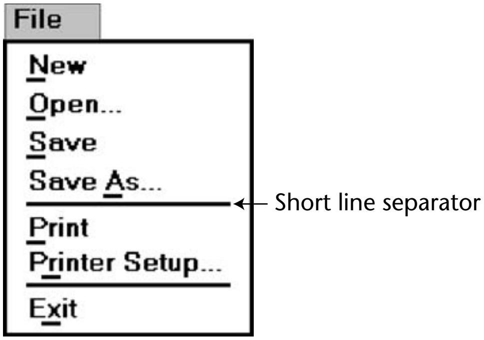


Figure 4.6: Partial line separators.

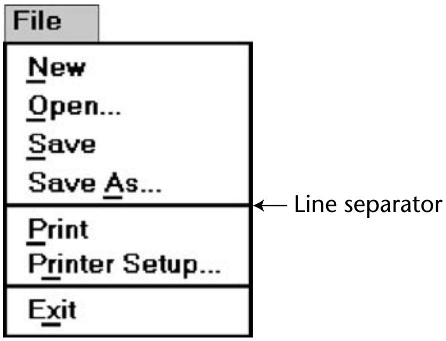


Figure 4.7: Extended line separators.

For independent groups of choices, extend the horizontal line from menu border to border. This will indicate to the user that the groupings are independent of one another. In summary, use a partial line for separating related choices; use an extended line for separating unrelated or independent choices.

Selection Support Menus

- When a small, discrete set of functions is accessed 90 percent or more of the time
  — Use Folded menus.
- When a small set of items is selected between 31 percent and 89 percent of the time and the other items are selected with lower frequencies
  — Use Split menus.
- If there is no small, discrete set of items that is used 30 percent of the time or more
  — Use traditional menus.
- Do not reorder menus.

https://hemanthrajhemu.github.io

Selection support menus support the user by providing faster access to frequently used functions. More frequent choices are positioned at the top of a menu where a person's eye initially goes when a menu is presented. High-frequency items at the top may be specified by the menu designer and remain static or unchangeable, or the menu may be restructured reflecting a person's use pattern over a period of time. Changeable menus are called *adaptive* menus. Duplicating the more frequent items in a separate section at the top is commonly called a *split* menu. Sears and Shneiderman (1994) showed that split menus were better than traditional menus both performance-wise and preference-wise. A traditional menu is illustrated in Figure 4.8, a split menu in Figure 4.9.

A version of the split menu is one called the *folded* menu. In this menu the high-frequency items appear first and alone. The complete menu appears after a time delay, or after the user clicks on a down arrow at the bottom of the initial menu. The additional choices are displayed "below the fold" as illustrated in Figure 4.10.

In addition to providing faster access to choices in a location where the user first looks, it is argued that adaptive menus aid user learning by simplifying menus. Also, where the menu changes are made based upon user behavior, a user's exact needs are anticipated and more completely fulfilled.

These menus may also create problems for users. If lower frequency items are incorrectly chosen for placement above the split or fold, high-frequency items placed below will take longer to perform. When items are relocated to different positions, a person's spatial memory is impacted. If users do not see the additional menu items, system learning may also be degraded. Some studies do show that users dislike the extra click or delay imposed by folded menus (Card, 1982; Somberg, 1987).

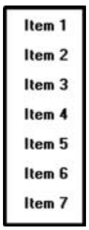


Figure 4.8: Traditional menu.

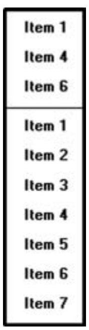


Figure 4.9: Split menu.

https://hemanthrajhemu.github.io

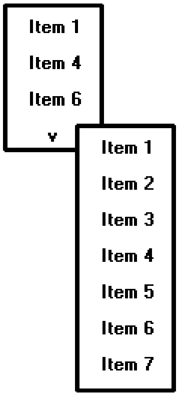


Figure 4.10: Folded menu.

To address some of the issues, Lee and Yoon (2004) conducted a study to determine when it was best to use various kinds of these menus. They evaluated traditional (static) menus, split menus, folded menus, and a fourth type they called *temporal*. A temporal menu is a traditional menu that first presents only the high-frequency items in their regular menu positions. After a short delay, the remaining lower-frequency items are filled in their normal positions as illustrated in Figure 4.11.

The study task was to select randomly selected items from menus containing seven choices. The menus were not adaptive in the sense that they were not reordered to reflect the choices being made. Two-thirds of the way through the test, however, the items were reordered to see the impact. The conclusion:

- Split menus had the fastest overall performance and were liked the best.
- For high-frequency items, split and folded menus were about equal.
- Performance on folded menus declined fastest as selection frequency went down (a wide range of functions being regularly used).
- After the item order switch, performance for both split and folded menus was poorer than that for the traditional and temporal menus.

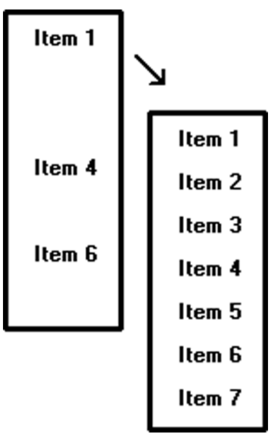


Figure 4.11: Temporal menu.

https://hemanthrajhemu.github.io

Based upon the results of this research Lee and Yoon developed a network to model the impact of selection frequency on selection times. The results, as presented in the preceding guidelines, are as follows:

- When a small, discrete set of functions is accessed 90 percent or more of the time, use folded menus.
- When a small set of items is selected between 31 percent and 89 percent of the time and the other items are selected with lower frequencies, use split menus.
- If there is no small, discrete set of items that is used 30 percent of the time or more, use traditional menus.

Although adaptivity is thought to be a desirable quality of a computer system, this appears to not be so for menu option ordering. Both split and folded menus in this study resulted in poorer performance after item reorganization. Another study compared static or fixed menus with dynamic menus whose options were continually reordered based upon the frequency in which they were chosen. Dynamic menus were slower to use and less preferred than static menus. The continual reordering interfered with menu order learning.

Phrasing the Menu

A menu must communicate to the user information about

- The nature and purpose of the menu itself.
- The nature and purpose of each presented choice.
- How the proper choice or choices may be selected.

Writing the content of menu components, the menu's title, the choice descriptions, and instructions, is often made difficult because of the varying experience levels of the menu users. At one extreme, there is the desire to explain, on the screen, everything in great detail. On the other hand, brevity is also important because of screen space constraints and limits on what people want to read. These conflicting goals often cause a trade-off between thoroughness and brevity. Also important in hierarchical menu systems is the role menus play in enabling a person to maintain a sense of place, or "Where am I now?" Also very important is a menu's ability to enable the user to accurately predict where a choice will lead, or what it will cause to happen, preventing user tedium and frustration. So, the menu content must be informative, but not intrusive. And it must balance the needs of all its expected users.

Following are guidelines for creating menu titles, choice descriptions, Web navigation links, and menu instructions. The standard graphical system conventions inscribed on menus, intent indicators, keyboard equivalents, and keyboard accelerators are also described.

https://hemanthrajhemu.github.io

Menu Titles

- Main menu:
  - — Create a short, simple, clear, and distinctive title, describing the purpose of the entire series of choices.
- Submenus:
  - — Submenu titles must be worded exactly the same as the menu choice previously selected to display them.
- General:
  - — Locate the title at the top of the listing of choices.
  - — Spell out the title fully using either an
    - Uppercase font.
    - Mixed-case font in the sentence or headline style.
  - — Superfluous titles may be omitted.

A meaningful menu title aids in defining the context of the menu and increases menu comprehension. An experimental study has demonstrated the value of titles to comprehension. Study participants were presented the detailed steps to perform a function. A descriptive title for the steps was (A) not included, (B) presented at the start of the steps, and (C) presented at the end of the steps. Participants given a title at the start of the steps (B), reported higher comprehension and recalled twice as many items as those who were not given a title (A), or who were presented the title at the end (C). The title established the context of the task, and knowing this context greatly aided comprehension.

Main menu. The menu title should immediately orient the viewer to the menu's content and purpose. It should be a short, clear, distinctive, and descriptive title, representing the entire series of choices. It's an important contextual and navigation component. A title such as MAIN MENU OPTIONS provides no information except that the user is probably at the top of a hierarchical menu tree.

Submenus. Submenu titles must be worded exactly the same way as the menu choice previously selected to display them. This will provide structural continuity and assure users that they are progressing as expected through a menu hierarchy.

General. Locate the title at the top of a listing of choices, in the title bar if one is available. Display the title in uppercase or in a mixed-case font using the sentence or headline style of presentation. Whichever style is chosen should be consistently followed for all menus. When using the headline style, capitalize the first letter of each significant title word. The case style chosen should be consistently used on all menus. Superfluous titles, titles that add nothing to the understanding of menu content and context, may be omitted. A pop-up menu requested during a text-editing task, for example, is displayed within the context of the task being performed. The presented choice descriptions by themselves (Copy, Font, and so on) provide the necessary context. Message windows do not need a title either; the text of the message provides the context.

https://hemanthrajhemu.github.io

Menu Choice Descriptions

- Create meaningful choice descriptions that are familiar, fully spelled out, concise, and distinctive.
- Descriptions may be single words, compound words, or multiple words or phrases.
  — Exception: Menu bar items should be a single word (if possible).
- Place the keyword first, usually a verb.
- Use the sentence or headline style.
- Use task-oriented not data-oriented wording.
- Use parallel construction.
- A menu choice must never have the same wording as its menu title.
- Identical choices on different menus should be worded identically.
- Choices should not be numbered.
  — Exception: If the listing is numeric in nature, graphic, or a list of varying items, it may be numbered.
- If menu options will be used in conjunction with a command language, the capitalization and syntax of the choices should be consistent with the command language.
- Word choices as commands to the computer.

Meaningful. Menu item descriptions should be composed of familiar and fully spelled out words. While abbreviations may occasionally be necessary, they should be kept to a minimum. If you are using an abbreviation, only use those that are standard or well known. Descriptions should also be concise, containing as few words as possible, and distinctive, constructed of words that make each choice clearly different from all others. Repeated use of the same word or words in multiple choice descriptions hinders distinctiveness and signals the necessity for creating a grouping whose title is based upon the repeated word.

Use high-imagery keywords, words that elicit a mental image of the object or action. Avoid low-imagery words that have more general connotations. For example, when obtaining a printout of a screen, the term “print” is much more descriptive than “list.”

In creating menu item descriptions, never assume that the description chosen by the designer will have the same meaning to the user. A study has found that the probability of two people choosing the same name or description for something ranged from 8 to 18 percent. Names chosen by experts were no better than those chosen by nonexperts. Therefore, iteratively test and refine the choices to achieve as much agreement in meaning by users as possible.

Size. Item descriptions may be single words, compound words, multiple words, or phrases. Menu bar items should be a single word, if possible. If a menu bar item must be a multiple word, visually tie the two words together by incorporating a hyphen between them. Web page content links will typically be phrases. Link writing guidelines are discussed in more detail in Step 8.

https://hemanthrajhemu.github.io

Keyword first. Arrange multi-item descriptions so that the descriptive and unique words appear at its beginning. This optimizes scanning and recognition while the user is learning the menu. Description phrasing and wording should also be consistent across all menus to aid learning further.

Capitalization. Use the headline style of presentation. Whichever style is chosen should be consistently followed for all menus.

Task-oriented wording. Task-oriented wording is preferable to data-oriented wording. Task-oriented wording usually positions a verb first, such as Manage Customer Information. An example of data-oriented wording would be to simply say Customers. What is being done with, for, or to customers is unclear in the latter.

Parallel construction. When choices are composed of phrases, use a parallel word construction in creating descriptions for related choices. Parallel construction would be: *Print* a File, *Execute* a Program, and *Eject* a Disk. An example of non-parallel construction is: *Print*; *Execute* a Program, and Disk *Eject*.

Relationship to title. A menu choice must never have the same wording as the title of the menu on which it is presented.

Consistency across menus. Identical choices on different menus should be worded the same.

Numbering. Items should not be numbered unless the listing is numeric in nature, graphic, or a list of varying items.

Command language. If menu options will be used in conjunction with a command language, the capitalization and syntax of the captions should be consistent with those of the command language.

Word as a command to computer. Phrase all menu choices as commands to the computer whenever possible. For example, say

Choose one:

Save and exit

Exit without saving

rather than

Do you want to save and exit?

Yes

No

Wording a choice as a command to the computer more clearly describes the action of what the command accomplishes. The Yes/No alternatives shown in the preceding example must be comprehended in conjunction with the question being asked. Wording a choice as a command also provides choice phrasing that is consistent with other system commands. A system, for example, often contains the standard commands Save and Exit. In addition, command wording enhances the learning of command mnemonics. Finally, this wording implies that the initiative is with the user in the dialog, not with the computer.

https://hemanthrajhemu.github.io

Menu Instructions

- For novice or inexperienced users, provide menu completion instructions.
  - — Place the instructions in a position just preceding the part, or parts, of the menu to which they apply.
  - — Present instructions in a mixed-case font in sentence style.
- For expert users, make these instructions easy to ignore by
  - — Presenting them in a consistent location.
  - — Displaying them in a unique type style and/or color.

People not familiar with a system and its menus may need guidance on how to complete a menu. Their needs may, however, have to be balanced against the needs of experienced users who may not want or desire such assistance. To satisfy the needs of all kinds of users at the same time necessitates that menu instructions be included on a menu, but that these instructions be easily ignored by those who do not need them.

Novice or inexperienced users. Provide explicit menu completion instructions for novice or inexperienced menu users. Place the instructions in a position just preceding the part, or parts, of the menu to which they apply. Present the instructions in a mixed-case, sentence-style font.

Expert users. When instructions are included on menus, they must be visually recognized as instructions. This will allow them to be easily ignored by the expert user when they are not needed, or no longer needed. Therefore, some visual aspect of the instruction must indicate that it is an instruction. As mentioned in Step 3 , designers of paper forms do this by presenting instructions in a different font or font style such as italics. The form user then immediately recognizes them as instructions, and they can be read or ignored as is desired.

