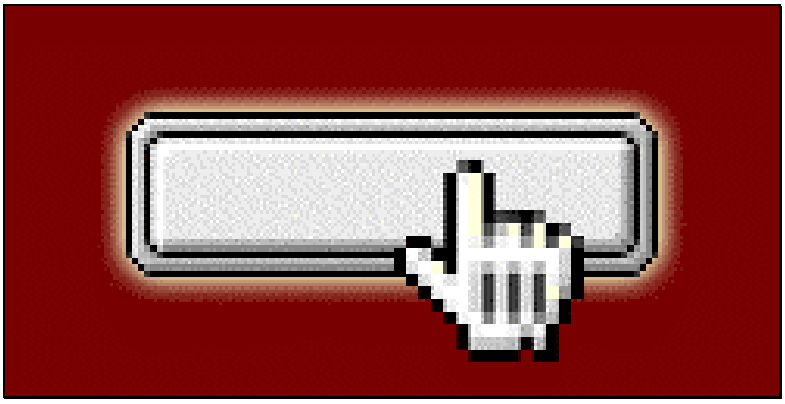


Web Site Interface Design Theory: *A Designer's Primer*

By Ari Feldman
http://www.flyingyogi.com

To date, employees from the following organizations have downloaded this document: **CIBC Bank, Boeing**, **Wunderman**, **Siemens Germany**, **NASA's Jet Propulsion Laboratory**, **Lockheed Martin**, **Renegade Marketing**, **IBM UK**, **The Australian Capital Territory**, **LearningFramework**, **Distance Learning Incorporated**, **Scandinavian Airlines**, **Dailey Advertising**, **Unilever** and numerous others.



READER COMMENTS:

“I want to let all 2500 people on this list [WWWAC List] know about Ari's great work on designing user-centric interfaces and also in being able to explain the why, what and how of the process. I put his work on the subject at a level equal or better than any of the boring texts I've read on human-machine interfaces, and certainly worthy of inclusion in anyone's library of research materials.”

— Martin Focazio, Internet Consultant

“Your document concentrates on the essentials, if understood and followed, will precipitate usability. It is devoid of the Gee Wiz which many books and documents gravitate toward. It is the fundamentals which I teach in my course, and I was surprised how close your document was to my lecture notes, and my vision of User Interface Design.”

— Dave Bockus, Instructor, Brock University, Canada

Table of Contents

What is a User Interface and Why do you need one?

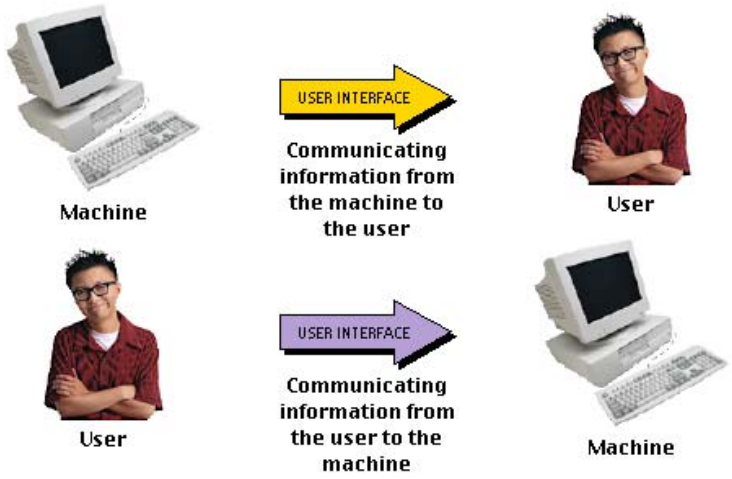
User Interface. You've probably heard the term before but do you really know what it means? If not, here's an official definition courtesy of the *Free Online Dictionary of Computing* (dated January, 19, 2001):

> "**User Interface** - The aspects of a computer system or program which can be seen (or heard or otherwise perceived) by the human user, and the commands and mechanisms the user uses to control its operation and input data."

In other words, a user interface is a set of rules, methods, and/or devices used to promote the interaction between people and machines. For our purposes, people mean *users* and machines mean *computers*.

To accomplish this, a user interface can take on many different forms but ultimately, it must *always* accomplish the same result: to enable a two-way dialog between the user and the computer as illustrated by Figure 1.

FIGURE 1: The Two-Way User-Machine Dialog via a User Interface



Software Interfaces vs. Web Site Interfaces

Despite the fact that the role of a user interface is universal, there are several important differences between software interfaces and web sites interfaces. These differences include:

- Functionality
- Layout Control
- Navigation
- Standards

Functionality

Software interfaces offer a larger array of interface elements than web site interfaces do. In addition to supporting the usual elements like buttons and forms, software applications can also use moveable windows, dialog boxes, tabbed dialogs, drop-down menus, and spin boxes.

In comparison, web site interfaces don't support any of these elements beyond text links, buttons, and forms. In addition, they need to rely on outside technologies such as Flash and Java to provide any additional interface functionality.

Layout Control

Software interfaces give the designer complete control over the environment. They allow interface elements to be created and placed with pixel-level precision, which enables very sophisticated interfaces to be developed.

In comparison, web site interfaces give the designer limited control over the environment. In a web interface, only a handful of elements can be controlled at the pixel-level while most web interface elements are limited to character-level positioning. Furthermore, the appearance of these elements is not consistent between the various browsers and platforms that are available.

Navigation

Traditional software interfaces can limit where a user can go within the environment. Software applications can disable interface elements as well as restrict the actions a given user can take.

On the web, the user is in complete control of the environment. Web site interfaces are transient in nature and allow users to move freely across different web pages and web sites without restrictions.

Standards

Software interfaces tend to look and behave the same, regardless of whether the user is typing a letter in a word processor or accessing records in a database.

Web interfaces, in comparison, tend to look and behave differently depending on how they are being accessed – i.e. the web, WebTV, PDA, or on a wireless device. Each web platform supports different standards and with various degrees of adherence to these standards.

Software Interfaces vs. Web Site Interfaces Comparison Summary

- Software interfaces offer users a much richer interface environment than web site interfaces.
- Software interfaces give the designer greater visual control over the environment presented to the user than do web site interfaces.
- Web site user interfaces are far less restrictive than software user interfaces when it comes to user navigation.

- Software interfaces tend to be much more consistent in their appearance and behavior than their web-based counterparts.

However, there's a bit more to user interfaces than just facilitating the communication between man and machine. You see, for any user interface to be considered truly effective, it must also meet these two criteria:

- It must be user-centric in its design
- It must stress usability

 **NOTE:** A user interface that fails to meet either of these criteria is said to be "broken" and ineffective. This, as you'll discover later, can cause a variety of user-related problems.

User-Centric Design

In order to serve the user, a user interface must be *user-centric* in nature, or completely user focused in both its design and implementation. This means that the interface (and its various components) needs to put the user first and take into account their various goals, needs, and expectations when accomplishing a set task. At the same time, it must also be flexible enough to accommodate the user's personal tastes and preferences.

FIGURE 2: User Centric Interface Design: It's all about accommodating the user

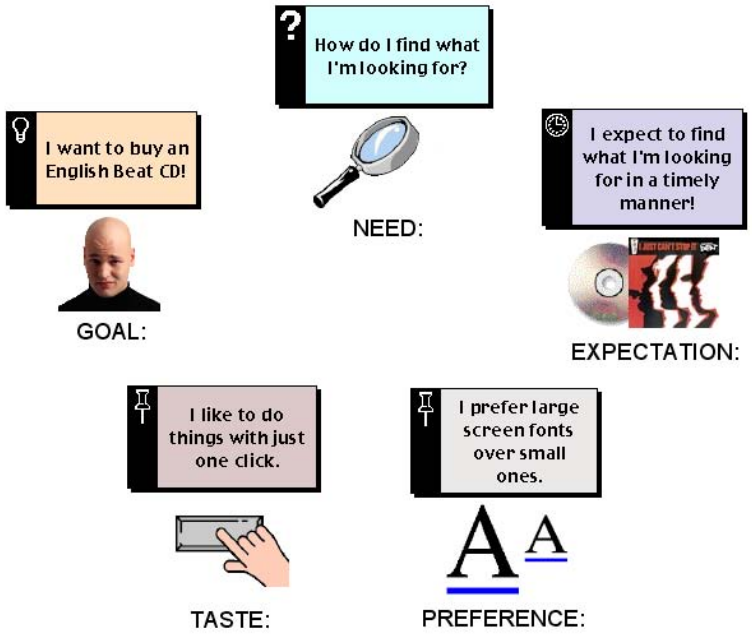


Figure 2 summarizes the importance and the need for user-centric interface design from the user's perspective.

There are two aspects of user-centric design that require special attention. These are:

- User Faculties
- Computer Experience

User Faculties

From the very outset of designing a user interface, it's very important to realize that not all users are created equally. Some users will be veritable geniuses while others will have the IQ of a carrot. In fact, every user will possess vastly different *faculties*, or levels of visual perception, visual acuity, cognitive ability, memory retention, motor skills, and intelligence. Among other things, these faculties can directly influence the user's ability to interact with and manipulate a given user interface.

Table 1 defines what exactly these faculties are and illustrates their potential implications in interface design.

TABLE 1: The User Faculty Matrix

Faculty	Purpose	Potential Implications	User's Perspective
Visual Perception	Defines how one sees and experiences things visually, particularly with regards to shape and color.	A user with poor visual perception may not perceive the importance of a particular interface object due to its color or shape.	"Do I press on the big green thing or the small red one?"
Visual Acuity	Measures the eye's ability to resolve details.	A user with poor visual acuity may find it difficult to focus on specific interface objects or areas of within the interface.	"Where exactly do I find the 'preferences' button?"
Cognitive Ability	Defines one's ability to understand and process what they learn.	A user with relatively low cognitive ability might not be able to figure out how a given interface works, despite being given hints and other cues about how to use it.	"How the does this thing work?"
Memory Retention	Defines one's ability to retain what they have learned.	A user with poor memory retention may forget how to use an interface despite having used it successfully before.	"Which button do I press again?"
Motor Skills	Defines one's ability to physically manipulate and interact with their environment, such as moving a mouse or typing at a keyboard.	A user with poor motor skills will almost certainly encounter trouble using an interface that requires extensive mouse input, etc.	"I find moving the mouse around the screen very awkward!"
Intelligence	Measures one's ability to comprehend, understand, and benefit from experience.	A highly intelligent user is more likely to figure out how to use an interface than a less intelligent user.	**Intelligent User:** "This interface is a snap to use as it's very similar to other interfaces I've used in the past." **Less Intelligent User:** "I can't figure this thing out. There are so many things to click and I don't know what I'm doing!"

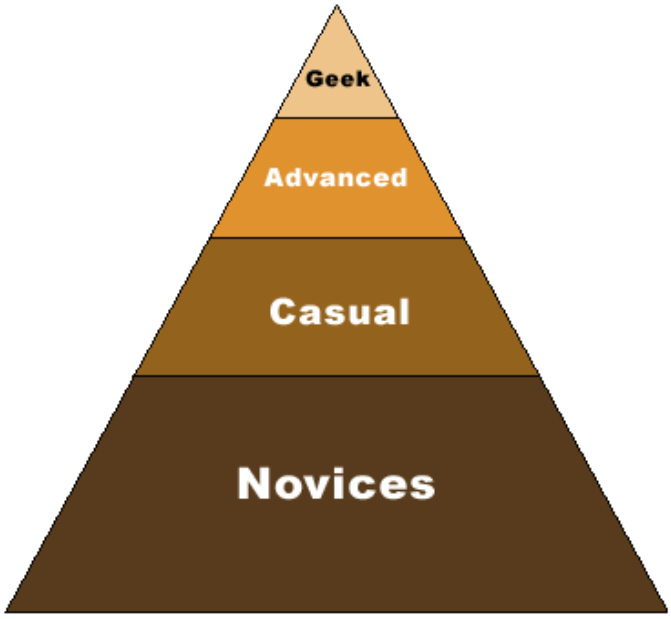
It's important to realize that despite your best efforts, you can never fully account for user faculties. However, with careful planning and consideration in how you design your interface, you *can* minimize the impact they have.

Computer Experience

In addition, users will also approach an interface with varying degrees of *computer experience*. For our purposes, computer experience represents more than just the number of years someone has worked with computers but also refers to their relative skill level and comfort in using them. For example, my mother has several years of computer experience but isn't very skilled or comfortable with using one. This being said, any of these factors can dramatically affect how the user interacts with a given user interface.

To illustrate the disparity of computer experience among different users, consider the chart pictured in Figure 3. In it, users are grouped according to their relative computer experience and skill, with the most sophisticated users occupying at the top of the pyramid and the least sophisticated users at the bottom.