To make instructions immediately recognizable as instructions on a menu, then, present them in a unique font or color. If one of these methods is used, however, cautions concerning the excessive use of different font styles (and colors, as shown in Step 12) must be heeded. Another, but less visually strong, technique is to identify the technique simply by its location. Begin the instruction to the left of the screen elements to which it applies, the left-justification identifying it as an instruction. Guidelines for writing text, including instructions, are discussed in Step 8.

Intent Indicators

- Cascade indicator:
  - — To indicate that selection of an item will lead to a submenu, place a triangle or right-pointing solid arrow following the choice.
  - — A cascade indicator must designate every cascaded menu.

https://hemanthrajhemu.github.io

- To a window indicator:
  - — For choices that result in displaying a window to collect more information, place an ellipsis (. . .) immediately following the choice.
  - — Exceptions — do not use when an action
    - Causes a warning window to be displayed.
    - May or may not lead to a window.
- Direct action items:
  - — For choices that directly perform an action, no special indicator should be placed on the menu.

Providing an indication of what will happen when a menu item is selected can enhance predictability and exploration of a graphical system. If a choice leads to another lower-level menu, include a *cascade indicator*, a right-pointing arrow, following the item description. If a choice leads *to a window*, include an ellipsis following the item description. Items causing a direct action will have no indicator. These intent indicators are illustrated in Figure 4.12.

IBM's SAA CUA designates choices leading to submenus or windows as *routing* choices, and items causing direct actions as action choices. A Microsoft Windows *intent indicator* simply implies that additional information is needed. This additional information request is usually presented in a window, but it need not necessarily be restricted to a window.

Keyboard Shortcuts

When a person is using a keyboard, it is inefficient to continually take mouse actions. Keyboard shortcuts allow actions to be taken using the keyboard as well as the mouse. In a study, keyboard shortcuts have been found to be significantly faster and more accurate than mouse clicks, and preferred by nearly all users (Jorgensen et al., 2002). Shortcuts include *keyboard equivalents* and *keyboard accelerators*.

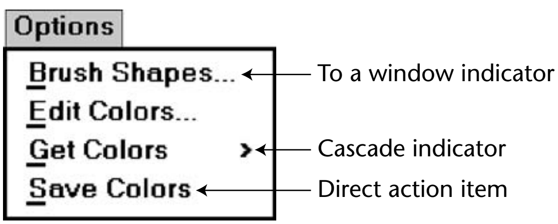


Figure 4.12: Intent indicators.

https://hemanthrajhemu.github.io

Keyboard Equivalents

- To facilitate keyboard selection of a menu choice, each menu item should be assigned a keyboard equivalent mnemonic.
- The mnemonic should be the first character of the menu item's description.
  - — If duplication exists in first characters, use another character in the duplicated item's description.
  - — Preferably choose the first succeeding consonant.
- Designate the mnemonic character by underlining it.
- Use industry-standard keyboard access equivalents when they exist.

Keyboard selection. The ability to select a menu alternative through the keyboard should always be provided. This is accomplished by providing a keyboard *equivalent* for each menu alternative.

Mnemonics. Keyboard equivalents that have meaningful associations with their corresponding choices will be more easily learned and remembered. Studies have found that simple truncation is a good method for creating mnemonics. Therefore, the first letter of the item description is the recommended mnemonic. Unfortunately, in following this method, duplications easily occur, so an alternative principle must also be provided. A simple scheme is to use the second consonant for duplicate items. This duplication-breaking scheme need not always be faithfully followed, however. Occasionally another letter in the menu item may be more meaningful to the user. In these cases, it should be selected.

Designation. Mnemonic codes can be visually indicated in a number of ways. The recommended method is an underline beneath the proper character within the choice. Other methods — a different character color, different character intensity, or a contrasting color bar through the relevant character — are visually more complex and should be avoided. Underlined keyboard equivalents are illustrated in Figure 4.13.

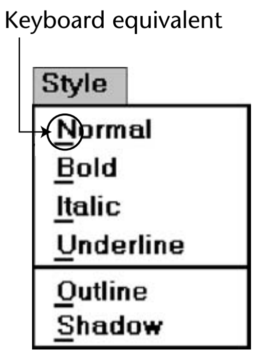


Figure 4.13: Keyboard equivalents.

https://hemanthrajhemu.github.io

Table 4.1: Standard Keyboard Equivalents

About	Help	Print	Send To
Apply	Help Topics	Print Preview	Show
Back	Insert	Properties	Size
Browse	Maximize	Redo	Split
Close	Minimize	Repeat	Stop
Copy	Move	Restore	Undo
Cut	New	Resume	View
Delete	Next	Retry	Yes
Edit	No	Rerun	
Exit	Open	Save	
File	Paste	Save As	
Find	Page Setup	Select All	

A great deal of commonality exists among these equivalents because they represent a wide variety of functions, many of which will rarely appear together on a single menu. If two actions with the same equivalents will be used within the same menu, one equivalent will have to be modified to make it unique.

Industry standards. Standard industry keyboard equivalents have been established for many common system menu choices. Where these standard equivalents have been established, they should be followed. Microsoft Windows calls keyboard equivalents *access keys*. Standard keyboard equivalents are shown in Table 4.1.

Keyboard Accelerators

- For frequently used items, provide a keyboard accelerator to facilitate keyboard selection.
- The accelerator may be one function key or a combination of keys.
  — Function key shortcuts are easier to learn than modifier plus letter shortcuts.
- Pressing no more than two keys simultaneously is preferred.
  — Do not exceed three simultaneous keystrokes.
- Use a plus (+) sign to indicate that two or more keys must be pressed at the same time.
- Accelerators should have some associative value to the item.
- Identify the keys by their actual key top engraving.
- If keyboard terminology differences exist, use
  — The most common keyboard terminology.
  — Terminology contained on the newest PCs.
- Separate the accelerator from the item description by three spaces.

https://hemanthrajhemu.github.io

- Right-align the key descriptions.
- Do not use accelerators for
  — Menu items that have cascaded menus.
  — Pop-up menus.
- Use industry-standard keyboard accelerators when they exist.

Accelerators are keys, or combinations of keys, that invoke an action regardless of cursor or pointer position. They are most commonly used to activate a menu item without opening the menu. They are most useful for frequent activities performed by experienced users. IBM's SAA CUA and Microsoft Windows call these keys *shortcut* keys. They may also be called *hot keys*. Many products have, within their guidelines, standard accelerator key recommendations as well as rules for creating new accelerator keys.

For frequently used items, assign a key, or combination of keys, to accomplish an action. Function key shortcuts are usually easier to learn than modifier plus letter shortcuts. Pressing no more than two keys simultaneously is preferred; three keystrokes is the maximum. Use a plus (+) sign to indicate on the screen menu that two or more keys must be pressed at the same time.

Accelerators should have some associative value to the item and be identified by their actual key-top engraving. In situations where multiple kinds of keyboards exist, and there are keyboard terminology differences, use the term most commonly found on the keyboards or use the term contained on the newest PC, if evolution to the new PCs is expected.

Display the accelerator right-aligned and enclosed in parentheses to the right of the choice. Incorporating these key names within parentheses indicates that they are prompts (which they actually are) and that they may be easily ignored when not being used. Most graphic systems do not place them within parentheses, giving them much too strong a visual emphasis. See Figure 4.14.

Do not use accelerators for menu items that lead to cascaded menus. Also, do not use accelerators on pop-up menus, because they are mouse driven. Use standard keyboard accelerators when they exist. Standard industry accelerators are shown in Table 4.2.

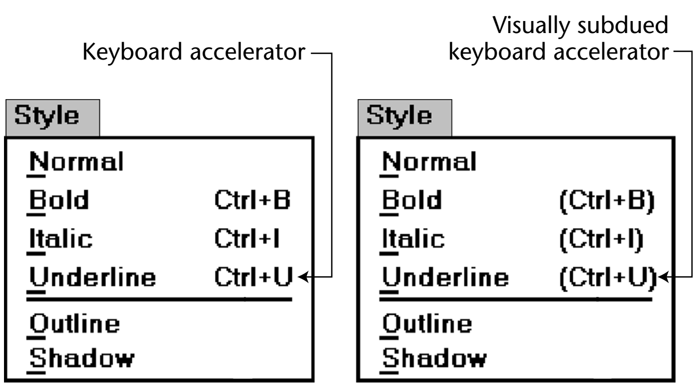


Figure 4.14: Keyboard accelerators.

https://hemanthrajhemu.github.io

Table 4.2: Standard Keyboard Accelerators

THIS ACCELERATOR:	DESIGNATES THIS ACTION:
Ctrl+C	Copy.
Ctrl+N	New.
Ctrl+O	Open.
Ctrl+P	Print.
Ctrl+S	Save.
Ctrl+V	Paste.
Ctrl+X	Cut.
Ctrl+Z	Undo.
F1	Display contextual help window.
Shift+F1	Activate context-sensitive help.
Shift+F10	Display pop-up menu.
Spacebar	Select (single mouse click).
Esc	Cancel.
Alt	Activate a menu bar.

Selecting Menu Choices

Menu items can be selected by pointing at the choice with a mechanical pointer, by pointing at the choice through the keyboard, or by keying a value designating the choice.

Initial Cursor Positioning

- If one option has a significantly higher probability of selection, position the cursor at that option.
- If repeating the previously selected option has the highest probability of occurrence, position the cursor at this option.
- If no option has a significantly higher probability of selection, position the cursor at the first option.

When a menu is first displayed, position the cursor at the option most likely to be chosen, or at the first option in the list if no option has a significantly higher probability of selection. If repeating the previously selected option has the highest probability of occurrence, position the cursor at this option.

https://hemanthrajhemu.github.io

Choice Selection

- Pointers:
  - — Select the choice by directly pointing at it with a mechanical device such as a mouse or trackball pointer, or light pen, or pointing with one's finger.
  - — Visually indicate
    - Which options can be selected.
    - When the option is directly under the pointer and can be selected.
  - — Visually distinguish single- and multiple-choice menu alternatives.
  - — If pointing with a mechanical device is the selection method used
    - The selectable target area should be at least twice the size of the active area of the pointing device or displayed pointer. In no case should it be less than 6 millimeters square.
    - Adequate separation must be provided between adjacent target areas.
  - — If finger pointing is the selection method used
    - The touch area must be a minimum of 20 to 30 millimeters square.
    - The touch area must encompass the entire caption plus one character around it.
- Keyboard:
  - — If moving the cursor to a menu choice
    - The up and down arrow keys should move the cursor up or down vertically oriented menu options.
    - The left and right cursor keys should move the cursor left or right between horizontally oriented menu options.
  - — If keying a choice identifier value within an entry field
    - Locate the entry field at the bottom of the last choice in the array of choices.
    - Uppercase, lowercase, and mixed -case typed entries should all be acceptable.
- Selection/execution:
  - — Provide separate actions for selecting and executing menu options.
  - — Indicate the selected choice through either
    - Highlighting it with a distinctive display technique.
    - Modifying the shape of the cursor.
  - — Permit unselecting choice before execution.
    - If a menu is multiple choice, permit all options to be selected before execution.
- Combining techniques:
  - — Permit alternative selection techniques, to provide flexibility.

Pointers. Items can be selected by being pointed at using a mechanical device such as a mouse, trackball, or light pen, or through touch pointing using one's finger (for touch-sensitive screens). Pressing a key, such as Transmit or Enter, or a mouse button, signals the choice to the computer. Always visually indicate in a distinctive manner which options are selectable and when the option is under the pointer and can be selected. Visually distinguish single- and multiple-choice arrays of menu choices.

https://hemanthrajhemu.github.io

An adequate pointing target area should be provided. This area should be at least twice the size of the active area of the displayed pointer of the pointing device. In no case should it be less than 6 millimeters square. To avoid unintended activation of the wrong option, provide adequate separation between selectable areas. Highlighting of the area when selected will also provide indication of an incorrect choice.

If finger pointing is the selection method used, an even larger touch area must be provided, a minimum of 20 to 30 millimeters. Single-character positions on a screen make poor targets for most fingers. Also, keep in mind that using a finger to signify a choice can be taxing on arm muscles, so this approach should only be used in casual or infrequent use situations.

Keyboard. If the user is moving the screen cursor to a menu choice, the up and down arrow keys should move the cursor up and down a vertical column of menu options. The left and right arrow keys should move the cursor left and right across a horizontal array of options. If the user is keying a choice identifier value within an entry field, locate the entry field at the bottom of the last choice in the array of choices. If the user is keying a mnemonic value, the entry should be acceptable in any case (upper, lower, and mixed).

Selection/execution. Provide separate actions for selecting and executing menu options. For example, require typing the mnemonic to select, and then a press of the Enter or Return key to execute. Or, with a mouse, require moving the pointer to the option to select, and then clicking to execute. A study (Chaparro et al., 2000) found that "pointing-and-clicking" rather than a "mouse-over" to open menus on cascading menus takes 18 percent less time, causes fewer errors, and is preferred by users. Always permit erroneous selections to be unselected and, in a multiple-choice menu, permit all options to be selected before execution.

The item selected should be highlighted in some way through a distinctive display technique such as bolding or changing its color. An alternative is to change the shape of the pointer itself. These methods provide direct visual feedback that the proper choice has been selected, reducing the probability of errors in choice selection.

Combining techniques. Permit alternative selection techniques to provide flexibility. If a pointing method is used, also provide a keyboard alternative to accomplish the same task. Pointing will probably be easier for the novice, but many experts prefer the keyboard alternative.

Defaults

- Provide a default whenever possible.
- Display as bold text.

Defaults aid system learning and enhance efficiency. Provide as many as possible. Indicate a default by displaying it in a bold text.

https://hemanthrajhemu.github.io

Unavailable Choices

- Unavailable choices should be dimmed or "grayed out."
- Do not add or remove items from a menu unless the user takes explicit action to add or remove them through the application.

Choices not available to the user should be made visually distinctive by dimming them or graying them out. They must not compete with active items for the user's attention. Items should not be added or removed from a menu unless the user takes explicit action to do so. Allowing the system to change menu items takes control away from the user and can also lead to user confusion.