FIGURE 3: The Computer Experience Pyramid



As you can see, this particular arrangement implies that there are far more novices than there are experts. This is no accident. As computers penetrated more and more homes, schools, and small businesses over the years, the face of the computer user has undergone radical change. Today's computer users now fully reflect the mainstream of society. These users are our parents, teachers, grandparents, aunts, uncles, brothers, and sisters and *not* the disheveled hackers or geek types that typified computer users just a few short years ago.

NOTE: Pay special attention to this trend, as it's only likely to continue over time and create even more challenges for those of us who design and develop user interfaces.

Although there is no simple way to attack the immediate problem of user computer literacy, we can create user interfaces that are familiar, comfortable, and non-threatening to users. Doing this will go a long way in overcoming the limits imposed by one's skill and computer experience. One of the best ways to accomplish this is to understand the concept of *usability*.

Usability

Usability measures the overall effectiveness with which a user can achieve their tasks and goals while using a given interface. An interface with a high level of usability means it is easy to learn, efficient, and fun to use while one with a low level of usability means the exact the opposite. As a designer, you should always strive to maximize the usability of your interface over anything else, including aesthetics. That being said, let's look at the two most important components of usability:

- Efficiency
- Ease of Use

Efficiency

Efficiency measures the savings of time and effort while a user performs a specific task. It really comes into its own when the user is carrying out particularly tedious or mission-critical tasks like data-entry (tedious) or placing stock trades (mission-critical), etc.

As a rule, the more efficient a user interface is, the faster the user will be able to complete their task. This not only saves the user valuable time but it also allows them to engage in more work in less time, thus enhancing their overall level of productivity. In addition, efficiency can also help the user reduce the number and severity of the errors that occur while they perform a task (i.e. typos, clicking the wrong button, etc.) by helping them to avoid engaging in unnecessary work. Therefore, it can be said that efficiency increases both the *quantity* and *quality* of a user's work.

Efficiency is a learned process. Users tend to become more efficient at performing certain tasks through time, experience, and repeated exposure to them. For example, if a user performs the same task every day for a number of years, they will become slightly more efficient (better, faster, etc.) each time they do it.

By studying and understanding how a user performs a given task, we can design interfaces that maximize and enhance their efficiency.

Ease of Use

Ease of use implies that a given interface is both simple and intuitive to use. The advantages of making interfaces easy to use are numerous as the easier a user interface is, the better the user experience, and the happier the user will be. Ease of use also contributes to user efficiency, as users are able to spend more of their time doing actual work rather than trying to figure out how to use the interface.

Therefore, you should consider usability to be one of your most important goals when designing a user interface.

Negative Emotional Reactions to User Interfaces

In general, users don't respond well to interfaces with poorly designed interfaces. In fact, depending on the exact circumstances, you can expect users to exhibit one or more of the following emotional reactions with they encounter usability problems with an interface, including:

- Confusion/Disorientation
- Frustration
- Boredom
- Anxiety
- Anger

Confusion/Disorientation

Confusion and/or disorientation can occur if the user is overwhelmed by the interface in some way. For example, they are presented with too many options or a distracting design, etc. When confused, a user simply won't know what to do next.

Because of this, confusion can create efficiency problems as the user's work is interrupted while they try to figure out their next steps. Confusion is dangerous to interface designers because it can lead to other emotional states, such as frustration.

Frustration

Frustration can occur when the user is unable to accomplish the current task due to some problem they encounter with the interface. For example, the interface might produce an unforeseen event such as an error condition, etc. Frustration is one of the worst user emotional responses because it can easily escalate to anger.

In addition, frustration can have a negative impact on a user's level of efficiency, as their work is interrupted while they try to deal with the problem or condition. Moreover, the user's feelings of frustration may eventually cause them to avoid attempting the same task again in the future.

Boredom

Boredom usually occurs after a user experiences an inordinate delay while using a given interface (see Latency). For example, the user clicks a button and nothing seems to happen, etc.

Because bored users effectively "tune out" while performing a task, their level of efficiency tends to drop and they often produce poor quality work. Moreover, excessive and repeated boredom can lead to frustration over time.

Anxiety

User anxiety may have a number of causes, including unusual external pressure (i.e. the boss looking over the user's shoulder), negative past experiences while using the computer, or just being presented with too many interface options at once (see Chunking).

Users who are anxious users are very likely to experience efficiency problems due to their uncertainty and fears about what to do next. They are also less likely to make future attempts at completing a given task due to these fears. Over time, a certain percentage of anxious users may also become frustrated while trying to use a problematic interface.

Anger

Anger is the most unpleasant of all of the emotions a user can experience and usually develops from feelings of extreme frustration with the interface. For example, the user clicks on a button and the wrong thing happens repeatedly (i.e. aborting an operation instead of continuing it, etc.).

Angry users tend to have poor levels of efficiency and are very unlikely to complete their current task. Even worse, angry users have a tendency to become violent, possibly to the point of injuring their computer, other users, or even themselves.

You are to avoid angering a user at all costs! Causing a user to become angry through an interface you designed is a cardinal sin and virtually inexcusable. If this happens, you should really consider another line of work.

While there's never a guarantee that you can prevent a user from having a negative experience while using a given interface, you can reduce the likelihood by maximizing the overall usability of the interfaces you create. Obviously, this is easier said than done. However, one of the best ways to accomplish it is to religiously follow and apply the guidelines and principles that are described in this document.

Layout & Organization in Web Interfaces

The following web interface design concepts are useful for organizing and arranging the various elements within a web interface:

- Chunking
- Grouping
- Hick's Law

Chunking

In interface design, *chunking* refers to the processing of small pieces or "chunks" of information. These pieces can consist of anything: numbers, letters, words, objects, or even sequences of events. Chunking is based on the work of psychologist George A. Miller, who published a famous study in 1956 that examined a user's capacity to store and process data.

According to Miller, the average person can only store, recall, and process seven bits of information – plus or minus two – at any given time. This ability can be summed up by a simple "magic number" formula:

7 +/- 2

As you can imagine, this theory has special implications to web interface design as it places certain limits on the number of choices we can present at the same time. If people are exposed to too many options at once, they may overload their capacity to process this information and become confused about what to do next.

FIGURE 4: Example of Poor Interface Chunking

Austria	Argentina	USA Alabama
Belgium	Armenia	USA Alaska
Bosnia Herzegovina	Australia New South Wales	USA Arizona
Bulgaria	Australia Northern Territory	USA Arkansas
Croatia	Australia Queensland	USA California, North
Cyprus	Australia South Australia	USA California, South
Czech Republic	Australia Victoria & Tasmania	USA Colorado
Denmark	Australia Western Australia	USA Connecticut
Estonia	Bahamas	USA Dakotas
Finland	Bahrain	USA Delaware
France	Belarus	USA Dist. Columbia
Germany	Bermuda	USA Florida
Greece	Bolivia	USA Georgia
Hungary	Botswana	USA Hawaii
Haiti	Brazil	USA Idaho
Iceland	Cambodia	USA Illinois
Ireland	Canada Atlantic	USA Indiana
Italy	Canada British Columbia	USA Iowa
Latvia	Canada Ontario	USA Kansas
Lithuania	Canada Prairies	USA Kentucky
Luxembourg	Canada Quebec	USA Louisiana
Macedonia	Canada Yukon & NW Terr.	USA Maine
Malta	Caribbean	USA Maryland
Monaco	Chile	USA Massachusetts
Montenegro	China	USA Michigan
Netherlands	Colombia	USA Minnesota

Many web portals and directories violate the concepts behind information chunking, as they provide too many options at once. Even experienced users may have difficulty sifting through this huge number of choices.

FIGURE 5: Example of Good Interface Chunking

This is an example of good information chunking in a web interface. Here, the interface presents the user with only six choices, which, according to Miller, is well within most users' information processing abilities.

FIGURE 6: Extended Chunking in a Web interface

Despite Miller's findings, you can exceed the "magic number" formula in certain situations if you break up your choices into separate groups (see Grouping). This menu includes eleven options arranged into two distinct groups, keeping the number of choices presented manageable.

Grouping

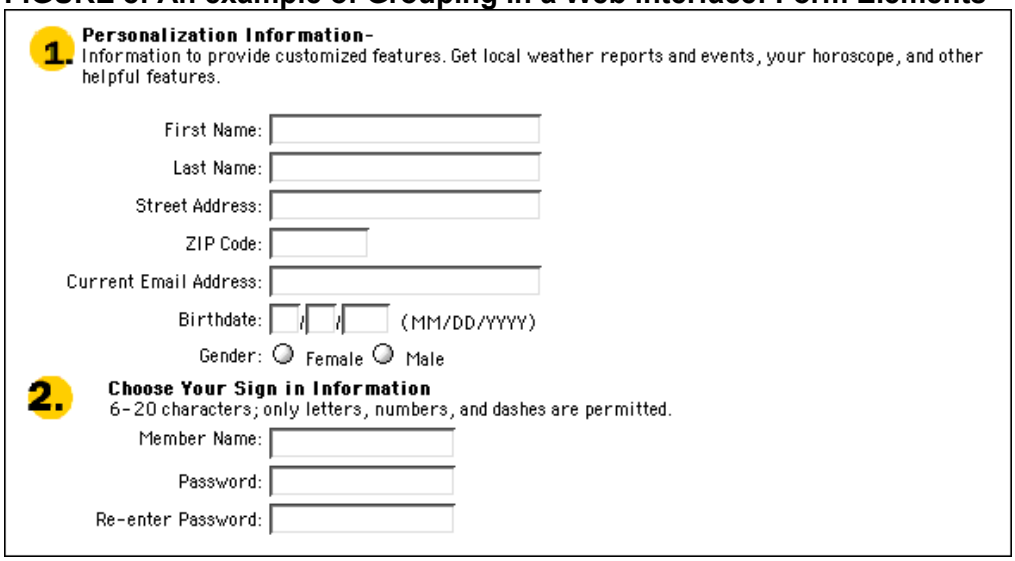
Grouping is a technique for clustering different web interface elements together. It has three important uses in a web interface:

1. It helps make the interface more aesthetically pleasing by creating balance and minimizing clutter.
2. It helps users to find what they're looking for faster.
3. It helps users to see relationships between certain interface elements, making the interface more intuitive to use.

FIGURE 7: An example of Grouping in a Web interface: Text Links

Grouping related elements together can help to minimize the impact of a cluttered interface by making information easier to find.

FIGURE 8: An example of Grouping in a Web interface: Form Elements



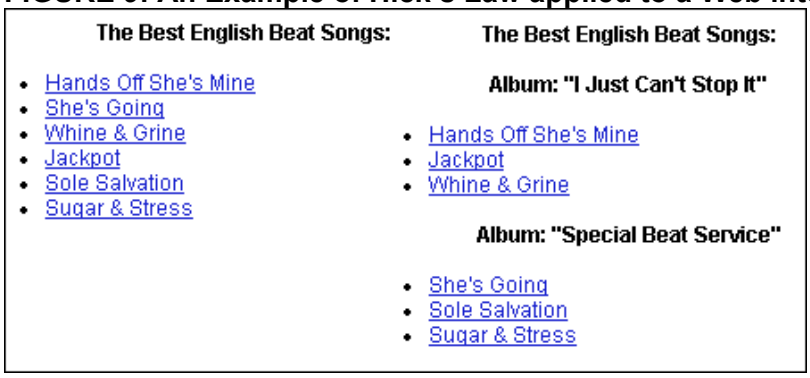
Grouping isn't restricted to text links. Any interface element can be grouped. Here, grouping is used to organize the contents of a web Form, making it easier for the user to distinguish between the various elements.

Hick's Law

Hick's Law is a set of formulas that mathematically prove the observations of how users react when confronted with multiple choices of a given type. It says that, all other things being equal, it's faster for a user to make selections from a single menu with many choices than two or more menus with fewer choices.

Hick's Law really comes into its own when you need to determine the best way to hierarchically arrange interface elements of a particular kind, such as text links or Form elements.

FIGURE 9: An Example of Hick's Law applied to a Web interface



Hick's Law can be used to simplify the organization of many web interfaces. In this example, it will take the user less time to choose a selection from a single menu of six elements than it will be from two menus of three elements each.

Usability & Experience in Web Interfaces

The following web interface design concepts make web interfaces more efficient, responsive, safer, and easier to use:

- Affordances
- Consistency
- Context
- Data Protection
- Default Devices
- Feedback
- Fitt's Law
- Forgiveness
- Latency
- Legibility
- Maintaining State
- Visibility

Affordances

Affordances are one of the cornerstones of web-based interface design. The term was originally coined by psychologist J.J. Gibson but later extended and applied to interface design by psychologist Donald A. Norman. According to Norman, affordances are features or devices that help users determine how an object might be used.

Norman identified two kinds of affordances: *Cognitive Affordances* and *Physical Affordances*. Cognitive affordances help the user to *know* something while physical affordances help the user to *do* something. It's a subtle yet important difference.