Mark Toggles or Settings

- Purpose:
  — Use to designate that an item or feature is active or inactive over a relatively long period of time.
  — Use to provide a reminder that an item or feature is active or inactive.
- Guidelines:
  — Position the indicator directly to the left of the option.
  — For situations where several nonexclusive choices may be selected, consider including one alternative that deselects all the items and reverts the state to the "normal" condition.

Purpose. Mark toggles or settings, illustrated in Figure 4.15, are menu items that toggle between active and not active. When it is active, an indicator is displayed adjacent to the item description. For nonexclusive choices, a check mark is displayed; for mutually exclusive choices, another distinctive symbol, such as a diamond or circle, is displayed. When the item is not active, no mark or symbol will appear.

Examples of items using mark toggles are having a specific application automatically loaded after the system is loaded; having windows automatically reduced to icons when they are made inactive; or making a setting without requiring a dialog box. The purpose of mark toggles is to activate or deactivate an attribute by setting one menu item.

Advantages/disadvantages. Mark toggles provide a visual indication of the state of an item. They are accessed quickly, but may not always be visible. Mark toggles are best suited to items or features that remain active or inactive over relatively long periods of time. They provide good reminders of the state that exists.

Guidelines. Position the mark toggle indicator directly to the left of the menu option. In situations where several nonexclusive choices may be selected on one menu, consider including one alternative that deselects all the items and reverts the state to the normal condition, as illustrated by "Regular" in Figure 4.15.

https://hemanthrajhemu.github.io

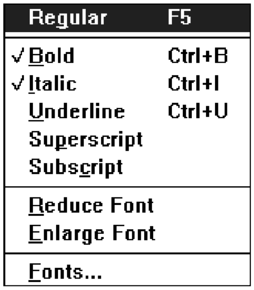


Figure 4.15: Mark toggles or settings.

Toggled Menu Items

- Purpose:
  — Use to designate two opposite commands that are accessed frequently.
  — Use when the menu item displayed will clearly indicate that the opposite condition currently exists.
- Guidelines:
  — Provide a meaningful, fully spelled-out description of the action.
  — Begin with a verb that unambiguously represents the outcome of the command.
  — Use mixed-case letters, with the first letter of each word capitalized.

Purpose. A toggled menu item is a one-menu item command that toggles back and forth between the current state and its alternative state. When the menu item is first displayed, it reflects the alternative state to the condition that currently exists. For example, in Figure 4.16, if a background grid is currently being displayed, the menu item reads *Hide Grid*. When Hide Grid is selected, the grid is removed from the window, and the menu item dynamically changes to reflect the opposite action. It will now read *Show Grid*. When a grid is again requested, it will change back to Hide Grid. The purpose of toggled menu item is to use a single menu item to designate and activate the one, opposite, alternative of a two-state command setting.

Advantages/disadvantages. Toggled menu items shorten menus, decrease visual clutter, provide quicker access, and foster faster comprehension of the command action. When they are located on a pull-down menu, however, the actions themselves are not always visible. The opposite action reflecting the current state of the attribute, because it too is not visible, can cause uncertainty for novice users concerning what the state actually is. Toggled menu items are also limited in use to commands only.

Guidelines. Use toggled menu items to designate two opposite commands that are accessed frequently. The menu item displayed must be one that clearly indicates that the opposite condition currently exists. The menu captions should clearly state what would happen if the menu item action were requested. It is most meaningful to begin the command with a verb.

https://hemanthrajhemu.github.io

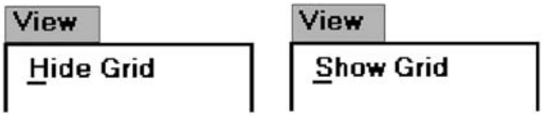


Figure 4.16: Toggled menu item.

Web Site Navigation

Navigation refers to the method people use to find what they want in a Web site. Navigation, and an efficient navigational structure, is the most important element in system usability. A simple and clear navigational structure is the backbone upon which all system features are draped. To navigate to a destination, people use available spatial and environmental information, a process called *wayfinding* (Lynch, 1960; Downs and Stea, 1973). Wayfinding involves four stages: *orientation*, *route decision*, *route monitoring*, and *destination recognition*.

Orientation. One's current location relative to nearby objects and the destination must be determined. This is called orientation. Dividing a space into small parts and providing landmarks and meaningful signage (titles) aids orientation. The result is that locations have identities that are easier to remember.

Route decision. A path must be chosen to get to the destination. Route decision-making is improved by minimizing the number of navigational choices, and by providing signs or prompts at decision points. People generally prefer shorter routes to longer routes, even if the shorter route is more complex. Routes are followed most efficiently when clear directions or signs are available. Site maps provide good mental representations of the space to be navigated, and are very useful if the space is large, complex, or poorly designed.

Route monitoring. The chosen route must be monitored to confirm that it is leading to the proper destination. Connect locations with paths that have clear beginnings, middles, and ends. A person should be able to gauge his or her progress as the path is followed. *Breadcrumbs*, a visible history of the path followed, aids monitoring, especially when a mistake has been made, and going back is necessary.

Destination recognition. The destination, when it is found, must be easily recognized. To aid recognition, provide clear and consistent identities to destinations.

In Web site design, the most successful sites have been found to be those that enhance wayfinding, possessing easy-to-use and understandable navigational systems. No amount of graphical "sparkle" has yet been able to overcome a poor navigational design. A Web site's organizational structure, its navigational tools and their obviousness, and the labels on its navigational elements influence a Web site's usability. The navigational elements of Web sites are referred to as *links*. Navigational elements that link usually consist of textual phrases or images. A person selects a link through clicking a device-based control, such as the mouse, on the link itself. A navigational link, therefore, is said to be *clickable*. A link, when selected or clicked, typically

https://hemanthrajhemu.github.io

causes a new page to be presented containing the content described by the link's label. A fundamental principle in Web page design is to clearly visually differentiate clickable links from other page elements.

Other design aspects also influence Web site usability. Among them are the navigational aids available, including indexes. In Web site navigation design, the unique, often incompatible, relationship the browser has to the Web site application being presented can also influence its navigational ease.

Web Site Navigation Problems

To fully understand what comprises good navigation, let's first look at some Web site navigational issues and problems, both technical and usage-related. The Web and its navigation is undoubtedly the most complex computer interface facing people today.

Technical issues. Unlike a graphical system application, whose screens tend to flow in an orderly and predictable manner, a Web site is composed of pages, each of which can, theoretically, be linked to any other page in the site. The graphical application user normally begins a process at a prescribed starting point and proceeds sequentially until a process or task is finished. Web users, on the other hand, can perform tasks or satisfy needs at will, easily moving between most pages in the "spider web" in any order desired, and even jumping to other spider webs when the urge arises. In an analogy to driving a car, the graphical system user is essentially following a freeway in Nevada. The Web site user is wandering around in downtown Boston without a road map and, encountering a road link (a bridge over the Charles River), suddenly finds himself in Cambridge.

The graphical system user must deal with only one operating system whose navigational characteristics are standard and fairly consistent. The Web user must confront two navigational systems, that of the browser being used and that of the Web site being viewed. A click of the browser Back button, for example, simply redisplays the page that was previously displayed. This page may have been in another Web site, and the user is transported there. Neither Web site is aware of this change. Nothing that might have been done on the page "moved back from" is changed. To move around within a Web site requires links within the Web site, either in the form of textual links or command buttons. The data must always be thought of as separate from the controls used to display it, not as being seamless, as occurs in graphical systems. Web users, especially novices, often do not recognize where the browser ends and the Web site begins.

Another problem: Because of the rapidly evolving and expanding nature of the Web, Web sites also have a tendency to grow and grow. As more and more is added, what may have been initially a reasonable structure and menu scheme slowly dissolves into a confusing mass of listings and linked pages. The result is unrelated information that is presented in no particular order.

Usage problems. The two most serious user problems in Web navigation are the heavy mental loads imposed to use the Web and the feeling of spatial disorientation that often occurs. This problem may also occur in hierarchically structured graphical systems. The *cognitive or mental overhead* the user must expend in making decisions concerning which links to follow, or to abandon, can be overwhelming.

https://hemanthrajhemu.github.io

Often, there are too many links presented on a page, many of whose meanings are not clear. Links frequently offer few clues to where they lead, how much information will be found at the other end, and how this information relates to the currently displayed page. For the user to reach a goal, each link's relevance to the task at hand must be determined. Another problem is that not all links on a page are always obvious. This often leads to trial-and-error behavior; the user aimlessly clicking to see what happens.

Feelings of *disorientation* are easily experienced when one becomes "lost in Web space." Studies have shown that most people do not seem to become familiar with the layout of sites or develop useful mental models of their structure (Spool et al., 1997). Many people also don't understand where they are in a Web site's information structure (Nielsen, 2000). A scrolling page can lead to loss of local context when the basic navigational elements, such as links to other local pages in the Web site, disappear. There are then no familiar landmarks to then navigate by. Long chains of links to reach relevant material can be tedious and also lead to loss of context, and a "Where am I?" reaction.

A research paper found that 39 percent of users of shopping sites failed in their buying attempts because the sites were too difficult to navigate. If people get buried in information, or lost on a side trip with no signposts or landmarks in sight, the most frequently implemented solution to the problem is to abandon the entire process.

Web Site Navigation Goals

As previously described, understanding a Web site's navigational scheme is made more difficult because Web sites usually have much less perceived structure than typical graphical system applications. Web pages can also be of any length and possess any number of links to any number of other pages. The user can wander at will or whim through multitudes of links, pages, and Web sites, and any meaningful structure a Web site design does possess can easily disappear from one's memory in the maze of directional twists and turns being made. The potential for getting lost is extremely high, unless numerous, obvious, and understandable landmarks are available as a guide.

MAXIM The time it takes to make a decision increases as the number of choices increases.

A well-designed navigation system facilitates quick and easy navigation between components whose structure and relationship are easily comprehendible. For the user, answers to the following questions must be obvious at all times during an interaction:

- Where am I now?
- Where did I come from?
- Where can I go from here?
- How can I get there quickly?

https://hemanthrajhemu.github.io

A good navigational scheme and features, and the proper navigational tools, will minimize, if not eliminate, the problems associated with cognitive or mental overload and feelings of disorientation.

Web Site Navigation Design

The section focuses specifically on Web site navigation design. It will review typical Web site organizational schemes; the navigational components of a site, including the browser command buttons, links, Web site toolbars, and Web site command buttons; and the characteristics and components of a Web site that contribute to maintaining a "sense of place." In designing a Web site navigation scheme there are two things to always remember. Never assume that users know as much about a site as the site designers do (this has been said before), and any page can be an entry point into the Web site.

Web Site Navigation Aids

- To aid Web site navigation and learning
  - — Provide a map or overview of the menu hierarchy.
  - — Provide clickability cues.
  - — Provide a "look ahead" at the next level of choices, alternatives that will be presented when a currently viewed choice is selected.
  - — Change the color of a link that has been clicked.
- Provide feedback concerning one's current location
  - — Provide navigation history.
  - — Match link text (or label) to the destination page heading.

Map or overview. As has been discussed, as one wanders deeper into a multilevel menu system, it is increasingly difficult to maintain a sense of position or orientation. The result is that getting lost in the maze is quite easy to do. The value of a map or overview in reducing disorientation has been demonstrated in some studies. In these studies, providing a graphic representation of a menu structure in map form, either in hard copy or online, resulted in fewer errors or wrong choices, faster navigation, and greater user satisfaction when compared to providing no guides or simply providing indexes or narrative descriptions of the menu structure. So, maps or graphic representations of the Web site structure are desirable and should be included on the Homepage. They should also be included in the Web site documentation (where available), and through a Help function.

Clickability cues. It should be obvious which items on a page are clickable. Provide a visual indication that an item or word on a page is clickable using techniques such as color, underlining, bullets, and arrows.

Look-aheads. Navigation and learning will be assisted if a person is able to browse the next level of choices before the currently displayed choice is selected. As the

https://hemanthrajhemu.github.io

cursor moves across a menu bar, for example, the pull-down menu may be automatically dropped, permitting review of the choices available if that menu bar item is selected. Such look-aheads are useful if ambiguity exists at higher-level choice points. They have been found to decrease errors and improve satisfaction. Menu search time may be longer, however.

Link color. Changing the color of a link that has been selected reminds a person that this destination has already been visited.

Navigation history. It has been found that being able to view, on the screen, the path one is following improves learning and performance, and reduces feelings of disorientation. Provide a navigation history that summarizes the menu choices made leading to the currently displayed menu or screen.

Link text. The importance of good Web site labeling has already been discussed. Matching link text and destination page heading will assure the user that the path being followed is correct.

Web Site Organization

- Divide content into logical fragments, units, or chunks.
- Establish a hierarchy of generality or importance.
- Structure the relationships among content fragments, units, or chunks.
  — Establish global or site-wide navigation requirements.
- Create a well-balanced hierarchical tree.
  — Restrict to two levels requiring no more than two clicks to reach deepest content, whenever possible.

It is easier to develop a clear and comprehendible navigation scheme if the Web site is organized and structured in a meaningful way. The design goal is a proper balance of menus and pages that can be easily and efficiently moved between.

Logical fragments, units, or chunks. Because of limitations in short-term human memory, smaller discrete fragments or chunks of information are often easier to navigate than long, undifferentiated units. The concept employed in Web site design, in reaction to this human memory frailty, is called *hypertext*. Hypertext is a nonlinear way of organizing information based upon the following principles:

- A large body of information exists that can be organized into fragments.
- The fragments relate to one another.
- The user needs only a small fraction of the fragments at any one time.

In organizing a Web site, information is first divided into logical fragments, units, or chunks. Coherent chunks that focus on a single topic is the desired goal. These small chunks of related information are easier to organize into the modular groups of information that will compose the organization scheme, and form the basis for hypertext links to be described shortly. A design-organizing aid, card sorting, is discussed in Step 2.

https://hemanthrajhemu.github.io

Hierarchy of generality or importance. Having identified the information units, information is now organized in according to importance or generality, from general to specific. A hierarchical tree is the most recommended organization scheme; Sun Microsystems (1998) suggests that whenever possible

- State conclusions and link to supporting details.
- Enumerate categories of information and link them to detailed listings.
- Summarize information and link to full-length treatments.