FIGURE 10: Cognitive Affordance Example

An example of a cognitive affordance is the common elevator button. Its shape, size and beveled appearance help you know how to use it. Thus, the button's design "affords" pressing.

FIGURE 11: Physical Affordance Example

A door handle is a type of physical affordance. Its unique contour and downward shape help you figure out how to open a door. Thus, the door handle's design "affords" pushing it down.

Cognitive affordances are particularly interesting because of their unique ability to guide the user within an interface. They can accomplish this in a number of ways, including:

- Helping the user to know what to do next
- Helping the user avoid potential errors and mistakes
- Helping the user know how to use an object or device
- Making the user aware of shortcuts and their alternatives

Affordances are the prime enablers of any web interface. They allow us to imbue different interface elements with a variety of powerful and self-explanatory visual and cognitive cues. These cues can then help the user understand their purpose and function, dramatically increasing the user's efficiency and experience within an interface. Without these cues, users of the interface might become confused, frustrated, or even distracted.

Characteristics such as color, shape, styling, size, depth, screen position, animation, context, and visibility are all examples of affordances. Furthermore, almost any interface element can have an affordance applied to it, including text.

FIGURE 12: The advantage of interface affordances

BUTTON A **BUTTON B**

Notice how color, shape, styling, and depth all work together to make Button A seem like it can be clicked. In contrast, Button B looks doesn't give the user any indication of what it is, let alone what can be done with it. Buttons that lack affordances are all too common in many web interfaces.

The best-designed interface elements actually employ several affordances concurrently (i.e. size, color, etc.) while the best-designed web interfaces contain multiple examples of affordances, as shown in Figures 13 and 14.

FIGURE 13: Well designed affordances in interface elements

Each of these interface buttons contains several affordances:

1. The shapes and beveled appearance of each button affords pressing, essentially showing the user how they work.
2. Labels make each button's function immediately known, helping the user to avoid mistakes.
3. The simple and clear graphic symbols on each button act as shortcuts to their function, which is especially useful to users with cognitive disabilities.

FIGURE 14: Multiple examples of affordances in a Web interface

This interface contains several examples of affordances:

Affordance #1: the main navigation buttons.

- The color, size, shape, and 3D styling of each button afford pushing. This action is further reinforced through graphical symbols and clear labels, which indicate the function and purpose of each button.

- Their location and placement at the top-center of the screen helps users to see and access them easily.

Affordance #2: the Shopping Bag Text link.

- The size, color, and styling of the text link (all caps, boldface, and underlining) all afford clicking. In addition, the presence of the small shopping bag graphic reinforces its purpose within the interface.

- The link's location and placement at the top-right of the screen helps users to see and access it. Since many E-Commerce sites traditionally use that page location for their "shopping bag" or "shopping cart" functions, users with prior web experience are likely to understand its function within the interface.

Affordance #3: the "Item of the Week"

- The element's unique shape, border color, and drop shadow all afford pressing. In addition, the accompanying graphic symbol of a T-Shirt helps the user understand its purpose.

- Setting this element slightly apart from the other elements and separating it with ample white space improves the chances of it being seen.

Look at this interface carefully, and you'll find that it offers several other affordances. Can you find them all? **Hint:** Interface elements with drop shadows or any underlined text links are affordances and thus, clickable. In addition, pay special attention to the size and styling of text elements. They serve as visual reinforcements to a function's importance within the interface.

In contrast, the worst web interfaces have few, if any obvious affordances in them as illustrated by the interface pictured in Figure 15.

FIGURE 15: How missing affordances can contribute to a poor Web Interface

The interface pictured in Figure 15 has few, if any visible affordances. Nothing immediately stands out and says to the user “click on me” or “this is how to use me”, etc. For example, which elements can be clicked on and which just provide information? Do you listen to the music samples by clicking the tiny speaker icons or the track titles? What do you click to add an album to your shopping cart? Any of these issues can be fatal to a web site’s usability, especially one that targets users of all experience levels as this one does.

NOTE: The vast majority of usability problems in web interfaces can be attributed to missing or improperly applied affordances.

Consistency

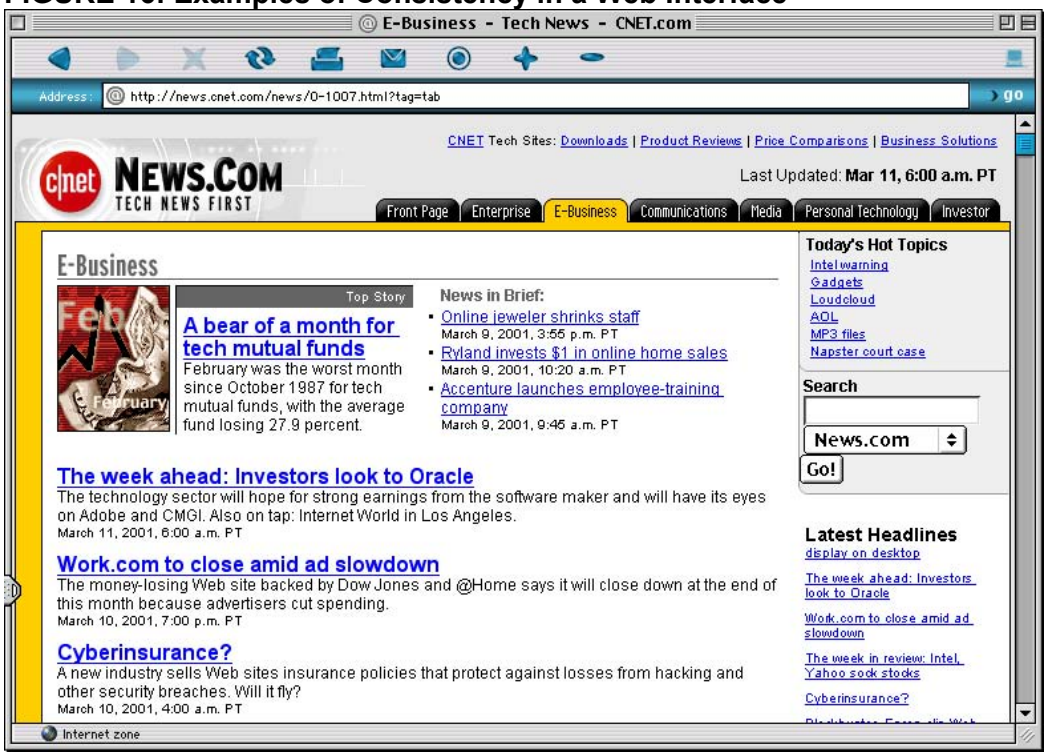
Consistency, one of the most important elements of web design, helps users anticipate the behavior of an interface based on their prior knowledge and/or computer experience. Simply put, consistency aids the user in both learning and remembering how a given interface works. To be consistent, all web interfaces should follow these basic rules:

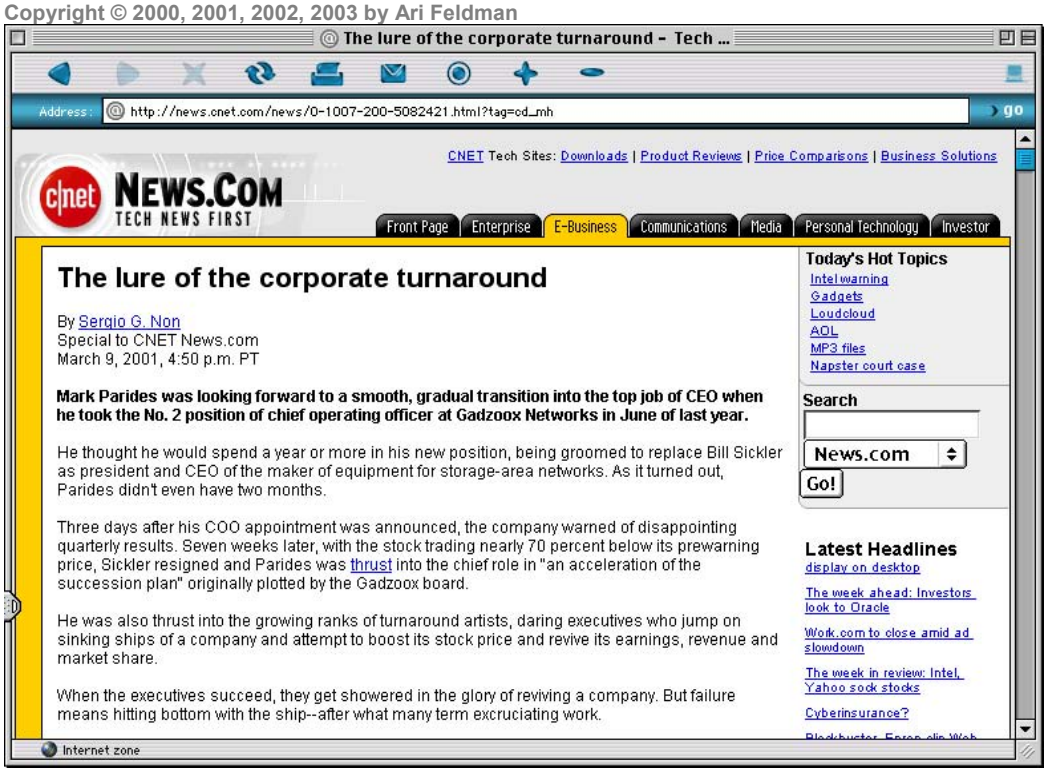
- Web interfaces that are likely to be reused should always be consistent in both their appearance and operation. Doing so minimizes the time a user spends re-learning the interface or its related pages.

- Once the appearance and operation of a web interface have been established, they should be standardized and not be suddenly changed. Abrupt change to the appearance or functionality of an established interface can lead to user confusion.

- Interface elements must always be consistent with their expected behaviors. Inconsistent element behavior can confuse, disorient, frustrate, and even alienate users.

FIGURE 16: Examples of Consistency in a Web Interface



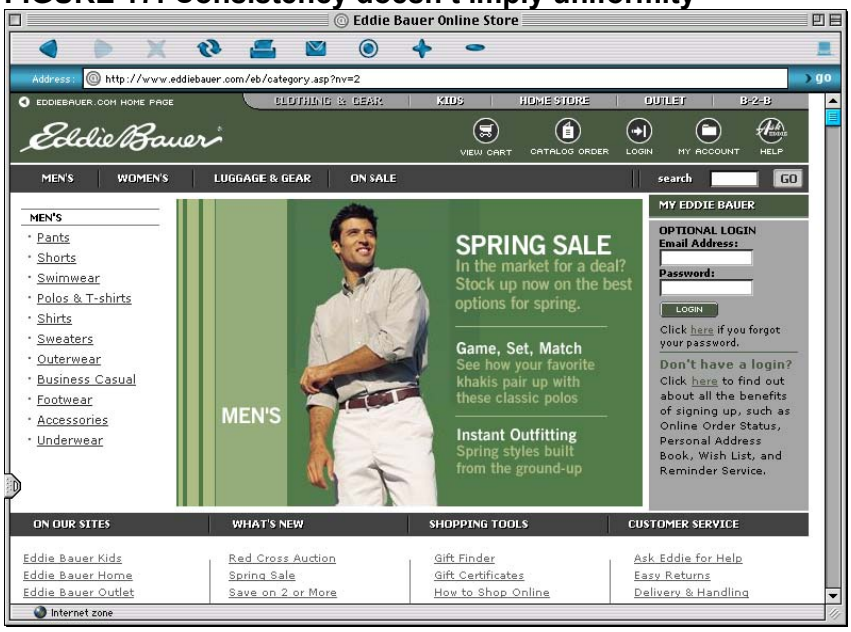


Consistent branding and Top Navigation bar

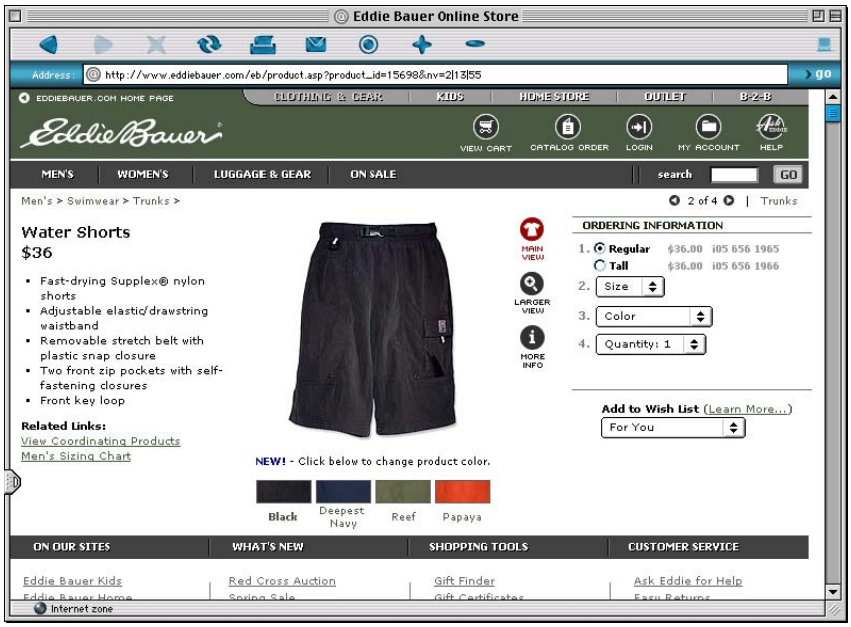
Consistent Side Navigation bar

Figure 16 shows elements taken from two pages of different hierarchical levels of the same web site. Notice the consistent use of color, text size, text links, button style, and menu placement. This consistency aids the user in recognizing the common visual elements and interface behaviors shared by pages and reduce the chances of the user becoming confused and/or disoriented while navigating through a site.

FIGURE 17: Consistency doesn't imply uniformity



Consistency doesn't imply uniformity. In fact, it's perfectly fine for pages within a web site to vary in their look and feel as long as their appearance remains somewhat predictable. Consider the examples in Figure 17. Although they appear to be different from each other, these pages maintain the element of consistency by sharing certain interface characteristics, such as the color scheme and the appearance and location of the top and bottom navigation bars.



Consistent Top Navigation Bar

Consistent Bottom Navigation Bar

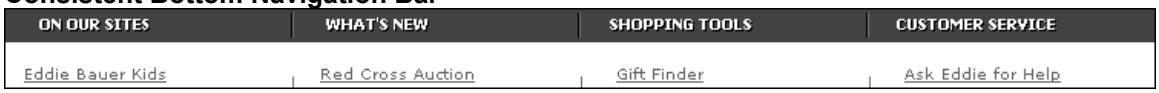
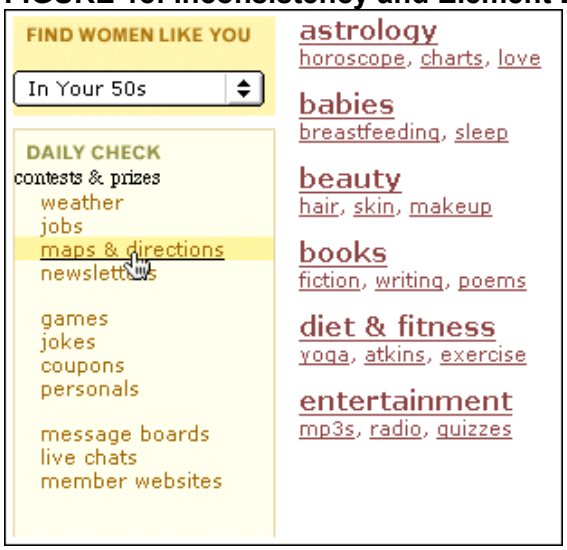


FIGURE 18: Inconsistency and Element Behavior



Inconsistency is a leading cause of user confusion and frustration while using an interface. Consider the problems caused by the inconsistent use of elements in this example: one side of the interface establishes underlined text links as being clickable while the other establishes non-underlined text links as being clickable. This can cause a great deal of confusion since the user's expectation of how the interface works has been drastically altered.

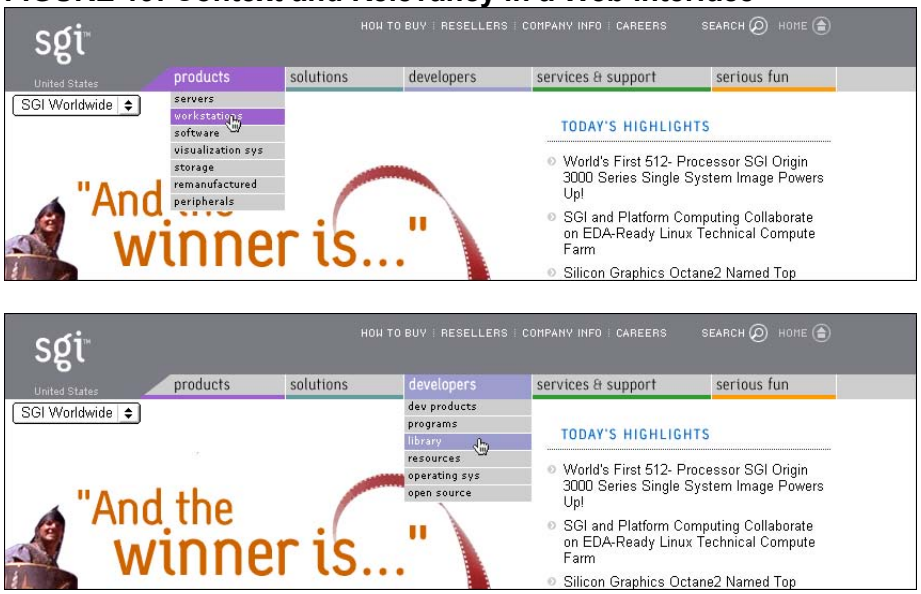
Context

Context makes web interfaces easier to learn and use by:

- Ensuring that the most appropriate options are always presented at the proper time and place, such as displaying menu choices that are relevant to a user's current activity
- Acting as a type of affordance by providing various hints (visual or cognitive) that help users interpret the particular function or purpose of a given interface element.

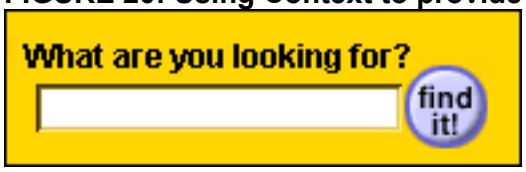
Web interfaces that lack context often take longer for users to learn and use because they fail to provide any helpful interface hints or options that are relevant to the user's actions.

FIGURE 19: Context and Relevancy in a Web interface



The navigation bar of the web site pictured in Figure 19 changes to reveal several context-sensitive options related to the currently selected menu item. The options presented are relevant, making the interface easier to use.

FIGURE 20: Using Context to provide hints to an interface element's function

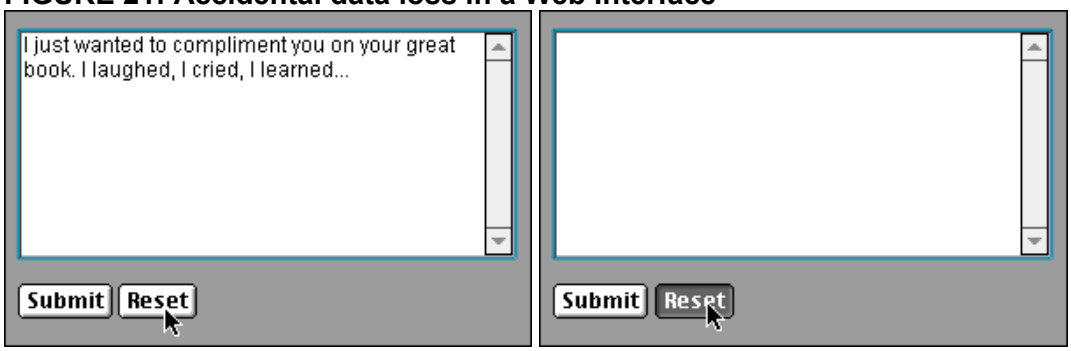


In this Form, both the prompting text and the button label provide context, effectively explaining the Form's function and purpose within the interface.

Data Protection

Given the transient nature of computer information, it's inevitable that some data loss will occur. When it does, it usually happens quickly and without much warning. Accidental data loss, or information lost as a result of a user mistake is the most common form of data loss. It's also one of the leading causes of user frustration (see Frustration earlier in this document) when working with web interfaces.

FIGURE 21: Accidental data loss in a Web Interface



It happens all of the time: the user enters information and accidentally clicks on the wrong button at the wrong time and **Poof**…their data is gone!

NOTE: Data Protection is related to the concepts of Forgiveness and Fitt's Law because they both can be used to minimize accidental data loss.

Default Devices

Any web interface that requires significant user input needs a *default device* to restore the interface back to its original, pristine state when mistakes or errors are made. Unfortunately, this action makes them potentially dangerous interface options. For example, consider Figure 22: if a user accidentally clicks on a default device while entering information, they will lose their data.

FIGURE 22: A Default Device in action

Altered State **Default State**

The “Reset” button (seen in all standard HTML forms) is an example of a default device. Clicking this button will automatically clear all of the user’s input and return the form to its original state.

Feedback

Whenever possible, a web interface should provide timely *feedback* – an acknowledgement that it recognizes a user-initiated action. Without such feedback, users might not be sure that their action has actually been received and processed by the computer. There are two ways to present feedback to the user:

- Audio Feedback
- Visual Feedback

Audio Feedback

Examples of audio feedback include beeps, whistles, buzzes, dings, or even spoken words and phrases. Although sound provides powerful feedback, there are situations where audio cues aren’t appropriate or simply can’t be used, like when the user’s computer doesn’t support sound (many computers in corporate environments don’t) or if the user is hearing-impaired. In addition, large audio files can lead to considerable interface delays.

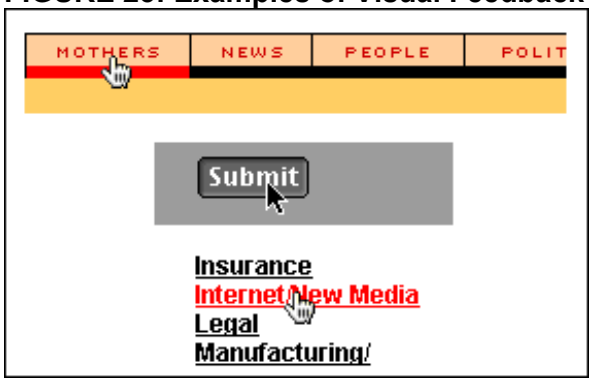
Despite its drawbacks, sound is still very useful in an interface as a secondary feedback mechanism. In fact, operating systems such as Windows and the Mac OS use it extensively for this very reason.

NOTE: Timely audio feedback is not possible using HTML. However, it can be achieved with DHTML, Java, Shockwave, or Flash technology.

Visual Feedback

Visual feedback can take many forms including color changes, animation, text messages, or even changes to an object's appearance. This versatility makes visual feedback the best all-around feedback mechanism, especially for those users with hearing impairments.

FIGURE 23: Examples of Visual Feedback in a Web Interface



As seen here, visual feedback can assume many forms and can greatly reduce the potential for user confusion.

NOTE: Forms are the only HTML interface elements that can provide immediate visual feedback. All other timely visual feedback techniques used in web interfaces require some combination of CSS, DHTML, Java, or Flash technology.

Fitt's Law

Fitt's Law is a qualitative measurement of the distance between objects and targets. As such, it has frequent application in web interface design. The two basic premises of Fitt's Law are:

1. The smaller an object is, the harder it will be to click on.
2. The farther away an object is, the more effort it will take the user to get to it.

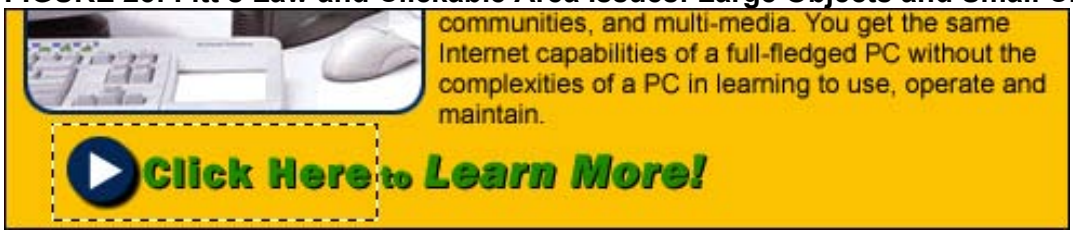
The first premise of Fitt's Law is useful in determining the *clickable* or active target area of a particular interface element. As a rule, smaller objects will have smaller clickable areas than larger objects, which make them more difficult to use. Fitt's Law can apply to any clickable interface element, including text links.

FIGURE 24: Fitt's Law and Clickable Area

The dotted line represents the clickable area of each button. Button A is smaller than Button B. Since Button A has a smaller clickable area than Button B, users are more likely to find Button A harder to click on and more difficult to use than Button B.

There isn't always a direct correlation between an object's size and its clickable area. For example, consider Figures 25 and 26.