A document organizational tree structure, (table of contents, chapters, sections, and subsections) is a good scheme, because people are very familiar with, and have an excellent mental model of this organization. Such a structure provides information about information sequence, information quantity, and the relationships existing between components. Other organizational schemes include topics followed by subtopics, or prioritization from most to least important. The objective is to allow the user to scan the page and then select relevant and useful content for further review. Excessive fragmentation of a long, sequential story, however, should be avoided. Reading will be impeded and printing made more difficult.

Structure the relationships. Identify the relationships that exist between various elements in the hierarchical tree. In a large Web site, two levels of navigation will exist. The first is movement within the subject area. This navigation includes moving *within a branch*—up to a parent page or down to a child page. It also involves navigating across branches to sibling pages or other sections of a site. What points on other tree branches it will be beneficial to go directly to then, must also be established. The second navigation type is *global or site-wide*. What other site features, such as search a facility, site maps, and other major content areas should be mentioned on each page? Do not mention all features on all pages. Restrict the number presented to the several most useful features.

To unveil the Web site's structure, use *progressive disclosure*. Heading levels, shown in varying type sizes (as on paper), will also be helpful in aiding understanding of site organization.

Hierarchical tree. Web site pages should be organized as offshoots of a single homepage. If a site has a large number of information categories, and each category contains a lot of content, create submenus to aid navigation. The design goal: a well-balanced hierarchical tree that facilitates quick access to all information and also helps people understand how the site is organized. The so-called spoke design, where every page is linked to every other page, has been found to lead to lower usability.

Hierarchical breadth has been found by many research studies to be greatly preferable to hierarchy depth. A few menus with a larger number of choices are better than a large number of menus each with a smaller amount of choices. When menu levels go to four, five, or more, the chance of users becoming lost or disoriented is greatly increased. As studies have found (Zaphiris and Mtei, 1998; Larson and Czerwinski, 1998), restrict, whenever possible, the hierarchical tree to two levels requiring no more than two clicks to reach the deepest content. A two-level structure encompasses a homepage and two additional levels below it.

https://hemanthrajhemu.github.io

Navigation Page Design

- Use appropriate menu types.
  - — Sequential menus for simple forward-moving tasks.
  - — Simultaneous menus for tasks that would otherwise require extensive Back button use.
- Keep navigation-only pages short.
- Limit prose text.
- Scrolling:
  - — Never require scrolling of navigation-only pages.
  - — Minimize the need for scrolling to view all links on pages containing content.
  - — Never require horizontal scrolling.

A navigation page contains no content and is designed to direct or redirect users. It may take the form of a homepage, a site map, an overview, and so forth.

Appropriate menu types. Use *sequential* menus, menus arranged in a predetermined order, for simple forward-moving tasks. Use *simultaneous* menus, menus displayed together, if the use of sequential menus will require extensive use of the Back button (Hochheiser and Shneiderman, 2000). Simultaneous menus are usually presented in frames, a feature supported by most browsers. A frame allows the display area to be divided into two or more sections whose contents behave like independent Web pages.

Short pages. Confine navigation pages to one screen whenever possible. Nonvisible items may never be noticed.

Prose text. When navigation pages contain many words, readers tend to scan looking for specific words and clicking links rather than reading the text associated with links (Koyani, 2004.)

Scrolling. Never require scrolling of navigation-only pages. Besides being tedious, not being able to see all links at the same time makes comparison of the alternatives for selection purposes much more difficult. For scrollable content pages, minimize the need to scroll to see all links. Also, ensure that all related links on a screen are seen together to facilitate comparison. Never require horizontal scrolling. It makes text reading difficult and users dislike it.

Components of a Web Navigation System

To move between Web site information fragments necessitates the creation of many navigation *links*. They are contained within a framework of tools or controls, including the browser's command buttons, textual phrases, Web site navigation bars, and Web site command buttons. Collectively, these are all referred to as links. Links are one of the most discussed issues in Web site design.

A link functions as a menu choice that, when selected, results in the connected information being displayed, or results in a file being opened or downloaded. A movement

https://hemanthrajhemu.github.io

link may transport the user to another location within a page, to a new site page, or to another Web site. Originally, due to the nature of technology at the time hypertext was employed in computer systems, links only consisted of textual or binary files. Utilization of hypertext on the Web allowed links to be created using images as well as text, so the term *hypermedia* was coined to reflect this expanded nature.

In addition to being the critical component in Web navigation, links give the user an idea of what a Web site, document, or page is all about. The wording of a textual link should enable a person to predict what lies submerged below, or what will happen if it is activated. Descriptive links let the user determine whether a link should be followed or not. This is a complex cost-benefit calculation that the user makes many times in a Web interaction session.

Providing an extensive collection of link navigation tools will focus the user on the Web site itself and its content, drawing attention away from the general-purpose browser links. Making these tools consistent and predictable will help the user create an understandable mental model of the site and its organization. To begin, several general link guidelines are

- All navigation elements must
  - Make sense in the absence of site context.
  - Be continually available.
  - Be obvious and distinctive.
  - Be consistent in appearance, function, and ordering.
  - Possess a textual label or description.
  - Offer multiple navigation paths.

Sensible. All navigation controls, in the absence of site context, must make sense to the user. The user may have "lost" the context, or the page or Web site may have been entered from almost anywhere.

Available. All navigational controls must be easy to access. If they are not readily available, the full advantages of hypermedia may not be achieved.

Obvious and distinctive. A navigation link or control must look like a navigation control. Its appearance to the user must immediately suggest that it is an entity to be clicked or otherwise selected. This is accomplished through a control's appearance as well as its location. Non-obvious link or control choices lead to aimless and tedious page clicking and ultimately confusion and frustration. Conversely, do not make any other screen element look like a navigation tool if it is not one.

The obviousness of a link is called its *affordance*. A control with high affordance will be quickly identified as a control. Bailey (2000) in a study compared the link affordances of the homepages for two large Web sites. Each page contained 29 links. The link affordance rate for one site was 97 percent (participants, on the average, identified 28.2 page links). The rate for the other site was only 76 percent (the average link identification rate being 21.9 per page). This difference was statistically significant. Because of the non-obviousness of one-quarter of the poorer site's links, its users would have spent longer times searching for links, and would probably not have even discovered some links. Techniques to create the necessary affordance and distinctiveness differ depending upon the kind of

https://hemanthrajhemu.github.io

link. Guidelines enabling the various controls to achieve distinctiveness are described in the following control-specific sections.

Consistent. Like all elements of the interface, navigation links, toolbars, and command buttons must be consistent in appearance and behavior.

Textual. All navigation must have a textual label or description. Navigation using textual descriptions is much preferable to graphical-only navigation because the purpose and function of graphic images are often unclear. They also take longer to download. Textual links are also necessary for users who do not have graphics, or who have chosen not to display graphics.

Provide multiple navigation paths. Offer multiple paths or ways to move around the Web. Provide structural components such as site maps, a table of contents, and indexes to go directly to a point of interest, provide content links to move around nonsequentially, and provide command buttons, such as Next and Previous, to move sequentially.

Navigation Elements

- Differentiate and group navigation elements.
  - — Provide a global navigation bar at the top of each page.
  - — Provide a local category or topical links navigation bar on the left side of a page.
    - For long lists, consider placing within a frame.
  - — Optionally, provide a secondary navigation column on the right side of the page.
  - — Provide explicit or embedded textual links within the contents area.
    - Consider duplicating embedded links in the left side navigation bar.
  - — Place minor illustrative, parenthetical, or footnote links at the end of the page.
  - — For long pages provide
    - "List of Content" Links.
    - Important global or local links in a navigation bar repeated at the page bottom.
- Create a common and consistent theme.
- Never create pages without navigational options.

A Web site contains at least three levels of navigation elements: *global* or *site-wide*; *local* and *specific*; and *minor* or *footnote*. Clearly differentiate these navigation elements from one another and locate them consistently from page to page. The recommended structure, illustrated in Figure 4.17, separates these navigation elements from content, making it easy for users to find each. People using Web sites are now becoming accustomed to finding important navigation elements at the page top and in panels on the left and right side.

Two studies looking at user performance in using, and preferences for, the location of Web site navigational options have been described by Bailey (2006). Alternatives reviewed included options located at the page top [T], down the left side [L], and down the right side [R]. The study researchers, Kingsburg and Andre (2004), began by surveying existing Web site structures. They reported that the most common navigational structures were: (1) An initial selection is made from a page top navigation panel followed by a selection, and subsequent selections, from the left navigation panel (TLL), and (2) initial and subsequent selections are all made from the left panel (LLL).

https://hemanthrajhemu.github.io

In a scenario requiring three navigational selections, their first study evaluated top and left panels only. In general, they found

- Navigation was faster
  - When the first and second selection was made from the left panel.
  - When the panel used for the first selection (top or left) was split or separated from the panels used for the second and third selections.
- People preferred
  - The first selection be made from the left panel.
  - The first and second selections both be made from same panel, top or left.
  - The second and third, or first, second and third selections be made from the same panel.
- The best performing and most preferred structures were
  - Left-left-left (LLL).
  - Left-top-top (LTT).

Next, Kingsburg and Andre added a right-side panel in a similarly structured study. They found

- Navigation was faster
  - When the first selection was made from the left panel (not top or right).
  - When all selections were made from the same panel.
- People preferred
  - When all selections were made from the same panel.

Among this study's findings were that a right panel is a viable design option. Overall conclusions were

- Selection limited to either the left or right panels resulted in best performance and was preferred by users.
- Performance-wise, it is better to start in the left, not the right panel.

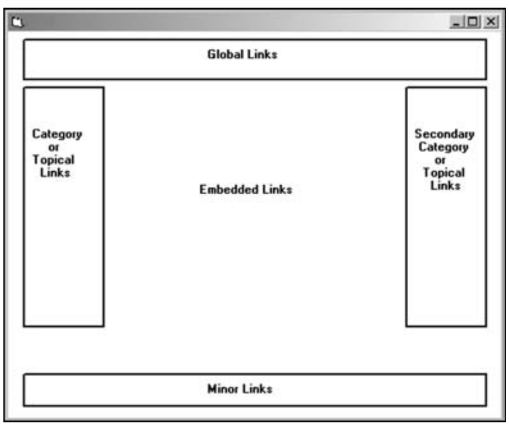


Figure 4.17: Web navigation component locations.

https://hemanthrajhemu.github.io

The navigation structures yielding slower performance and lower preference ratings were Top-Top-Top (TTT), Top-Left-Top (TLT), and Right-Top-Right (RTR). Perhaps these alternatives came out poorly because of less efficient scanning organization (TTT), excessive eye and pointer movement (TLT, RTR), or backward eye flow (RTR).

Another recent study by Oulasvirta et al. (2004) found that for people reading left-to-right languages, the tendency is to look to the left for the navigation panel.

Global. Global or site-wide navigation elements provide access to the site's total scope or categories of available information. An evolving standard in design is to locate the global navigation elements horizontally at a page's top. Locating the global links at the page top makes sense if one considers the logical flow of information through a screen. A selection from this global area eventually results in display of a page and its content, a top-to-bottom sequential eye flow. In the eye-tracking study reported by Nielsen (2006) in Step 3, a user's first search of a Web page horizontally across the page top (top bar in the F) may reflect an expectancy that important navigational elements are across the page top.

Category or topical. Local, specific and contextual navigation elements within the category or topical area being presented are typically displayed in a columnar array down the left page side. For long lists consider placing the links within a frame navigation panel. A study found users preferred non-scrollable frames rather than having the links move off as a page is scrolled (Bernard et al. 2001d). A second listing of links can also be presented in a column on the right side. Again, in the eye-tracking study reported by Nielsen (2006) in Step 3, a user's early vertical search of a Web page's left side (vertical bar in the F) may reflect an expectancy that important navigational elements are also along the left side of the page.

Embedded links. Phrases or embedded links will be provided within the contents area of a Web page. An *embedded* link is one found in the middle of prose or continuous text. Embedded links are frequently used to lead to supporting information or provide definitions of terms. They are designate by an underline and a unique color. Because users preferred redundant links, consider duplicating embedded links in the left side navigation bar (Bernard et al., 2001d).

Minor. Minor illustrative, parenthetical, or footnote links can be arrayed horizontally at the page bottom.

List of Content. For long pages with sections that are not visible without page scrolling include a set of links to each page section at the top of the page. These "anchor" or "within page" links provide a reminder of the page's contents, a page outline that can easily be reviewed, and a quick way to navigate to desired sections. These links also assist people in getting to a specific section if they arrive from a different page.

Important links. For long scrolling pages, repeat important global or local links at the page bottom. When finishing a page, the user, then, will not have to scroll upward to locate important navigation links.

Common theme. A common and consistent Web site navigation theme will enable people to more easily understand and learn its structure. Incorporate different

https://hemanthrajhemu.github.io

styles for these different navigation elements to aid people in understanding the differences in their meaning and function.

Always present options. All pages must have navigation options. Never create pages without navigation options. Many Web pages contain links opening a new browser window, thereby disabling the browser Back button. If the new window opens in a full screen people may not realize that they have been directed to another window and they may not know what to do. Links that do not behave as expected inhibit a person's understanding of a system. If such links are incorporated within a Web site, always include a prominent action control on the new window to close it and return the user to the original window.

Other Web Site Navigation Elements

In addition to Navigation bars, many other Web site elements are important components of the Web navigation system. Among these are overviews, including executive summaries, site maps, indexes, and tables of contents. Other elements are historical trails and search engines.

Overviews

- Provide
  - — An executive summary that provides a preview of the site and contains links to all major concepts.
  - — A site map illustrating the site's hierarchical structure and the relationships of components.
  - — Both global and local maps.
  - — An alphabetized site index.
  - — A table of contents.
- Allow accessibility from any point in the Web site.

Overviews provide a top-level view of a site's organization and content. Having an understanding of how a site is organized, the landmarks available within it, and the content it contains, assists the navigation process. In driving an automobile, referring to a road map before embarking on trip usually results in reaching one's destination faster, easier recovery from inadvertent wrong turns, a better ability to handle any unexpected detours that may be encountered, and a less stressful trip.

Overviews are most useful if provided in several forms. They may be needed during a Web interaction as well as before starting into a site. A graphical system help function, for example, may be available in tutorial form, be accessible by topics, or be organized in alphabetic form for easy scanning. It is difficult to predict the user's exact need at any moment in a session.