FIGURE 25: Fitt's Law and Clickable Area Issues: Large Objects and Small Clickable Areas

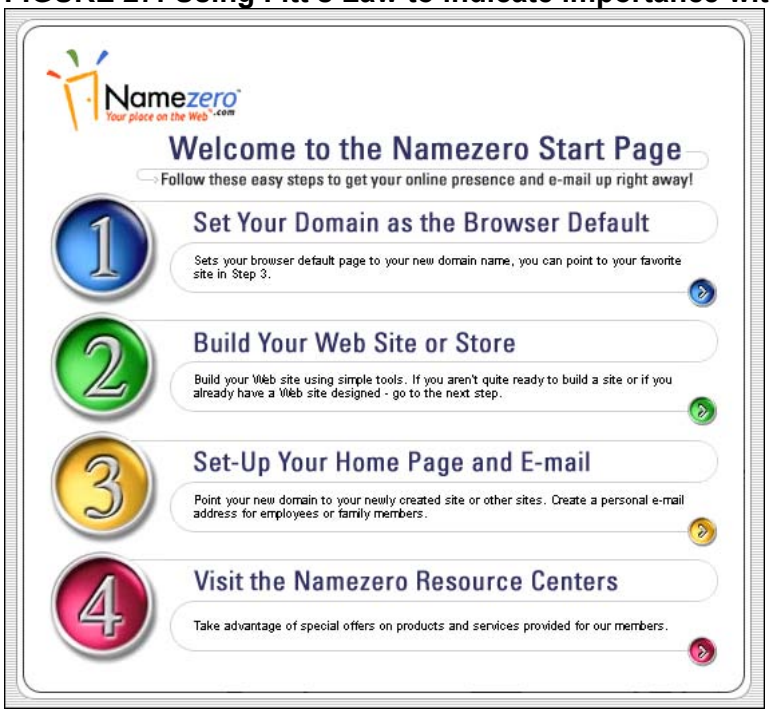


Imagemaps are frequently used by web pages to make irregularly shaped graphic elements clickable. Many imagemaps have relatively small clickable areas even when they are applied to large graphics, as the dotted line indicates. Therefore, some interface elements can be more difficult to use despite their size and screen presence.

FIGURE 26: Fitt's Law and Clickable Area Issues: Small Objects and Large Clickable Areas

Figure 26 illustrates this aspect of Fitt's Law in reverse. Here, the interface element is relatively small but its clickable area is actually quite large.

FIGURE 27: Using Fitt's Law to indicate importance within an Web interface

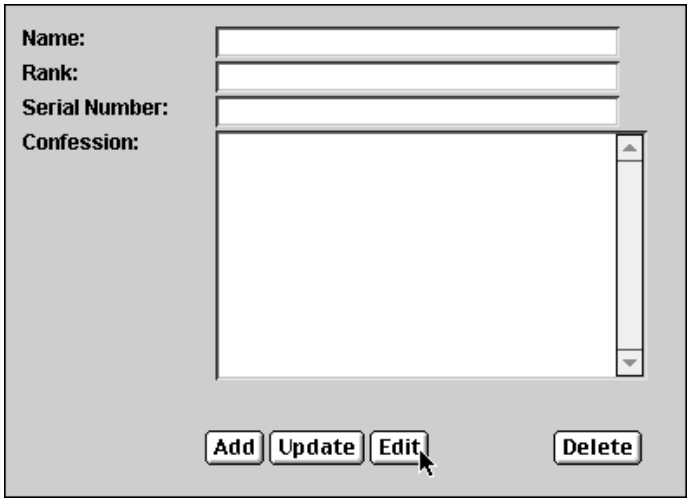


This premise of Fitt's Law shouldn't be interpreted to mean that you should always use bigger objects over smaller objects in your interface. Rather, use Fitt's Law to justify using larger elements when you want to show their importance in an interface. This is because larger elements are more likely to be clicked on than their smaller counterparts, as illustrated by Figure 27.

NOTE: Small "object targets" or clickable areas are one of the most common usability issues found in web site interfaces. Many users with have motor function difficulties (see Accessibility) tend to have problems clicking on such elements.

The second premise of Fitt's Law focuses on the proximity of individual interface elements. Fitt's Law tells us that interface elements that are farther part will require more time and effort on the user's part to access than elements that are closer together. Therefore, one can infer that interface elements that are closer together are also faster to access and easier to use. This aspect of Fitt's Law can have a profound impact on how you choose to group the elements in an interface, as shown in Figure 28.

FIGURE 28: Fitt's Law and Interface Element Proximity

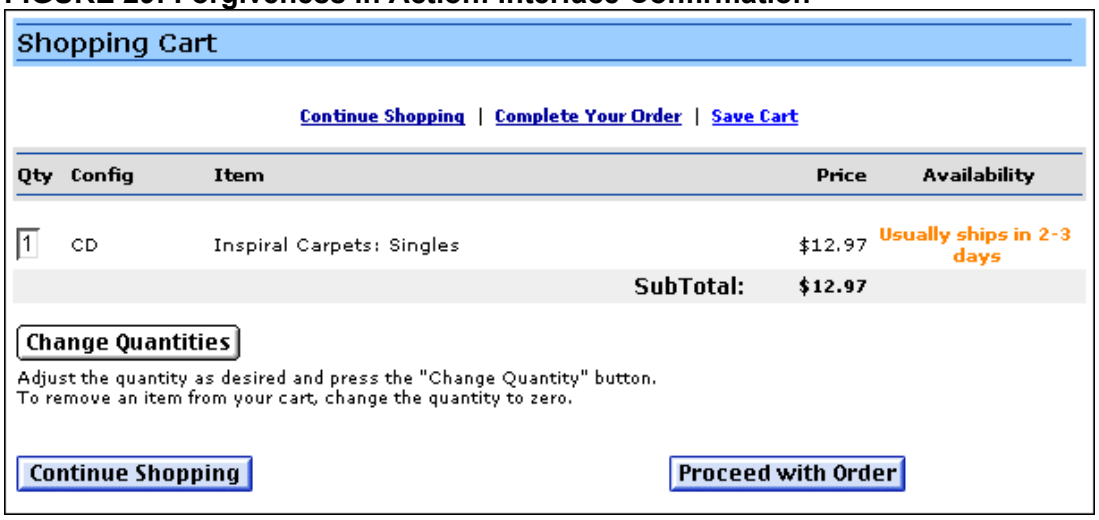


According to the second premise of Fitt's Law, frequently used options should be placed close together so users can access them faster while infrequently used options can be placed further away. By extending the same logic, you can make web interfaces safer by placing potentially dangerous options like "Reset" buttons out of the user's reach.

Forgiveness

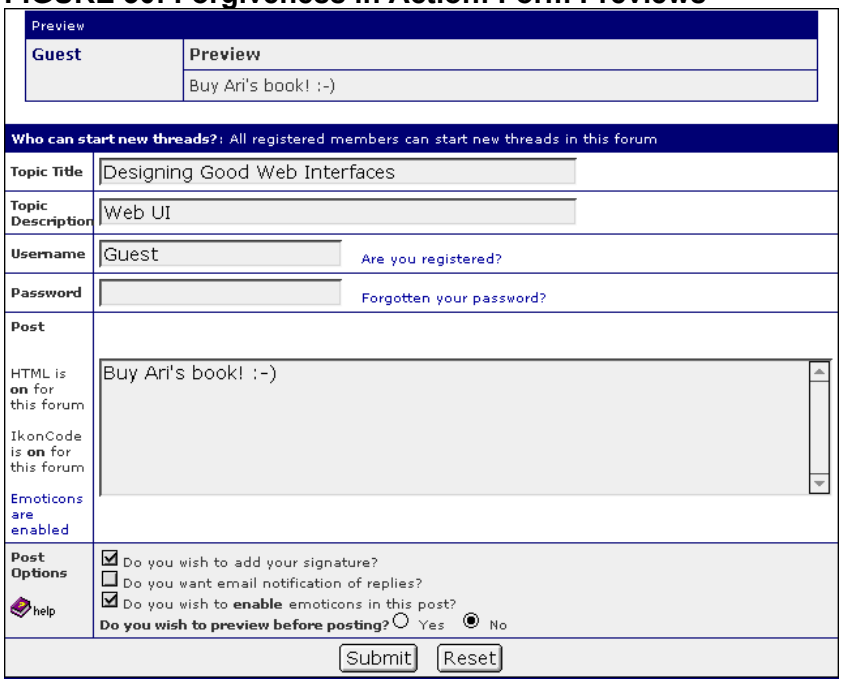
In a web interface, *forgiveness* provides users with a graceful way out of a potentially bad situation. It prevents users from taking foolish, harmful, or even dangerous actions and even gives them a chance to back out *before* things go too far.

FIGURE 29: Forgiveness in Action: Interface Confirmation



Risky or important interface actions should always be preceded with some type of *confirmation*. Confirmations warn the user of potential dangers, they give the user chance to re-think the planned action, and they offer the user a way out of the situation. Figure 29 illustrates how many E-Commerce sites use confirmations to make sure that users are satisfied with their product selections prior to completing their transactions.

FIGURE 30: Forgiveness in Action: Form Previews



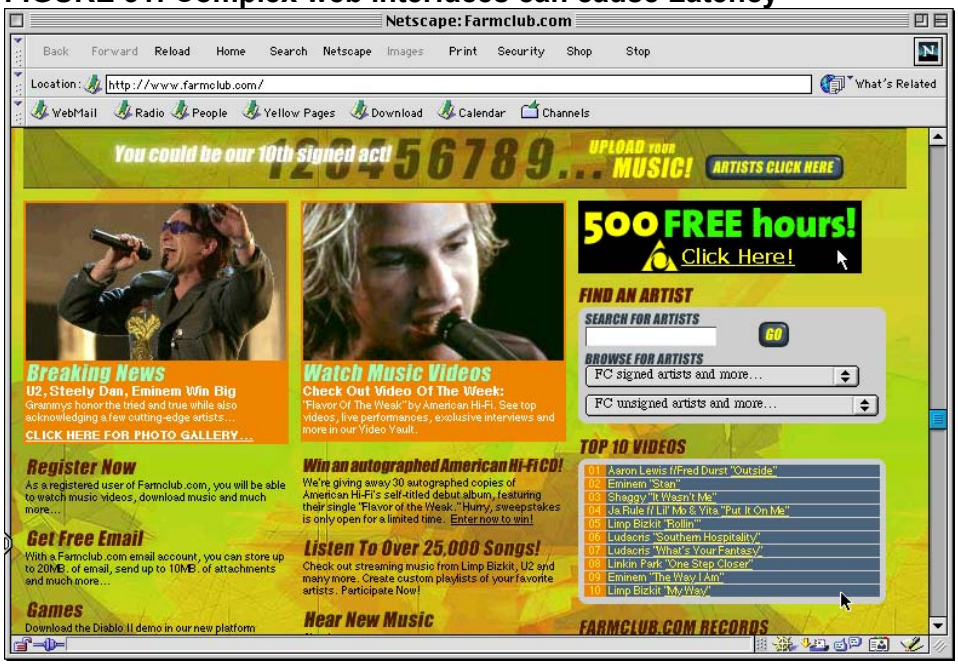
HTML Forms that allow users to preview their contents prior to submitting them are also examples of Forgiveness. In the message board Form shown in Figure 30, the user has a chance to preview and edit their posting before they submit it.

Latency

Latency is a fancy term for any computer-generated delay. Computer users want immediate responses to their commands and actions. In fact, studies have shown that users can become bored or distracted within 10 seconds of experiencing latency. For obvious reasons, this isn't something we want to happen.

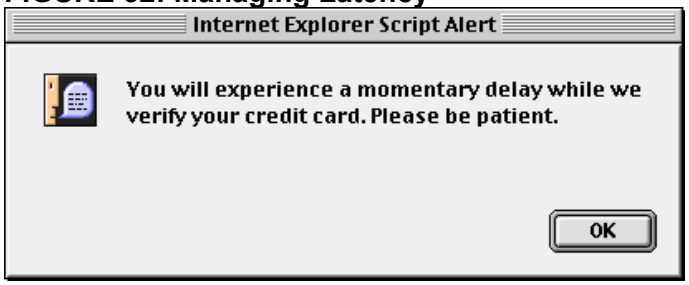
Unfortunately, almost any computer-related activity can cause latency. Web pages and interfaces are particularly prone to latency, especially if they contain complex layouts, many graphical elements, or use technologies such as Flash or Java that bog down their display.

FIGURE 31: Complex web interfaces can cause Latency



Due to its complexity and heavy use of graphical elements, the web page shown in Figure 31 creates a lot of latency, especially for those users with slower Internet connections (see Accessibility).

FIGURE 32: Managing Latency

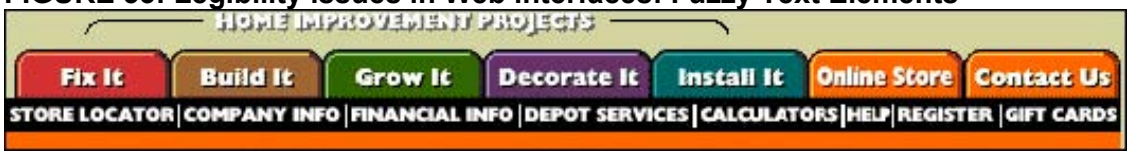


The best approach to managing latency is to alter the user's perception of it, particularly if long delays are involved. There are several ways to accomplish this, including simplifying the interface's design, limiting the use of external technologies such as Java or Flash, or by providing meaningful status messages like the one shown in Figure 32.