An *executive summary* will provide an overview of the site in narrative form and contain links to all major concepts. For large Web sites, a *site map* can be used to illustrate the site's hierarchical structure in either graphical or textual form. These elements provide a prospective on one's position in the spatial hierarchy. A good site map helps

https://hemanthrajhemu.github.io

facilitate site learning and should encourage comprehensive exploration of a site. Maps may be made available at both a global or local level within the site, depending upon the site's complexity. They can be designed to resemble a traditional table of contents or a simple index. An *alphabetized site index* will permit quick access through keywords and specific topics. A *table of contents*, structured as in a printed book, will permit review of major topics and the subtopics within. Because, based upon user studies, site maps are not always obvious and easy to find, a clear link saying *Site Map* should be placed on a consistent location on every page. A site map should be presented in one page, if possible. Do not exceed a couple of screenfuls, however. All of a Web site's overview elements should be accessible from any point within the site.

Historical Trails

- Provide
  - — Breadcrumb trails.
    - Locate at the top of the page below the navigation links.
  - — History lists.
  - — History trees.
  - — Footprints.
  - — Bookmarks.

Historical navigation aids try to show the user's position in an information space by showing where they have come from, or where they have been. Seeing a navigation path is thought to enable a user to better understand the context of the currently displayed page. Displayed paths also provide a means to easily return to places of interest.

Breadcrumb trail. A *breadcrumb trail* in a hierarchical Web site structure is a sequential textual listing of pages traversed from the parent page to the page currently being displayed. A trail, illustrated in Figure 4.18, is also a series of links that permit the user to go back to any page in the sequence with one click. Breadcrumb trails normally appear near the top of a page.

Studies have shown that spontaneous breadcrumb usage is not high in Web site navigation. The question being asked is, Why? Is it because they are not noticeable on a page, is it because people don't know what they are, or is because people don't care about them? Studies exploring their value have had mixed results. Some studies have found more efficient navigation and/or improved user satisfaction (Bowler et al., 2001; Maldonado and Resnick, 2002; Hull et al., 2004). Another found no navigation benefits (Lida et al., 2003). So, the jury remains out on their utility. Perhaps, like scrolling, people will learn to use breadcrumbs. Because they do provide some users some benefits now, and perhaps usage will increase with user experience, their inclusion on a page still appears beneficial.

At this moment no standard exists for how to separate the page names in a trail. Symbols used include an arrow (->), a colon (:), a greater than sign (>), and a slash (/). Until a standard evolves, any of the above symbols remain acceptable. Do not use anything else, however. Position a breadcrumb trail at the top of the page below any existing navigation links.

https://hemanthrajhemu.github.io

useit.com → Papers and Essays → Heuristic Evaluation → List of Heuristics

IBM developerWorks : Web architecture : Web architecture articles

Weather > Pacific Rim > Australia > **Sydney**

Figure 4.18: Breadcrumb trails.

Other historical trail elements. A *history list* is a sequential textual listing of sites or pages visited over a specific time period, a session, a day, or some other time period. A *history tree* is an overview map of a site's structure with pages already visited marked by an indicator such as a plus sign, check mark, or asterisk. The markings serve as *footprints*, guiding the user back to pages of interest, and/or signaling which have already been seen and may no longer be of interest. A *bookmark* is similar to a history list except that it is designated by the user to mark locations of continuing interest.

Search Facility

- Provide a search facility.

Another form of navigation support is provided by a site search facility. Provide such a facility within larger sites. Search facilities were addressed in Step 3.

MYTH **Real users don't mind complex navigation.**

Links and Actions

A Web page consists of a collection of links, both textual and graphic in nature, and a sprinkling of toolbars and command buttons. Links are commonly used to go to information, usually on another page. Commands and toolbars are commonly used to perform actions. How should links be presented to make them obvious? What kinds of links should be included on a page? How many links should exist on a page? These and similar questions are addressed next.

Kinds of Links

Typically, three kinds of links are provided for Web sites: internal, anchor, and external.

Internal links provide navigation within a Web site, permitting the user to freely move about between a site's pages. *Anchor* links, also called *associative* or *within-page links*, are used when a page is exceptionally long. A page contents list is presented at the top of the page with links to the corresponding information or section within the page. When clicked, the corresponding section is then displayed. *External* links point to new pages on other Web sites.

Navigational elements consist of textual phrases, images, and command buttons.

https://hemanthrajhemu.github.io

Textual Phrase Links

- Provide a mix of textual phrase links
  — In explicit menus.
  — Embedded within page text.

Textual phrases are words, or short pieces of highlighted text, serving as links. Text links are the preferred style of link in Web page design because they are more easily recognizable as clickable, download faster, are more understandable than images, and are preferred by users. They can also be easily modified visually to indicate that they have already been clicked. Another advantage is that using text links enables people with text-only and deactivated graphical browsers to see the navigation options.

Textual phrase links possess two distinct structures: explicit and embedded. An *explicit* menu is a listing of textual phrase links set apart from the main page content, often in toolbars or panels. These listings usually include links to various Web site topics, links to site global features such as the site map or search facility, and perhaps links to other related sites. These listings closely resemble typical screen menu arrays in their structure and presentation. A typical explicit menu is shown in Figure 4.19. An *embedded* menu is a link contained within the textual content of a page. Certain words or phrases are designated as links, highlighted, and when selected display the linked component for the user. An embedded menu is illustrated in Figure 4.20. Web sites usually contain both explicit link listings and embedded links in various mixes.

Lai and Waugh (1995) studied the effect of explicit listing hierarchical menus and embedded menus on a search task. They compared the three menu structures: (1) menus composed solely of explicit listings, (2) solely embedded menus, and (3) mixed explicit and embedded menus. They found that the best menu structure was determined by the kind of search task performed. Explicit listings worked better for straightforward search tasks, whereas menus containing embedded links worked best for complex and not fully known searches. The embedded menus improved search accuracy, but not search efficiency.

Bernard and Hull (2002) found no reliable performance differences between embedded links and explicit links. Users, however, preferred to have them embedded. Bernard et al., (2001d) found that redundant links were preferred by users, that is, duplicate links that were both embedded and outside the text on the left margin. Providing a mix of explicit listings and embedded menus in Web site design will best satisfy the needs of all site users.

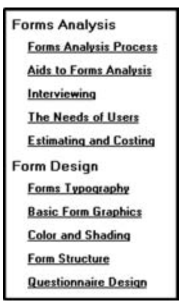


Figure 4.19: Textual explicit links.

https://hemanthrajhemu.github.io

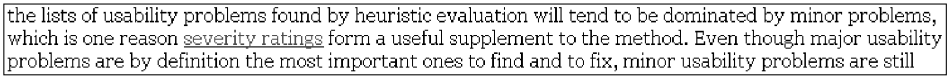

the lists of usability problems found by heuristic evaluation will tend to be dominated by minor problems, which is one reason severity ratings form a useful supplement to the method. Even though major usability problems are by definition the most important ones to find and to fix, minor usability problems are still

Figure 4.20: Textual embedded link.

Image Links

Graphical images or icons may appear in an array in the form of a navigation bar, as illustrated in Figure 4.21, or be individually located at relevant points within a page. As just described, text links are more easily recognizable than images, even if the images contain textual phrases. In one study people showed considerable confusion concerning whether or not certain page images were clickable (Koyani et al., 2004). People could not tell whether the images were true links without placing the pointer over the images. This is a slow process. Guidelines for creating and displaying graphics and icons are discussed in Step 11.

Command Buttons and Toolbars

Command buttons and toolbars, used to perform actions, may appear in an array in the form of a navigation bar, or be individually located at relevant points in a page. The advantage of standard Windows-type graphical command buttons are that many people commonly recognize them as clickable elements. This may not be true for unique and more stylized buttons, however. Command buttons and toolbars should never be used to retrieve or show information. Always use links for this purpose. Guidelines for creating and displaying command buttons and toolbars are discussed in Step 7.

Figure 4.21: Graphical or iconic navigation bars.

Figure 4.22: Command button navigation bar.

https://hemanthrajhemu.github.io

Presenting Links

- Provide consistent clickability cues.
  - — Avoid misleading cues to click.
- Use Textual links.
  - — Underline all link text, including that
    - Embedded in page content.
    - Contained in explicit menu listings.
    - Contained in headings.
    - Used as graphical labels.
  - — Exceptions:
    - Links on navigation-only menus and in lists do not necessarily need underlining.
  - — Do not underline any text that is not a link.
  - — Distinguish between unselected/unvisited links and selected/visited links.
  - — Make unselected /unvisited links blue.
    - Make selected/visited links purple.
    - Never show other text in the chosen selected/unselected colors.
- Distinguish internal, external, and anchor links.
  - — Identify external links by
    - Including destination URL address below the link.
    - Including an "exit disclaimer" adjacent to the link.
    - Providing an interim page after clicking an external link.
  - — Identify an anchor link with a "Page Contents" heading.
- Graphical links:
  - — Clearly identify clickable regions of images.
  - — Distinguish graphical links from decorative graphics through underlining graphical text labels.
- Links in toolbars:
  - — Distinguish links contained in toolbars through:
    - Presenting in consistent locations.
    - Using different colored backgrounds.
- Fat links:
  - — Consider fat links, if appropriate.

Links must be easy to find. They must not be confused with other screen graphics or textual content. Having to search for links can be a tedious and frustrating process. Whether a link has been navigated before must also be obvious. When looking for something new, continually embarking down a path already traveled can also be frustrating.

Clickability cues. It should be obvious which items on a page are clickable. Provide a visual indication that an item or word on a page is clickable using techniques such as color, underlining, bullets, and arrows. People should not have to move the cursor around a Web site to determine what is clickable. Scanning a page with a cursor is called *minesweeping*. Scanning with one's eyes is much faster than minesweeping. Be consistent in the use of conventions and techniques to indicate

https://hemanthrajhemu.github.io

links. Use of a particular symbol should (1) always indicate clickability, or (2) never indicate clickability. Otherwise the user will be confused and a mental model of the system will be more difficult to learn.

Underline text links. To identify a link, the well-established convention is to underline the link text. All link text must be underlined, including that embedded in page content, that presented in explicit listings, that contained in headings, and that taking the form of labels in graphical images.

If design clearly indicates a page area's navigational function, underlining may be safely eliminated. Consider, however, displaying an underline under non-underlined links when the pointer is placed over the text. This will reinforce the element's clickability.

Designate used links. Unselected or unvisited links must be distinguishable from selected or visited links. The ability to understand what links have been followed is one of the few standard navigational aids available in browsers. In a study, providing this type of feedback was the only variable that increased a person's speed in finding information (Koyani et al., (2004). Stick with the default colors of blue for links already followed and purple for links not yet ventured down. While the choice of blue as a text color was poor because of its degraded reading ability, it is now well learned. Van Schaik and Ling (2003) found that blue links are easier to click than black ones, even though black ones have higher visual contrast and are easier to see. Blue's use is recommended because it is now very familiar. Using nonstandard link colors can lead to problems. It is difficult to remember which color means what, thereby increasing link selection errors. It can also lead to confusion with normal underlined text in a document. If on a Web site multiple links lead to the same destination, change all links to the "visited" color. Finally, to avoid viewer confusion, never present other page components in a link's chosen selected/unselected colors.

Distinguish kinds of links. Visually distinguish links leading to different Web destinations. A study found that people often assume that a link will take them to another page in the Web site where they are browsing (Koyani et al., 2004). If the link simply moves within a page, or goes to another Web site entirely, confusion can result. A link that moves to another Web site can also be aggravating if the person was not ready to move on. A link's destination should be as predictable as the content at the other end.

Koyani et al. recommend the following methods for notifying a user that a link will leave the current Web site. Include the *URL address* of the destination below the link text as shown in Figure 4.23. Provide an *Exit disclaimer* command button adjacent to the link as shown in Figure 4.24. Provide an *interim page* after clicking the external link and before going to the new Web site as illustrated in Figure 4.25. Whichever technique is chosen, it should be followed consistently within a Web site.

Designation of these different destination types can also be accomplished by grouping links by type, giving them a descriptive heading, and placing them in unique and consistent locations on a page. Anchor links, for example, can be given a heading such as "Page Contents" as illustrated in Figure 4.26. A destination convention has yet to be established in Web site design. When one is established, it should be applied.

https://hemanthrajhemu.github.io

Figure 4.23: Destination URL address.

Figure 4.24: Web site exit disclaimer.

Figure 4.25: Web site interim page.

Figure 4.26: Anchor links table of contents.

Graphical links. Clearly identify clickable regions of images, either the entire image or the relevant sections of the image. Users should not have to use the mouse pointer to locate the clickable area or areas. If care is not exercised, graphical links may be also confused with decorative graphics. A principle of graphical or icon design is to always provide a text label (see Step 11). Distinguish graphical links by underlining the graphical text labels, as is done with plain link text.

Links in toolbars. Distinguish links contained in toolbars from page content by presenting the toolbars in consistent locations, and/or displaying them in backgrounds of a contrasting color to the page content. Global toolbars, most often in iconic or button form, are becoming consistently arrayed across the top of a page. Category or topical toolbars are now commonly arrayed down the left side or right side of the page. Toolbars containing textual listings that are similar to page content can be emphasized and differentiated through presenting them with a background and style that contrasts with the content.

https://hemanthrajhemu.github.io

Table 4.3: Links to Avoid (or Links that Aggravate the User)

Orphan Link	A link leading to a page that does not possess any navigation options.
Boomerang Link	A links that returns to the exact same spot.
Gotcha Link	A link that leads to little or no content.
False Alarm Link	A warning to not follow a link you really should follow.
Mystery Link	A link that does not look like a link because it is not properly labeled or does not possess a raised appearance.
Link-mania	Linking every time the same keyword is mentioned in a page.
Link-drunk	A long succession of links that must be followed to reach the destination.
Stairmaster Links	No Next link in a series of pages, necessitating continual return to a table of contents.
Gratuitous Link	A link to other sites to return a favor.
Missed Opportunities	For useful links.