Legibility

Legibility measures the clarity and distinctiveness of individual interface elements. Although traditionally applied to text, legibility can actually be a function of any interface element, including buttons and icons. Interface elements that are overly decorative, fuzzy, blurry, improperly sized, or use jarring color combinations can be difficult to distinguish on-screen and can lead to user confusion, frustration, and inefficiency.

FIGURE 33: Legibility issues in Web interfaces: Fuzzy Text Elements



Fuzzy text can often be difficult to read on-screen, especially for users with low vision (see Accessibility).

FIGURE 34: Legibility issues in Web interfaces: Overly Decorative Text Elements

Overly decorative text tends to be very difficult for most users to read on-screen, thus negating much of its usefulness in an interface to convey information.

FIGURE 35: Legibility issues in Web interfaces: Poorly Designed Buttons and Icons

Buttons or icons that are unusually small, have poorly chosen color schemes, shading, or that have fuzzy captions and labels can prevent users from comprehending their meaning and purpose.

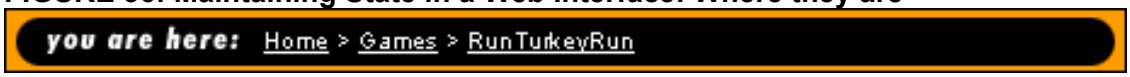
Maintaining State

When you *maintain state* in a web interface, you are helping the user to keep track of various interface-related events, including:

- Where they are
- What they've done
- What's currently happening

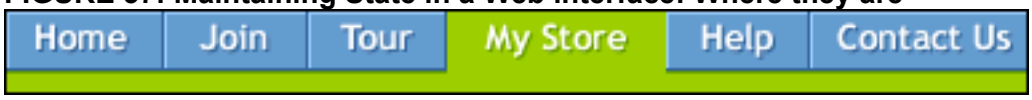
Web interfaces that keep track of these actions are more user friendly, more responsive, less confusing, and reduce the likelihood of the user repeating their work (*read:* they increase user efficiency).

FIGURE 36: Maintaining State in a Web Interface: Where they are



Many sites use *navigation paths* (also called navigation strings or navigation pipes) to show the user where they currently are. Such mechanisms maintain state by preventing users from becoming confused or disoriented as they navigate around a large web site.

FIGURE 37: Maintaining State in a Web interface: Where they are



Many smaller sites use simple graphical indicators like the one shown in Figure 37 rather than navigation paths to show users where they currently are.

FIGURE 38: Maintaining State in a Web Interface: What they've done

When clicked, text links can change color to provide the user with a visual trail of where they have been within a given web site. In Figure 38, the magenta text link represents areas of the site that the user has already visited, thus maintaining state by showing users what they've already done. This helps them to avoid duplicating work and increases their efficiency.

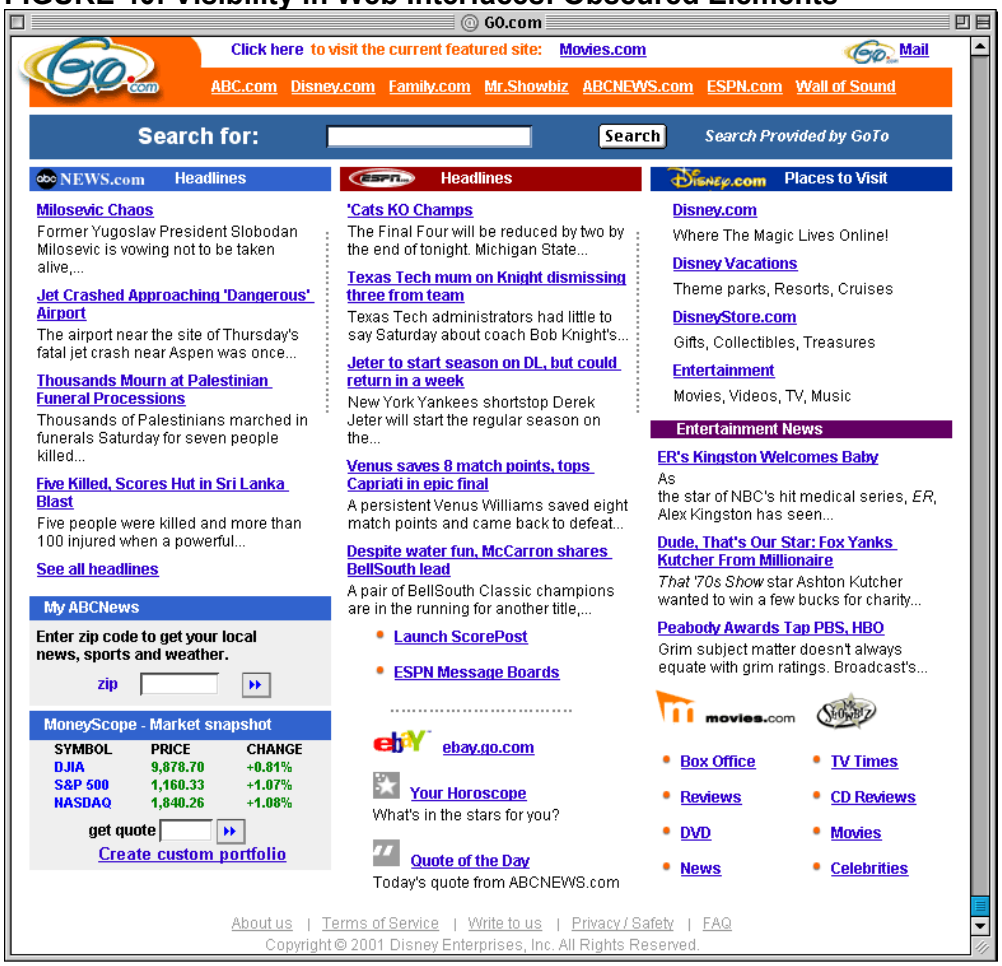
FIGURE 39: Maintaining State in a web interface: What's happening

Whenever possible, a web interface should always tell the user what's currently happening to prevent potential user confusion. Progress bars (as shown in Figure 39) are one way to do this. They maintain state by showing the user exactly what's going on in real-time.

Visibility

Visibility is a type of affordance, since visible interface elements facilitate user interaction. Visibility isn't necessarily a binary state – i.e. visible or not visible. In a web interface, an element can be considered invisible simply if the user can't see it. Many web interfaces unintentionally (or intentionally) obscure or bury elements, making them difficult for users to find on-screen. From the user's perspective, such interface elements might as well not exist.

FIGURE 40: Visibility in Web Interfaces: Obscured Elements



Bottom Text Links

In Figure 40, certain text links such as the "Terms of Service", "Write to Us", and "Privacy Statement" are located at the very bottom of the web page, effectively concealing them from the user. To make matters worse, these elements are rendered in a light grey color, obscuring them even further.

FIGURE 41: Visibility in Web interfaces: Hidden Elements

The "Home" icon

In this example, the "Home" icon in this interface is extremely small, effectively hiding it from all but the most observant of users.

Audience Considerations

The topics discussed here aren't interface design concepts *per se* but are instead design-related considerations that can be used to make a web site usable by the broadest possible audience The topics covered in this section include:

- Accessibility
- Internationalization
- Metaphors

Accessibility

Whenever possible, a web site interface should be made *accessible*, or able to accommodate users with specific disabilities and/or special needs. Accessibility accomplishes two important things: first, it makes it easier *to* use a given web interface and second, it increases the number of people who *can* use a given web interface.

Adding accessibility to a web site interface isn't just a nice courtesy; it's actually the law in several countries, including the United States (i.e. The 1996 Americans with Disabilities Act). As a designer, you can expect a certain percentage of users to come to a web interface with *physical disabilities* or *material disadvantages*. In some cases, there will be users with both.

Physical Disabilities

The more serious of the two, physical disabilities can include such problems as:

- Visual Impairments
- Auditory Impairments
- Motor Function Disabilities
- Cognitive Disabilities

Visual impairments

Visual impairments are one the most widespread of all user disabilities. In fact, as of 1998, approximately 3.5% of all Internet users were classified as being visually impaired. Users with visual impairments run the gamut from experiencing slightly reduced visual acuity to total blindness.

FIGURE 42: Text size and users with minor visual impairments

Dow	10618.08	-24.45▼
Nasdaq	2250.54	-57.96▼
S&P 500	1265.74	-1.91▼
Russell 2000	484.33	-3.98▼
NYSE	633.99	-0.50▼

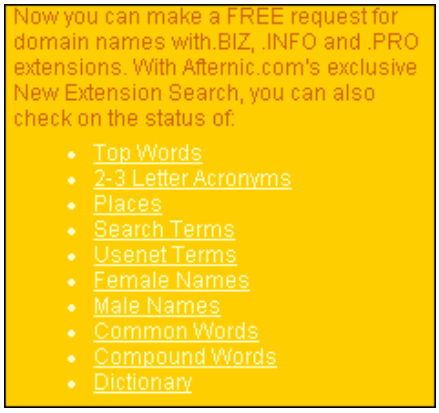
Before accessibility is considered

Dow	10618.08	-24.45▼
Nasdaq	2250.54	-57.96▼
S&P 500	1265.74	-1.91▼
Russell 2000	484.33	-3.98▼
NYSE	633.99	-0.50▼

After accessibility is considered

Users who suffer from slight visual impairments often find it difficult to read small text and/or experience eyestrain at the end of long stints in front of the computer. The ability of these users to interact with a given interface can be greatly enhanced if the size of text elements is increased.

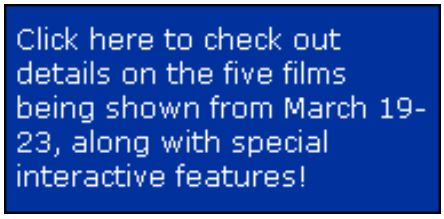
FIGURE 43: Visual concerns of users with Low Vision: Poorly Designed Elements



Poor Contrast: Text

Poor Contrast: Button

FIGURE 44: Visual concerns of users with Low Vision: Well Designed Elements



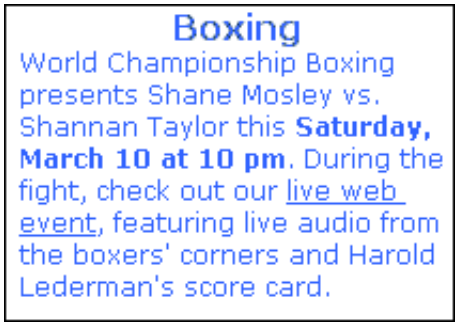
Good Contrast: Text

Good Contrast: Button

Users with vision equal to or below 20/80 are classified as having low vision. These users require a higher level of contrast between foreground and background elements, especially for text and small button elements. Figures 43 and 44 show examples of both problematic and properly designed interfaces elements.

NOTE: Users with vision below 20/200 are considered blind and may require converting interface information into spoken words.

FIGURE 45: Visual Concerns with Blue



Using certain shades of blue to display large amounts of text can cause problems for users regardless of their vision. This is because blue is a relatively difficult color to see especially when placed against a lightly colored background.

Other types of visual impairments users can experience include a reduced field of vision, a condition that limits a person's focus to only a small area, and *colorblindness*. Colorblindness is of particular interest since it's estimated that as many as 8% of all men and 0.5% of all women have some form of it.

FIGURE 46: What users with normal color vision see

Colorblindness is a physical condition that occurs when one or more of the types of cones inside the eye that are responsible for perceiving color fail to function correctly. When this happens, the viewer can't properly distinguish certain colors or combinations of colors. The most commonly experienced form of colorblindness is known as *red-green color perception deficiency*. Figures 46 and 47 demonstrate how users with normal vision and colorblindness actually see certain objects.

FIGURE 47: What users with various forms of colorblindness see

NOTE: Colorblind users may experience problems distinguishing between certain colors used in a web interface design. Accommodate them by always providing clear, secondary cues (see the section on Affordances) that do not rely exclusively on color to convey essential interface information.

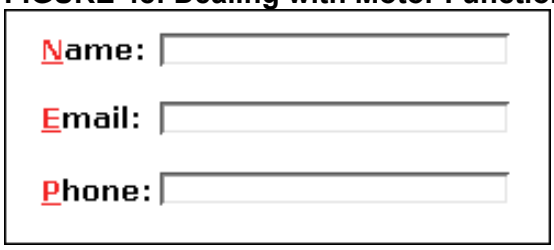
Auditory Impairments

According to 1998 survey data, about 1.7% of Internet users experienced some form of hearing or auditory impairment. These users may have to rely more heavily on a web interface's visual cues to successfully interact with it. Such cues may include animation or the liberal use of color and text.