Fat links. Links pointing to more than one page are called fat links. Because the browsers Firefox and Safari support tabbed browsing, it is possible to have a link open up into multiple tabs. Several destinations can then be accessed at once.

Some kinds of links to avoid are summarized in Table 4.3.

Types of Links

- Internal links within a page:
  - For long pages, include anchor links to internal page content.
- Internal links within a Web site:
  - On all pages include links to
    - The Web site homepage.
    - Global Web site features.
    - Other main pages, navigation points, or categories.
    - The likely Web site starting point.
    - Main pages with links to the displayed page.
  - On sequential pages, include links to the
    - Next page.
    - Previous page.
  - Also consider including links to
    - Places of related interest.
    - Important pages.
    - Background or explanatory information.
    - Supplemental information.

https://hemanthrajhemu.github.io

  - New or changed content.
  - Web site Quit or Exit.
- — Repeat important links.

■ External links:
- — Most appropriate for informational sites.
- — Provide links to relevant information on other Web sites.
  - Related content.
  - Reference information.
  - Background reading.
- — Place external links on a separate page.
- — Provide an indication when a link goes outside the current site.

Internally within a page. For long Web site pages, include *anchor* links for important content within the page. Place these links at the top of the page and identify them by a heading as anchor links.

Internally *within a Web site*, on all pages include the following links:

Homepage. A home link will transport the user directly to the site's homepage, a stable and safe starting point to escape to in times of difficulty. Easy access is also achieved when the user is ready to start over, or ready to commence a new navigation. A home link eliminates the necessity for sequential backward movement up through a series of pages.

Global features. Provide links to a site's global features, including the highest level of information categories and utilities such as the Search facility.

Other main pages, navigation points, sections, or categories. Do not link to all sections of the site from all pages; to do so will be overwhelming. To provide easy navigation throughout a site, provide links to a site's major navigation points, sections, or categories of information. Pages linked to must however, possess substantive content.

MYTH **Why do users need a road map of a Web site? They'll know where to go.**

The likely Web site starting point. Provide links to the site's likely starting point, the homepage, a site map, or an index.

Main pages with links to page. Provide links back to the main pages that have links to the displayed page. A return link describing the page one is going back to provides better predictability and much clearer context. It also provides escapability. While the browser's Back button will accomplish the same thing, it does not say what it is going back too, in case the user has forgotten where arrival was from. It also keeps navigation within the application itself (as opposed to the browser).

https://hemanthrajhemu.github.io

For *sequential pages*, provide easily accessible links to adjacent pages.

Next. To allow sequential movement downward through pages, place a Next link at the end of each page. Explain, whenever possible, what will happen or where one will go when the link is selected. Without this link, the user will have to continually refer to a table of contents or menu listing to continue navigation. This link will also allow users, should they choose, to leaf through the site as they would a printed book.

Previous. Also include a Previous link returning the user to the prior page in the Web site structure, thereby reversing direction in screen navigation. The browser Back button will only return the user to the last page viewed. This will facilitate movement through a site for those entering from another Web site into the page. Leafing backward through the page hierarchy will also be easy. Locate this link at the end of the page. For long pages, also include a Previous link at the page top.

Also consider including links such as the following:

Places of related interest. Provide links to other pages with related content. Wherever the user's attention is likely to be captured, provide a direct link to related places. Also, during a search, especially when using a search facility, people rarely land directly on the desired page. Often, however, they get close. Provide links to the answers they are most likely looking for.

Important pages. Provide links to important or high-priority areas or pages you want to make sure the user sees.

Background or explanatory information. Provide links to background or explanatory information to aid users who do not have the necessary knowledge to understand or use the page. Every page must be considered independent, and its content must be understood based upon the assumption that the user has seen no other related pages. New or infrequent users may require clarification of technical concepts, a dictionary, a glossary, and an expanded discussion of important content.

Supplemental information. Use links to provide supplemental information like definitions of terms and abbreviations.

New or changed content. Draw attention to new or changed content by making it easy to notice and go directly to. A prominently placed *What's New?* link can be used for this purpose.

Quit or Exit. The Web has no way to stop running an application without closing the browser or leaving by a link. Non-Web platforms have clearly marked exit paths, including a Quit or Exit on the File Menu. Provide this command so the users can confirm that an application is finished and any entered data should be saved. This command may be included on a special exit page showing external links and other useful information.

Repeat important links. Create links to satisfy disparate user goals. Redundant links (different links to the same page) may be useful in satisfying these varied needs. Make important or critical content accessible through more than one link. Establishing multiple ways to access the same information allows users who may approach a problem from a different direction or mindset to be able to find the information.

https://hemanthrajhemu.github.io

Provide *external* links to other relevant Web sites and information sources. There is some evidence that the inclusion of outbound links increases a site's credibility. It indicates that the site authors have done their homework, and they are not afraid to let users visit other sites.

Informational sites. Links to external or foreign sites are most appropriate for informational sites, where browsing is a primary usage purpose. In applications, where a task must be completed, focusing on the task is the most important aspect of design.

Related content. Provide links to relevant information on other Web sites, including sites with similar content to that mentioned in the site being browsed. Also provide links to other resources, repositories, reference information, and background reading.

Separate page. Links that go outside of the Web site can be placed on a separate page. To accomplish this, use a *See Also* link to this additional page. Placing these links on a separate page will not disrupt the flow of the displayed pages, and will not tempt people to leave the site before they have adequately reviewed it.

Outside indication. Identify links leading away from the site by a heading or another unique designation. Also inform users that they are leaving the displayed site for another Web site (see previous discussion).

Writing Link Labels

- Create meaningful labels
  — Containing action words.
  — Positioning keywords at the beginning.
  — Long enough to be understood.
  — Clearly indicating the link destination or resulting action.
  — Matching link name with its destination page.
- Ensure that embedded links are descriptive.
- Provide glosses or link labels to assist link understanding.

Link labels must be descriptive, differentiable, and predictive. The aforementioned brief guidelines are thoroughly discussed in Step 8.

Number of Links

- Every page should contain at least one link.
- Be conservative in the total number of links presented on a screen.
  — Restrict embedded links to those most important, pertinent, and interesting.
  — Place less relevant links in a listing.

One link. At a minimum, every page should contain at least one link. To follow a path and then reach a dead end is frustrating. Also, a dead-end page, if accessed from another Web site, provides no means for the user to navigate to other site pages.

https://hemanthrajhemu.github.io

Conservative number. How many links presented on a page is ultimately determined by the complexity of the site and its content. Menu research indicates that without choice groupings, links should be limited to four to eight, with groupings, limited to 18 to 24. Some experts in Web design recommend even fewer, a maximum of 8 to 12 links. In general, the more links contained on a page, the more decisions concerning which link to follow are imposed on the user. Where any link ambiguity exists, the odds of guessing correctly which link to follow diminish. A smaller number of links also increases the likelihood that those being looked for will be noticed. It is not necessary to mention all features of a site on all pages. To reduce the number of links, restrict those presented to the most important site content or useful features.

Embedded links. The trade-off that must be addressed in creating embedded links is linkability versus readability. Embedded links can be a distraction and reduce page readability, especially if used in abundance. They may also be overlooked in text scanning, especially if the scanning is not carefully done. Embedded links, however, can provide more meaningful context, adjacent phrases or sentence words being useful in understanding the link's purpose. The best trade-off is to incorporate embedded links in moderation. Reserve them for the most important, pertinent, and interesting document points. If other relevant content exists, present it in an explicit link listing.

Other Link Guidelines

In general, many of the principles in menu design presented earlier in this step, and to be presented in Step 7, also apply to presenting and organizing links. These guidelines should be reviewed in conjunction with this brief summary that follows:

- Grouping:
  - Group links by the most relevant menu-grouping scheme.
  - Separate visually the following types of navigation:
    - Upward to the immediate parent page.
    - Upward to the beginning of the section or category of information.
    - Across to main sections or categories of information.
    - To basic utilities.
- Ordering:
  - Order links by the most relevant menu choice-ordering scheme.
- For multilevel pages, provide one simple action to
  - Return to the next higher-level page.
  - Return to the homepage.
- Heading:
  - Where appropriate, provide a listing heading describing the organizing category, principle, or theme.
- Size:
  - Provide graphical images and command buttons of sufficient and equal size.

https://hemanthrajhemu.github.io

- Spacing:
  - Create equal spacing between choices graphical image and textual listing toolbars.
- Inapplicability:
  - Disable and display dimmed links conditionally not applicable.

Grouping. Place links of a similar purpose and function together. Develop groupings using the most relevant grouping scheme. In Web navigation, it is useful to visually separate the following types of links: (1) upward to the immediate parent page, (2) upward to the beginning of the presented section or category of information, (3) across to main sections or categories of information, and (4) to basic utilities. People make better link choices when they can readily eliminate wrong links. Grouping helps this process.

Ordering. Arrange the links by the most relevant menu-choice-ordering scheme, such as importance, frequency of use, or sequence of use, as previously described.

Multilevel pages. Navigation through menu levels should be accomplished through simple actions. It should always be very easy to return to the next higher-level page and the homepage.

Headings. When appropriate, provide an introductory word or phrase at the top of the link list as a heading. Inform viewers about the list's organizing category, principle, or theme. Establishing list context will aid users in selecting the correct link.

Size. To achieve balance, create a visually pleasing composition, make all links readily identifiable as links, create icons and command buttons of equal size. The size of any text inscribed on icons or buttons should also be consistent in size. In addition, explicit listings of textual links should be of the same size.

Spacing. To also achieve balance and a visually pleasing composition, all groups of links composed of icons, command buttons, listings of textual links should be equally spaced.

Inapplicability. Links that are irrelevant in a given situation should be disabled and displayed dimmed-out.

Link Maintenance

- Maintain correct internal links.
- Frequently check and correct external links.

As sites are modified, internal links may have to be revised. Carefully check sequential pages if the Next and Previous links are used within the site. External links should also be checked and corrected frequently. Because of the volatile nature of the Web, a linked site's content may change, its location may change, or a site may cease to exist. The credibility of a site's entire content suffers if it is not properly maintained.

https://hemanthrajhemu.github.io

Maintaining a Sense of Place

As has been said several times, a sense of place — where one currently is in the labyrinth of the Web — is often difficult to maintain. A site's organizational structure is often complex, and the boundaries between sites often seem nonexistent. Navigation links can transport a person from anywhere to anywhere, as does the *Star Trek* spaceship transporter machine. (While this machine moves the human to a new environment, the Web moves the new environment to the human.) These radical shifts in context created by jumping around information space through links can be extremely confusing. It is important that one's location be continually reinforced, because people desire stability and assurance that they are where they think they are. They also need a sense of exactly where they can go from their current location.

Paper documents create a sense of where one is located through a mixture of graphical and textual cues supplied by their design, including the varying fonts and images used. Cues are also provided by the organizational scheme outlined in the table of contents, and the physical sensation of the entire document itself. Looking at where a bookmark is placed in a novel provides an excellent indication of one's location in the reading space. The answers to questions like "Can I finish before the aircraft lands and the business conference starts?" are capable of being predicted with some reliability. Electronic documents provide few of these physical cues. To provide a sense of place, plentiful and explicit cues relating to site context and organization must be provided. These cues are provided by the site's overall design characteristics and the specific orientation elements included within the Web site.

Design Characteristics That Aid in Maintaining a Sense of Place

- To assist maintaining a sense of place within a Web site,
  — Provide a simple hierarchical tree structure.
  — Provide ease of movement to important site features.
- To assist maintaining a sense of place across multiple Web sites,
  — Provide consistency in all Web site design elements, including
    - Graphical identity schemes.
    - Component presentation.
    - Component organization and location.

Within a Web site. A simple hierarchical tree structure with obvious and linked major categories is an easily understood organization scheme. Easy identification of important site features, and ease of movement to them, is also important.

Across multiple Web sites. Design consistency contributes significantly to maintaining one's sense of place when one is moving between multiple sites. Design consistency gives a site a unique look and feel that becomes obvious as links are followed within it. Moving to a new site will be clearly evident when the design scheme changes. Consistency in the graphical identity scheme, use of colors, patterns, graphics, font styles, and so forth, will be the most noticeable aspects. Consistency in component presentation, organization, and location are also very important.

https://hemanthrajhemu.github.io

Design Elements That Aid in Maintaining a Sense of Place

- Provide a home base.
- Use recurring navigation tools on all pages.
- Use recurring elements on all pages.
- Provide page numbers for sequential pages.
- Provide ongoing feedback that shows where users are in a site.
- Provide on-demand aids that illustrate the user's location within a site.
  — Site maps.
  — Table of contents.
- Provide clearly written link labels.

Home base. As previously mentioned, a site's homepage is a stable, concrete, and safe anchor point to escape to in times of difficulty.

Recurring navigation tools. Standard navigation tools should appear on every page. In addition to creating uniformity in sight appearance, recurring tools create a more stable page environment, enhance navigation learning, and increase the user's control of the dialog.

Recurring page elements. Repeated page elements, such as titles, banners, logos, and icons, also create site uniformity. Omanson, Cline, Kilpatrick, and Dunkerton (1998) found that the page element that most significantly aided user orientation was the site logo.

Page numbers. For a long series of sequential pages, provide page numbers on each page to indicate where in the page string one is located. Another useful feature is to convert page numbers into links and present them on each page. A search, for example, may return a dozen pages of matches. At the bottom of each page inscribe, in link form, page numbers, as follows:

1 2 3 4 5 6 7 8 9 10 11 12

An estimation of document length is obtained, and the pages need not be viewed in sequential order.

Ongoing feedback showing location in Web site. Provide an historical trail, such as a breadcrumb trail, that shows where the user is located within a branch of a site. In addition to providing context for the displayed page, the trail permits easy return to any page up the trail.

On-demand aids illustrating location. Navigation aids, such as a site map or table of contents, when retrieved should show exactly where the user's current location fits within the structure of the site. The current position should be highlighted within the presented information structure. Ideally, in a site map, the complete navigation path from the homepage through intermediate pages to the current page should be presented.

Clearly written links. Labels that clearly indicate the function of the link, its destination, or its resulting action, reduce disorientation. Bad links are less likely to be followed and aimless wandering reduced.

https://hemanthrajhemu.github.io

Kinds of Graphical Menus

Providing the proper kinds of graphical menus to perform system tasks is also critical to system success. The best kind of menu to use in each situation depends on several factors. The following must be considered:

- The number of items to be presented in the menu.
- How often the menu is used.
- How often the menu contents may change.