Motor Function Disabilities

As many as 2% of all Internet users in 1998 had some form of motor function-related disability, making them unable to perform certain manual tasks, such as moving a mouse or typing at a keyboard. Other users may be able to do both but only at a reduced level of efficiency. To accommodate such users, a web interface needs to be flexible in how it receives and processes user input. Whenever possible, the user should be able to choose their input method and never be forced into using the mouse over the keyboard or vice-versa.

FIGURE 48: Dealing with Motor Function Disabilities in a Web Interface



Accesskeys, a method of accessing HTML Forms entirely with the keyboard, is an excellent way of making an interface accessible to such users.

Cognitive Disabilities

Users with cognitive disabilities include those with short-and long-term memory loss, perception problems, and developmental disabilities. In 1998, approximately 0.5% of all Internet users fell into one or more of these categories. Other types of cognitive disabilities exist as well. These include learning impairments like dyslexia, illiteracy, or even foreign users learning the native language used by an interface (see Internationalization). Web interfaces can often be made more accessible to these users by providing better and easier to understand metaphors (see Metaphors).

Material Disadvantages

Unlike those who suffer from physical impairments, users with material disadvantages suffer from a variety of equipment-related issues, including:

- Inferior Hardware
- Obsolete Software
- Poor Connectivity

Inferior Hardware

Users with inferior hardware access web interfaces with obsolete or underpowered computer equipment. This can mean anything from a having slow machine, which increases computer-related delays (see Latency) to having a small, monochrome, or text-only screen, which can affect the appearance of certain visual elements within a given interface.

FIGURE 49: Inferior Hardware and Web Interfaces

Users with monochrome screens can't see color information, which can cause them to miss important visual cues that could prove essential to their understanding and use of the interface. Without color, how do users know which areas in this interface are important? How do they know which links they've visited?

Obsolete Software

Users with obsolete software run versions of web browsers and operating systems that are anywhere from one to three versions behind the current industry standards. Such users can experience a variety of issues, including information display and latency problems due to bugs, poor performance, or missing features.

FIGURE 50: Obsolete Software and Web interfaces

Lynx was developed in the early 1990s as one of the first web browsers and can't display graphics. Yet, despite being obsolete, it continues to be used because it doesn't require a graphics-capable screen and because it's very fast at accessing information, especially over slower Internet connections (see Poor Connectivity).

Poor Connectivity

Despite the significant growth and improvements made in Internet connectivity in recent years, the vast majority of users are still accessing the Internet at 56Kbps speeds or less. Therefore, you can expect large numbers of users to experience delays while using web-based interfaces, particularly those that contain many graphical elements, use "rich" technologies such as Java, Shockwave, and Flash, or perform significant amounts of back-end processing.

NOTE: You are more likely to encounter material disadvantages among users in public schools, public universities, public libraries, small businesses, a good number of homes, and in developing countries. Therefore, in order to make your web interfaces truly accessible; avoid adding unnecessary "bells and whistles" and always design for the lowest common denominator.

Internationalization

With the Internet now reaching millions of users beyond the shores of the United States, it's essential that your web site interfaces be *internationalized*: both usable by and accessible (see Accessibility) by a global audience.

Internationalization is a process for developing culturally "neutral" web interfaces. In order for a web interface to be considered "internationalized", it must make accommodations in a number of areas, including:

- Text Translation
- Element Layout
- Graphical Symbols
- Colors
- Formatting

Text Translation

Translating from English is one of the most difficult aspects of internationalizing a web interface for two reasons. First, many foreign languages have different spellings, articles of use, characters, verbs, and adjectives than English. Second, many English terms and phrases have no direct foreign equivalents, including such common web site jargon as "upload", "e-mail", and "login". Either issue can require text elements, button labels, or icon captions to be completely re-written and re-designed.

Text length tends to be one of the most difficult aspects of translating text for a web interface. Consider the examples shown in Table 2.

TABLE 2: Length Issues with the Internationalization of Text

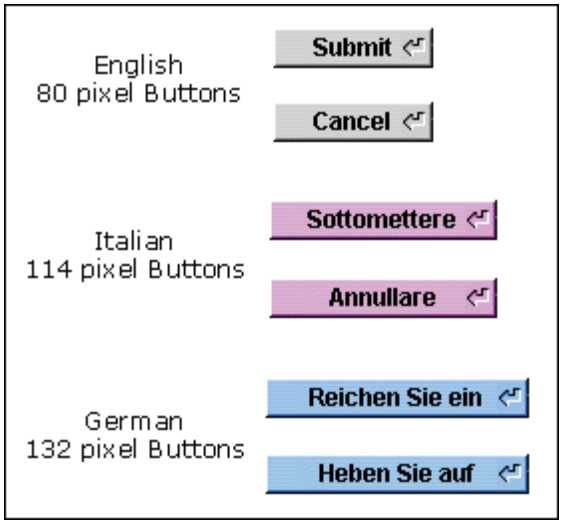
Language	Interface Jargon	# of Characters
English	Shopping Cart	13
French	Charrette commerciale	21
German	Einkaufenden Wagen	18
Portuguese	A Carreta de compras	20
Spanish	El Carrito que hace de compras	30
Language	**Interface Jargon**	**# of Characters**
English	Register Now	12
French	Enregistrer Maintenant	22
German	Registrieren Sie Jetzt	22
Portuguese	Registre Aagora	15
Spanish	Registre Ahora	14
Language	**Interface Jargon**	**# of Characters**
English	Account Information	19
French	Expliquer l'Information	23
German	Begründen Sie Informationen	27
Portuguese	A Informação de conta	21
Spanish	Dé cuenta Información	21

Foreign phrases often contain more characters than their English counterparts, so you may have to adapt the design and placement of textual elements within a web interface accordingly.

Element Layout

Because the translation of text elements into other languages can require more space within the interface, it's likely that the size and position of certain interface elements, such as buttons and forms will also have to change.

FIGURE 51: International Issues with Element Layout



Foreign languages that use the Roman character set often require more space than English does. Non-western languages may require even more room. These differences can create all sorts of havoc with the layout of fixed-size graphical interface elements such as buttons and icons.

Graphical Symbols

Graphical symbols pose particularly difficult problems for internationalizing web interfaces, as many American graphic symbols don't translate very well into other cultures. For example, common hand gestures can actually prove offensive in some foreign countries. Similarly, many graphic symbols used to represent animals, religious activities, holidays, and even sporting events can create translation problems due to the vast cultural differences in both their interpretations and meanings.

FIGURE 52: Internationalization Issues with Graphical Symbols

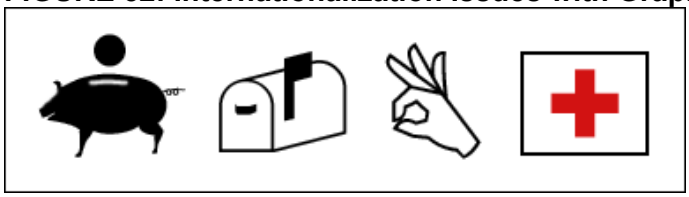
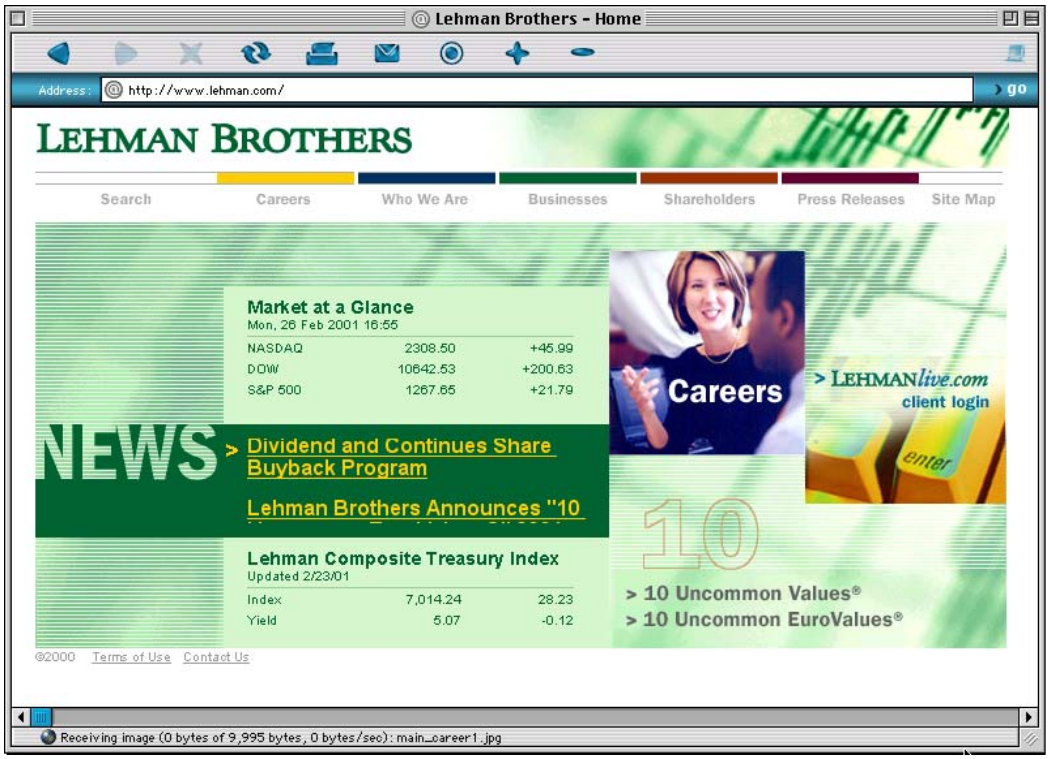


Figure 52 illustrates the problem very eloquently. Pigs in any guise are offensive to Muslims; the appearance of mailboxes varies throughout the world, the "Ok" gesture turns out to be a major insult in South America, and crosses have no real significance to over four billion people.

Colors

Colors have different cultural meanings. Although few color selections will prove offensive, many have the potential to communicate conflicting and/or misleading messages to foreign users. Therefore, colors must be chosen carefully and intentionally in order to convey the right message as illustrated by Figure 53.

FIGURE 53: Internationalization Issues with Colors



In western cultures, the color green symbolizes wealth while in middle-eastern cultures it symbolizes strength. Both meanings have very positive connotations that are especially important to this investment bank, which draws its elite clientele from users in both cultures.

Table 3 shows some of the more common cultural associations for colors among different regions and countries.

TABLE 3: Common Cross-cultural Color Associations

Color	Western Europe & USA	Eastern Europe & Balkans	Japan	China	Brazil	Middle East	Nigeria
Red	Danger Excitement	Communism Beauty	Passion Strength	Communism Festivity	Anger Vibrancy	Danger Evil	Death
Yellow	Cowardice Caution	Nature	Sunshine Nature	Power Masculinity	Wealth	Happiness Prosperity	Sunshine High Status
Blue	Trust Peace	Hope Peace	Villainy Cold	Strength Immortality	Happiness Freedom	Protection Virtue	Calm Peace
Green	Wealth Masculinity	Nature	Energy Youth	Desirability Youth	Hope Wealth	Strength Fertility	Wealth Fertility
Black	Death Elegance	N/A	Mystery Anger	Life Stability	Death	Mystery Evil	Maturity Masculinity
White	Purity Good	Intelligence Neatness	Death	Death Mourning	Sophistication Authority	Purity	Purity

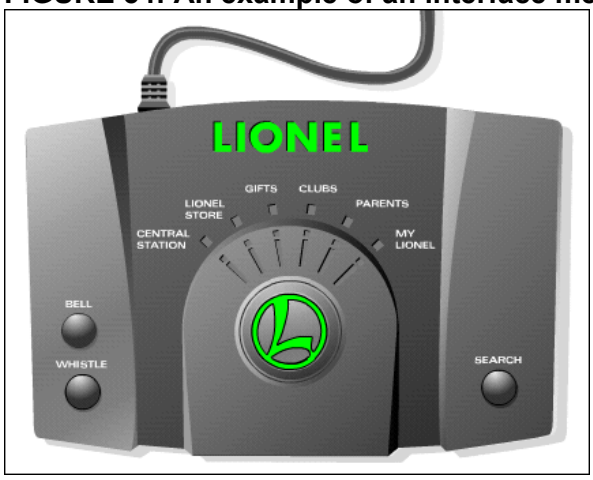
Formatting

Date styles, numbers, times, addresses, telephone numbers, currencies, calendars, paper sizes, and units of measure can all vary from country to country and region to region. In some cases, these differences are minor while in others they are quite considerable. Your interfaces, particularly those that make use of Form elements, should take into account of these differences if they are to be truly accessible and usable by foreign users.

Metaphors

Metaphors are devices used by interface designers to provide visual equivalents for various concepts and ideas that can't be easily or directly explained. They are particularly useful when space within the interface is limited.

FIGURE 54: An example of an interface metaphor



When it comes to metaphors, a picture *is* worth a thousand words. For example, It's much easier to use a graphical metaphor to represent a remote control than it is to explain the concept of one.