Each kind of common graphical menu will be described in terms of purpose, advantages, disadvantages, and suggested proper usage. Design guidelines for each kind are also presented. A proper usage summary for the various kinds of menus are shown in Table 4.4 at the end of the menu discussion.

Menu Bar

- Proper usage:
  - — To identify and provide access to common and frequently used application actions that take place in a wide variety of different windows.
  - — A menu bar choice by itself should not initiate an action.

The highest-level graphical system menu is commonly called the menu bar. A menu bar consists of a collection of descriptions that serve as headings or titles for a series of actions on an associated pull-down menu. A menu bar choice by itself should not initiate an action.

The menu is typically arrayed in a horizontal row at the top of a window. Occasionally a menu bar is referred to as a collection of menu *titles*. In reality it is a menu in itself, and it is appropriate to simply refer to it as a menu. A menu bar is the starting point for many dialogs. Consistency in menu bar design and use will present to the user a stable, familiar, and comfortable starting point for all interactions. Menu bars are most effectively used for presenting common, frequent, or critical actions used on many windows in a variety of circumstances.

Menu bars often consist of a series of textual words, as represented in Figure 4.27. Macintosh, Presentation Manager, and Microsoft Windows illustrate examples of this textual approach. Some products have placed the choices within buttons, as represented in Figure 4.28. An example of this approach is Sun Microsystems' Open Look, which calls them *menu buttons*. There are also combinations of both. OSF/Motif presents a list of textual choices, but when one is selected, it resembles a button. Motif refers to these as *cascade buttons*.

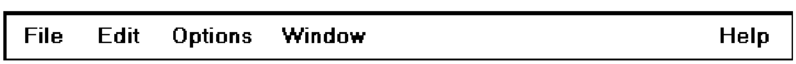


Figure 4.27: Menu bar composed of text.

https://hemanthrajhemu.github.io

Figure 4.28: Menu bar composed of buttons.

Each menu bar item is the top level of a hierarchical menu. It will have a pull-down menu associated with it, detailing the specific actions that may be performed. Some products have tried to circumvent this pull-down rule and have included items in menu bars that are direct actions themselves. These direct action items have frequently been designated by an exclamation point (!) following the menu bar description. The inclusion of direct items in a menu bar should be avoided. It creates inconsistency in menu bar use and may easily cause an action to be erroneously selected. Menu bars should always possess an associated pull-down menu.

Menu bars are used to present application alternatives or choices to the screen user. Typically, each system provides a default set of menu bar commands (for example, File, Edit, View, Window, Help).

The *advantages* of menu bars are that they

- Are always visible, reminding the user of their existence.
- Are easy to browse through.
- Are easy to locate consistently on the screen.
- Usually do not obscure the screen working area.
- Usually are not obscured by windows and dialog boxes.
- Allow for use of keyboard equivalents.

The *disadvantages* of menu bars are that

- They consume a full row of screen space.
- They require looking away from the main working area to find.
- They require moving pointer from the main working area to select.
- The menu options are smaller than full-size buttons, slowing selection time.
- Their horizontal orientation is less efficient for scanning.
- Their horizontal orientation limits number of choices that can be displayed.

Item Descriptions

- The menu item descriptions must clearly reflect the kinds of choices available.
- Menu item descriptions will be the “titles” for pull-down menus associated with them.
- Use mixed-case letters to describe choices.
- Use single-word choices whenever possible.
- Do not display choices that are never available to the user.

The menu item descriptions must clearly reflect alternatives available. Choices should be composed of mixed-case single words. Typically, only the first letter of the

https://hemanthrajhemu.github.io

choice is capitalized. Acronyms, abbreviations, or proper nouns that are normally capitalized may be capitalized. Choices should never be numbered.

If a multiple-word item must be used for clarity, consider including a hyphen between the multiple words to associate the words and differentiate them from other items. Do not display choices that are never available to the user.

Organization

- Follow standard platform ordering schemes where they exist.
  — Place application-specific choices where they fit best.
- Order choices left-to-right with
  — Most frequent choices to the left.
  — Related information grouped together.
- Choices found on more than one menu bar should be consistently positioned.
- Left-justify choices within the line.
- When choices can be logically grouped, provide visual logical groupings, if possible.
- Help, when included, should be located at the right side of the bar.

Figure 4.29

Follow standard platform ordering schemes where they exist. Place application-specific choices where they fit best. Order all choices left-to-right, with the most frequently elected choices to the left and related information grouped together. Choices found on more than one menu bar should be consistently positioned.

Left-justify all choices within the line (as opposed to centering them when there are not enough choices to completely fill the line). However, always locate Help, when included, at the far right side. Right side positioning will always keep Help in a consistent location within the bar. Also, provide visual groupings of all related choices, if space on the bar permits.

MAXIM **Hierarchical organization is the simplest structure for visualizing and understanding complexity.**

Pull-Down Menu

- Proper usage:
  — To initiate frequently used application actions that take place on a wide variety of different windows.
  — A small number of items.
  — Items best represented textually.
  — Items whose content rarely changes.

https://hemanthrajhemu.github.io

Selection of an alternative from the menu bar results in the display of the exact actions available to the user. These choices are displayed in a vertically arrayed listing that appears to pull down from the bar. Hence, these listings, as illustrated in Figure 4.30, are typically referred to as *pull-downs*. Other identification terms may be used, such as *drop-downs*.

Pull-downs are first-level menus used to provide access to common and frequently used application actions that take place on a wide variety of different windows. They are most useful for a small number of rarely changing items, usually about five to ten. Larger numbers of choices become awkward to use, being best handled by incorporating cascade menus (see discussion that follows). Pull-downs are best suited for items represented textually, but graphical presentations, such as colors, patterns, and shades, may also be used.

The *advantages* of pull-down menus are

- The menu bar cues a reminder of their existence.
- They may be located relatively consistently on the screen.
- No window space is consumed when they are not used.
- They are easy to browse through.
- Their vertical orientation is most efficient for scanning.
- Their vertical orientation is most efficient for grouping.
- Their vertical orientation permits more choices to be displayed.
- They allow for display of both keyboard equivalents and accelerators.

The *disadvantages* of pull-down menus are

- They require searching and selecting from another menu before seeing options.
- They require looking away from main working area to read.
- They require moving the pointer out of working area to select (unless using keyboard equivalents).
- The items are smaller than full-size buttons, slowing selection time.
- The may obscure the screen working area.

In Web use, for searching tasks, pull-down menus provide fastest performance. For browsing tasks, using the combined global/local navigation elements provided the fastest performance (Yu and Roh, 2002).

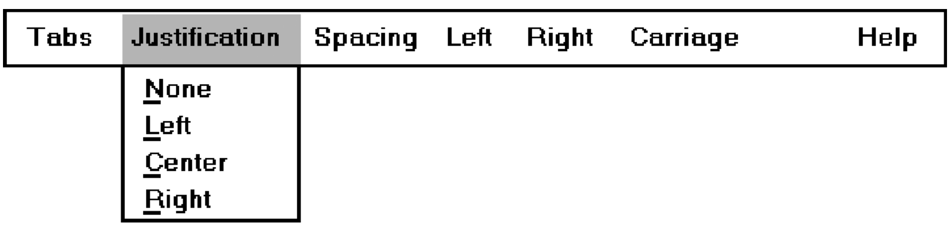


Figure 4.30: Menu bar pull-down.

https://hemanthrajhemu.github.io

Display

- Display all possible alternatives.
- Gray-out or dim items that cannot be chosen due to the current state of an application.

Display all possible alternatives on a pull-down. Gray-out or dim items that cannot be chosen due to the current state of an application. If all items are, at any one point, not applicable, they must still be capable of being retrieved for perusal through the menu bar.

Size

- Must contain a minimum of two choices.
- Restrict to no more than five to ten choices, preferably eight or less.

A typical pull-down consists of about five to ten choices, although more or less are sometimes seen. A pull-down should always contain more than one choice. Because of their vertical orientation, there is space for more choices containing longer descriptions than on a menu bar, and they can easily be positioned on one screen.

Organization

- Follow standard platform ordering schemes when they exist.
  — Place application-specific choices where they fit best.
- Place frequent or critical items at the top.
- Separate destructive choices from other choices.
- Provide a traditional, split, or folded structure, as necessary.
  — If a folded menu is used, visually differentiate the opened choices from those high frequency choices first displayed.
- Align choices into columns, with
  — Most frequent choices toward the top.
  — Related choices grouped together.
  — Choices found on more than one pull-down consistently positioned.
- Left-align choice descriptions.
- Multicolumn menus are not desirable. If necessary, organize top-to-bottom, then left-to-right.

Follow standard platform ordering schemes when they exist. Place application-specific choices where they fit best. Place frequent or critical items at the top of the listing, and separate destructive choices from other choices. Align all pull-down choices into

https://hemanthrajhemu.github.io

columns with their descriptions left-aligned. Locate most frequently chosen alternatives toward the top, and group related choices together. Provide a traditional, split, or folded structure, as necessary (see *Selection Support Menus* previously described). If a folded menu is used, visually differentiate the opened choices from those high frequency choices first displayed.

Choices found on more than one pull-down should be consistently positioned. Multicolumn menus are not desirable; if necessary, organize pull-downs from top-to-bottom, then left-to-right.

Groupings

- Provide groupings of related pull-down choices.
  - — Incorporate a solid line between major groupings.
  - — Incorporate a dotted or dashed line between subgroups.
  - — Left-justify the lines under the first letter of the columnized choice descriptions.
  - — Right-justify the lines under the last character of the longest choice description.
  - — Display the solid line in the same color as the choice descriptions.

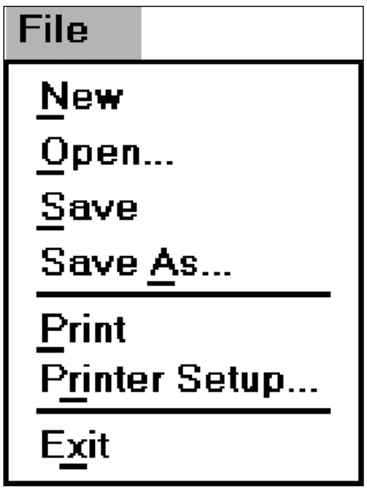


Figure 4.31

Indicate groupings of related choices by inscribing a line between each group. The line, or lines, should only extend from the first character of the descriptions to the end of the longest description, as shown above.

Some common style guides recommend that the line extend from pull-down border to border. Many other system pull-downs also follow this border-to-border approach. This extended line, however, results in too strong a visual separation between pull-down parts. The parts should be separated, but not too strongly.

https://hemanthrajhemu.github.io

Cascading Menus

- Proper usage:
  - — To reduce the number of choices presented together for selection (reduce menu breadth).
  - — When a menu specifies many alternatives and the alternatives can be grouped in meaningful related sets on a lower-level menu.
  - — When a choice leads to a short, fixed list of single-choice properties.
  - — When there are several fixed sets of related options.
  - — To simplify a menu.
  - — Avoid using for frequent, repetitive commands.

A *cascading* menu is a submenu derived from a higher-level menu, most typically a pull-down. Cascades may also be attached to other cascades or pop-up menus, however. Cascading menus are located to the right of the menu item on the previous menu to which they are related, as illustrated in Figure 4.32. Menu items that lead to cascading menus are typically indicated by a right-pointing triangle.

Cascading menus are developed to simplify menus by reducing the number of choices that appear together on one menu. Cascades can be used when many alternatives exist that can be grouped meaningfully. The top-level menu may contain the grouping category headings, and the cascaded menu the items in each group. Any menu choices with a fixed set of related options may utilize cascades.

The *advantages* of cascading menus are that

- The top-level menus are simplified because some choices are hidden.
- More first-letter mnemonics are available because menus possess fewer alternatives.
- High-level command browsing is easier because subtopics are hidden.

The *disadvantages* of cascading menus are

- Access to submenu items requires more steps.
- Access to submenu items requires a change in pointer movement direction.
- Exhaustive browsing is more difficult; some alternatives remain hidden as pull-downs become visible.

Changing pointer movement from a vertically oriented menu such as a pull-down to an adjacent cascade is an error-prone manual movement. Sliding the mouse and its pointer horizontally is not a very precise hand movement. As the pointer moves horizontally across the menu from which the cascade is selected it has a tendency to move vertically as well, sometimes exiting the menu over an item above or below the desired choice. When this occurs in Microsoft Windows the cascade displayed is not the one desired, but the cascade for the adjacent choice over which the pointer exited. The wrong cascade is then presented to the user, and the selection process must be repeated. Apple minimizes this problem by presenting a movement "cone" for the selected choice. This cone gradually widens as it approaches the cascade, extending somewhat over the adjacent choices. If the mouse and pointer exit the menu within an adjacent choice, but still within this cone, the originally designated cascade is still presented. The Apple solution is much more understanding of human motor limitations.

https://hemanthrajhemu.github.io

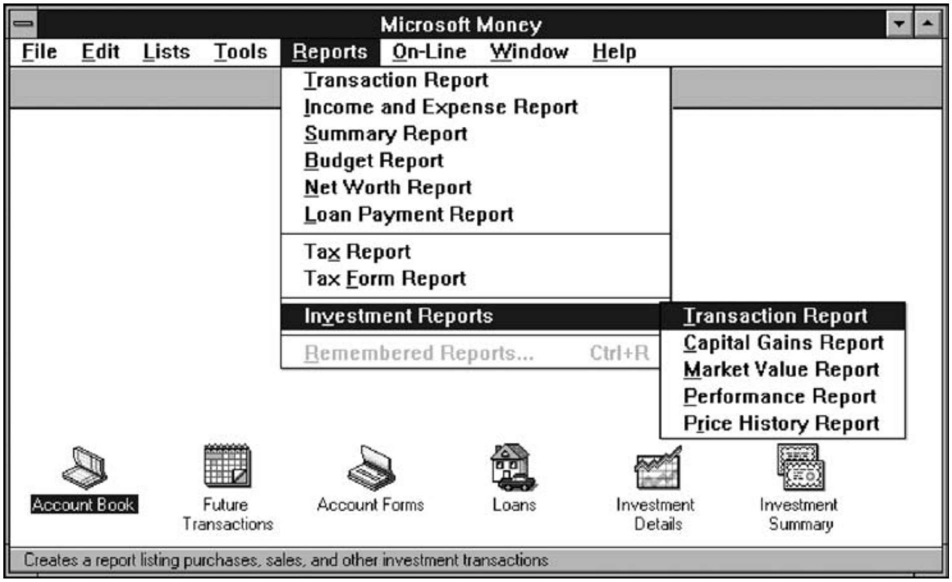


Figure 4.32: Cascading menu.

Cascade Indicator

- Place an arrow or right-pointing triangle to the right of each menu choice description leading to a cascade menu.
- Separate the indicator from the choice description by one space.
- Display the indicator in the same color as the choice descriptions.

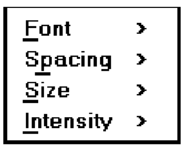


Figure 4.33

To indicate that another lower-level menu will appear when a menu item is selected, place an arrow or right-pointing triangle immediately to its right. Display the cascade indicator in the same color as the choice descriptions.

Levels

- Do not exceed three menu levels (two cascades).
  — Only one cascading menu is preferred.

https://hemanthrajhemu.github.io

Each additional cascade level presented reduces ease of access and increases visual clutter. The number of cascade levels presented should represent a balance between menu simplification, ease in menu comprehension, and ease in item selection. Whenever possible, do not exceed three levels of menus (original and two cascades). Try to limit cascades to one. If too many cascade levels are derived, create additional pull-down menus, or provide a window for some alternatives. A window is useful for establishing independent settings or the setting of multiple options. A toolbar may also be used to eliminate the necessity for traversing cascades.

Pop-Up Menus

- Use to present alternatives or choices within the context of the task.

Choices may also be presented to the user on the screen through *pop-up* menus, vertically arrayed listings that only appear when specifically requested. Pop-up menus may be requested when the mouse pointer is positioned over a designated or *hot* area of the screen (a window border or text, for example) or over a designated icon. In look, they usually resemble pull-down menus, as shown in Figure 4.34.

The kinds of choices displayed in pop-up menus are context sensitive, depending on where the pointer is positioned when the request is made. They are most useful for presenting alternatives within the context of the user's immediate task. If positioned over text, for example, a pop-up might include text-specific commands.

The *advantages* of pop-up menus are

- They appear in the working area.
- They do not use window space when not displayed.
- No pointer movement is needed if selected by button.
- Their vertical orientation is most efficient scanning.
- Their vertical orientation most efficient for grouping.
- Their vertical orientation allows more choices to be displayed.
- They may be able to remain showing ("pinned") when used frequently.
- They allow for display of both keyboard equivalents and accelerators.

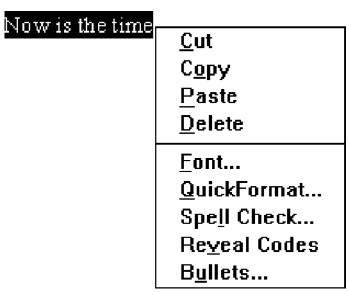


Figure 4.34: Pop-up menu.

https://hemanthrajhemu.github.io

The *disadvantages* of pop-up menus are

- Their existence must be learned and remembered.
- Means for selecting them must be learned and remembered.
- They require a special action to see the menu (mouse click).
- Items are smaller than full-size buttons, slowing selection time.
- They may obscure the screen working area.
- Their display locations may not be consistent.

For experienced users, pop-up menus are an alternative to retrieve frequently used contextual choices in pull-down menus. Choices should be limited in number and stable or infrequently changing in content.

Windows contains many contextual pop-up menus. They are also referred to as *context menus* or *shortcut menus*. Examples include the *window pop-up* and an *icon pop-up*, which presents operations of the objects represented by icons.

Display

- Provide a pop-up menu for common, frequent, contextual actions.
  — If the pointer is positioned over an object possessing more than one quality (for example, both text and graphics), at minimum present actions common to all object qualities.
- Items that cannot be chosen due to the current state of an application should not be displayed.
- Continue to display a pop-up until
  — A choice is selected.
  — An action outside the pop-up is initiated.
  — The user removes the pop-up.

Provide a pop-up menu for common, frequent, contextual actions. If the pointer is positioned over an object possessing more than one quality (for example, both text and graphics), at minimum present actions common to all object qualities. Items that cannot be chosen due to the current state of an application should not be displayed.

Continue to display a pop-up until the user selects a choice, initiates an action outside the pop-up, or requests that the pop-up be removed.

Location

- Position the pop-up
  — Centered and to the right of the object from which it was requested.
  — Close enough to the pointer so that the pointer can be easily moved onto the menu.
  — But not so close that the pointer is positioned on an item, possibly leading to accidental selection.

https://hemanthrajhemu.github.io

- If the pointer is positioned in such a manner that the pop-up would appear off-screen or clipped, position the menu
  — As close as possible to the object, but not covering the object.
  — So that it appears fully on the screen.

Position a pop-up menu in a consistent location relative to the object from which it is requested. The preferable location is centered to the right. Locate the pop-up close enough to the pointer so that the pointer can be easily moved onto the menu. Positioning of the pointer on the menu itself could lead to accidental selection of an action.

If the pointer is positioned in such a manner that a right-centered position would force the pop-up partially or fully off the screen, locate the pop-up fully on the screen as close as possible to the object. Do not move the pointer to make a menu fit in the most desirable location.

Size

- Restrict the pop-up to no more than five to ten choices, preferably eight or less.

Limit pop-up menus to about eight choices or fewer. If a large number of choices are needed, consider creating cascading menus. Minimize the number of levels of cascades, however, to provide ease of access and prevent visual clutter.

Tear-Off Menus

- Follow all relevant guidelines for pull-down menus.

A *tear-off* menu is a pull-down menu that can be positioned anywhere on the screen for constant referral. As such, it possesses all the characteristics of a pull-down. It may also be called a *pushpin*, *detachable*, or *roll-up* menu. Its purpose is to present alternatives or choices to the screen user that are needed infrequently at some times and heavily at other times.

Advantages/disadvantages. No space is consumed on the screen when the menu is not needed. When needed, it can remain continuously displayed. It does require extra steps to retrieve, and it may obscure the screen working area.

Tear-off menus are most useful for expert users. Use these menus in situations where the items are sometimes frequently selected and other times infrequently selected. Items should be small in number and rarely change in content. A typical use would be to detach and permanently leave displayed a pull-down menu when it must be frequently used.

https://hemanthrajhemu.github.io

Because a tear-off menu is a pull-down style, all pull-down guidelines should be followed.

Iconic Menus

- Use to remind users of the functions, commands, attributes, or application choices available.
- Create icons that
  - — Help enhance recognition and hasten option selection.
  - — Are concrete and meaningful.
  - — Clearly represent choices.

An iconic menu is the portrayal of menu items or objects in a graphic or pictorial form. The purpose of an iconic menu is to remind users of the functions, commands, attributes, or application choices available.

Advantages/disadvantages. Pictures help facilitate memory of applications, and their larger size increases speed of selection. Pictures do, however, consume considerably more screen space than text, and they are difficult to organize for scanning efficiency. To create meaningful icons requires special skills and an extended amount of time. Iconic menus should be used to designate applications or special functions within an application. Icons must be meaningful and clear. They should help enhance recognition and hasten option selection. See Step 11 for a complete review of icon design guidelines.

Pie Menus

- Consider using for
  - — Mouse-driven selections, with one- or two-level hierarchies, short lists, and choices conducive to the format.

A *pie* menu is a circular representation of menu items, as illustrated in Figure 4.35 that can be used as an alternative to a pull-down or pop-up menu. Research has found that this style of menu yields higher performance than the typical vertical array, especially when the menu tasks are unrelated. Their basic advantage is that, when presented with the mouse pointer positioned in the pie's center, average movement to any pie wedge is shorter. Mayhew (1992) concludes that pie menus might work well for mouse-driven selections with one- or two-level hierarchies, short choice listings, and data conducive to the format. Performance advantages for keyboard selection are doubtful, however.

https://hemanthrajhemu.github.io

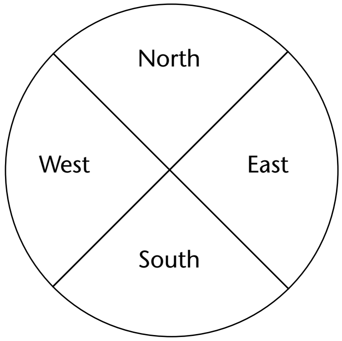


Figure 4.35: Pie menu.

Table 4.4: Menu Proper Usage Summary

Menu Bar	To identify and provide access to • Common and frequently used application actions. • Actions that take place in a wide variety of different windows.
Pull-Down Menu	For frequently used application actions that take place in a wide variety of different windows, • A small number of items (five–ten). • Items rarely changing in content.
Cascading Menu	To simplify a higher-level menu. To provide easier browsing of a higher-level menu. For mutually exclusive choices. Restrict to one–two cascades.
Pop-Up Menu	For • Frequent users. • Frequently used contextual commands. • A small number of items (five–ten). • Items rarely changing in content. • Items that require a small amount of screen space.
Tear-Off Menu	For items • Sometimes frequently selected. • Sometimes infrequently selected. • Small in number (five–ten). • Rarely changing in content.
Iconic Menu	To designate applications available. To designate special functions within an application.

https://hemanthrajhemu.github.io

Graphical Menu Examples

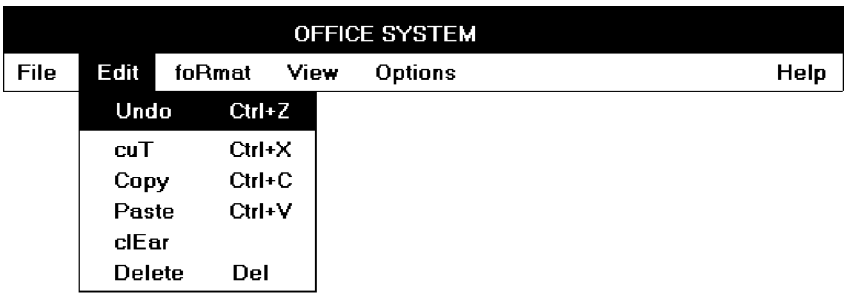
What follows are examples of poor and proper menu design.

Example 1

An improperly presented menu bar and pull-down.

Menu 1.1

What are the problems in the way this menu bar and pull-down menu are presented? (1) Keyboard mnemonics are designated by capital letters. Note the uncommon shape of "foRmat," "cuT," and "clEar" when the mnemonic is not the first letter of the word. (2) Item groupings do not exist in the pull-down. The differences in basic functions are not obvious, and the more destructive operations (Undo, Clear, and Delete) are positioned close to standard actions, increasing the potential for accidental selection. (3) The keyboard accelerators are adjacent to the choice descriptions and not set off in any way. Therefore, these alternate, and supplemental, actions visually compete with choice descriptions for the viewer's attention.

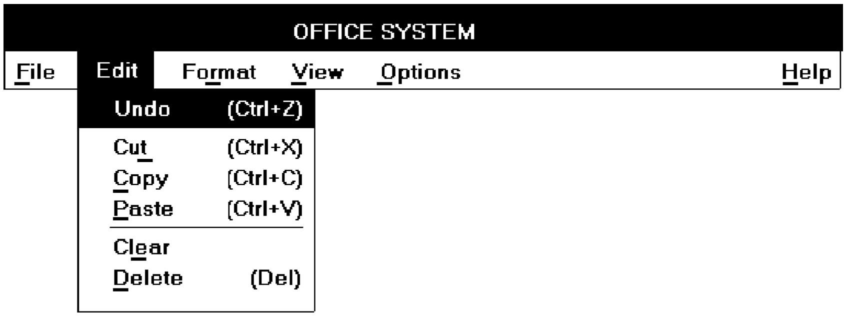


Menu 1.1

Menu 1.2

Keyboard mnemonics are designated by underlines, not capital letters. Choice descriptions now assume more common and recognizable shapes. Groupings, through use of white space, are established for choices in the pull-down. The different functions are much more obvious and separation is provided for the destructive actions. The different groupings are visually reinforced through use of separating lines. The lines are not extended to the pull-down border so as not to completely disassociate the choices. Keyboard alternatives are right-aligned to move them further from the choice descriptions. They are also enclosed in parentheses to visually deemphasize them, thereby reducing their visual competition with the choices. Choice descriptions are now more obvious.

https://hemanthrajhemu.github.io



Menu 1.2

More examples and an exercise for Step 4 can be found on this book's companion Web site, www.wiley.com/college/galitz.

https://hemanthrajhemu.github.io

FUTURE VISION BIE

By K B Hemanth Raj

Visit : https://hemanthrajhemu.github.io

Quick Links for Faster Access.

CSE 8th Semester - https://hemanthrajhemu.github.io/CSE8/

ISE 8th Semester - https://hemanthrajhemu.github.io/ISE8/

ECE 8th Semester - https://hemanthrajhemu.github.io/ECE8/

8th Semester CSE - TEXTBOOK - NOTES - QP - SCANNER & MORE

17CS81 IOT - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS81/

17CS82 BDA - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS82/

17CS832 UID - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS832/

17CS834 SMS - https://hemanthrajhemu.github.io/CSE8/17SCHEME/17CS834/

8th Semester Computer Science & Engineering (CSE)

8th Semester CSE Text Books: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Text-Book.html

8th Semester CSE Notes: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Notes.html

8th Semester CSE Question Paper: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Paper.html

8th Semester CSE Scanner: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Scanner.html

8th Semester CSE Question Bank: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Question-Bank.html

8th Semester CSE Answer Script: https://hemanthrajhemu.github.io/CSE8/17SCHEME/Answer-Script.html

Contribution Link:

https://hemanthrajhemu.github.io/Contribution/

Stay Connected... get Updated... ask your queries...

Join Telegram to get Instant Updates:
https://telegram.me/joinchat/AAAAFTtp8kuvCHALxuMaQ

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/futurevisionbie/