Metaphors rely almost exclusively on items drawn from the user's memory, understanding, and perception in order to work. If chosen correctly, metaphors can be incredibly effective in helping users to immediately understand the functions and options available in a given web interface. Unfortunately, badly chosen metaphors can also be a major source of user confusion. Therefore, two issues should be carefully considered when choosing a metaphor for an interface:

- User Understanding
- User Interpretation

User Understanding

Visitors come to web sites with varying degrees of computer and life experience. Since metaphors are drawn from the user's own memories and experiences, you can easily develop an interface metaphor only to discover that your target user doesn't actually understand it or its intended purpose.

FIGURE 55: Problems with User Understanding of Metaphors: General Audiences

Disk drives are found in over 170 million computers worldwide. Consequently, diskettes are used by many interfaces (both web and non-web) to symbolize file-saving functions. However, if the user is new to computers and doesn't know what a diskette is or what to do with it, the intention and function implied by this metaphor will be completely lost on them.

FIGURE 56: Problems with User Understanding Metaphors: International Audiences

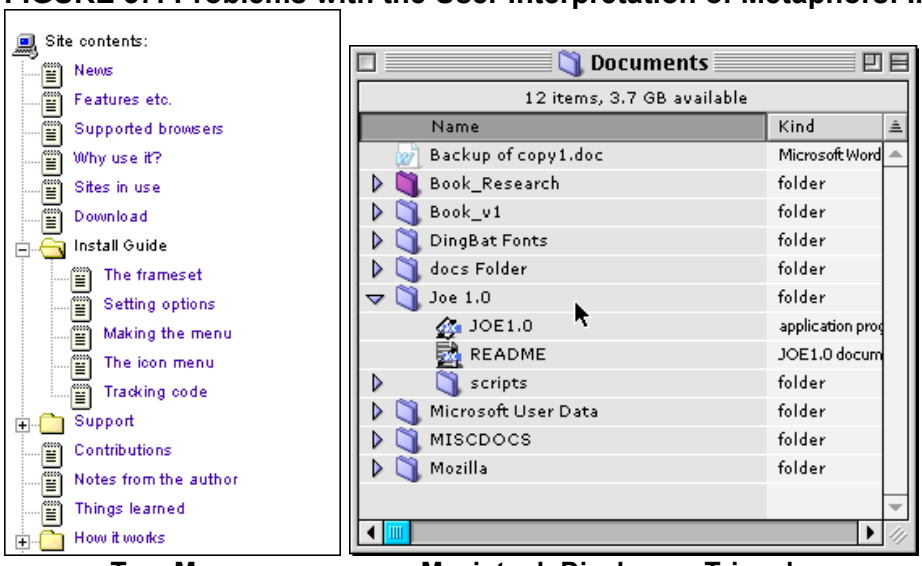
Trashcans such as the one pictured here are frequently used in interfaces (both web and non-web) to symbolically represent a way to delete files. However, in many Asian countries, wicker baskets are used to store garbage rather than steel cans. Therefore, users from these countries may not actually understand the intended purpose behind this particular metaphor.

User Interpretation

You can go to the trouble of creating an elaborate interface metaphor only to discover that users won't interpret its meaning in quite the same way you do. Remember that users will come to an interface with a variety of faculties and life experiences. Consequently, what they think and know won't always be in line with *your* expectations and *vice-versa*.

Figures 57 through 59 illustrate some of the most common problems of user interpretation of interface metaphors.

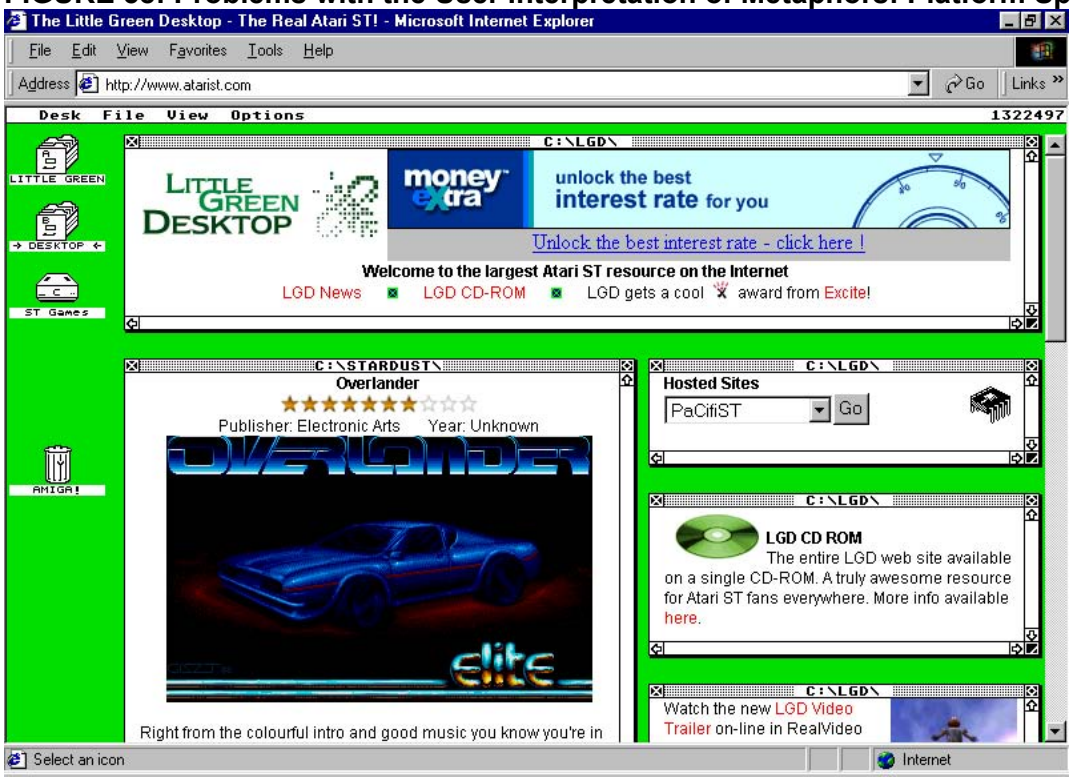
FIGURE 57: Problems with the User Interpretation of Metaphors: Interface Semantics



Tree Menu **Macintosh Disclosure Triangles**

Figure 57 is a classic example of how interface semantics can cause problems with how a metaphor is interpreted by a user. This example shows a "tree" menu, which is a standard interface metaphor in Windows and Java applications. It's also frequently used as a navigational device by many web sites. However, on the Macintosh, tree menus don't exist but disclosure triangles do. Although disclosure triangles perform essentially the same function as tree menus, they're actually a completely different metaphor. Therefore, users familiar only with the Macintosh interface may experience problems using tree menus.

FIGURE 58: Problems with the User Interpretation of Metaphors: Platform Specificity



Platform-specific metaphors like the one pictured in Figure 58 make notoriously bad web interfaces for two reasons:

1. They discriminate against users who aren't knowledgeable about a particular platform – in this case GEM (Graphical Environment Manager), a desktop interface that was an alternative to Microsoft Windows during the late 1980s and early 1990s. However, since GEM's popularity has long since waned, very few users will be familiar with it, let alone understand how to use or navigate around this interface.

2. They seldom work like their "real" desktop counterparts. For example, you can't drag around files or close windows in this implementation of the interface as you could on a real machine running GEM; therefore, you risk confusing users who might not understand these differences.

FIGURE 59: Problems with the User Interpretation of Metaphors: Graphic Symbols

Depending on their past experiences, some users will interpret this icon as a "search" function while other users will interpret it as "zoom" function. In fact, either interpretation is valid, making this metaphor a poor choice because its actual meaning within the interface is ambiguous and unclear.

Although problems with understanding and interpreting metaphors can never be eliminated, their impact can be reduced by carefully coordinating the selection of your metaphors with the mental and cultural experiences of your audience. Figures 60 through 62 provide some examples of some particularly well executed interface metaphors.

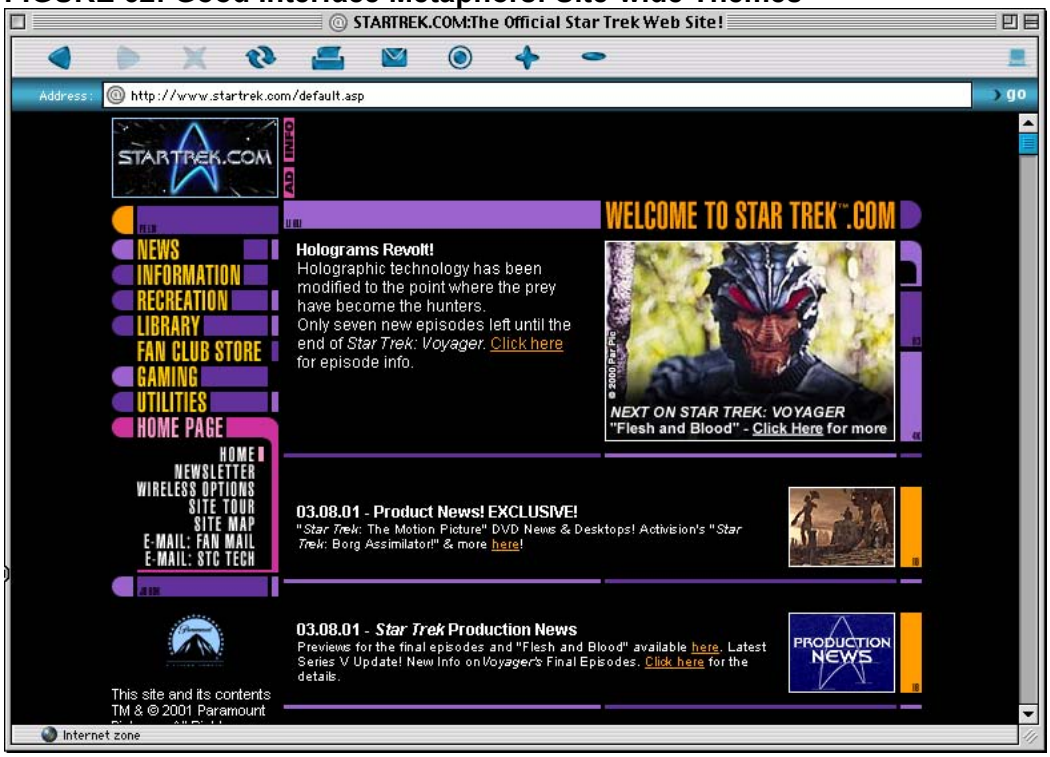
FIGURE 60: Good Interface Metaphors: Icons

These icons are all good interface metaphors because they're based on most users' life experiences and have associations with familiar and largely universal "real world" objects. For example, the *Tracking* function is represented by a barcode symbol, the *Rates* function is represented by a calculator, and the *Transit Time* function is represented by a wall clock.

FIGURE 61: Good Interface Metaphors: Tabbed Interfaces

Tabs are very popular web interface metaphors because they're based on familiar, every-day objects – the paper divider tabs used in many homes, schools and offices. This relationship makes their purpose in an interface immediately understood, even by those users with relatively limited computer experience.

FIGURE 62: Good Interface Metaphors: Site-wide Themes



Some web sites use thematic metaphors in their interfaces to make themselves more visually interesting. This interface makes extensive use of "Okudagrams" or the stylistic trappings used by Star Trek-era display terminals. To a Star Trek fan, they are instantly recognizable and enhance the user's experience. Yet, despite being very specific to the genre, this metaphor doesn't discriminate against non-"trekkies" and is actually usable by all types of users.

Document References and Bibliography

Albers, Josef. *Interaction of Color*. New Haven and London: Yale University Press, 1975

Apple Computer, Inc. Macintosh Human Interface Guidelines. Addison-Wesley Publishing Company, 1992

Berkun, Scott. Fitt's UI Law Applied to the Web. Microsoft Corporation, May/June 2000

Birren, Faber. *Color and Human Response*. New York, NY: John Wiley & Sons, Inc., 1978

Fernandes, Tony, *Global Interface Design: A Guide to Designing International User Interfaces*, AP Professional, Boston, 1995

Galitz, Wilbert O. *The Essential Guide to User Interface Design*. New York, NY: John Wiley & Sons, Inc., 1997

Georgia Tech University, GVU's WWW User Survey: Disability, Internet. 1998

Horton, William, *The Icon Book: Visual Symbols for Computer Systems and Documentation*, New York, NY: John Wiley & Sons, Inc., 1994

Howlett, Virginia., *Visual Interface Design for Windows*. New York, NY: John Wiley & Sons, Inc., 1996

Microsoft Corporation. The Microsoft® Windows® Guidelines for Accessible Software Design. Internet. 7 May 1997

Miller, George A. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information., *The Psychological Review*, 1956, vol. 63, pp. 81-97

Murrell, K.A., Human Computer Interface Design in a multi-cultural multi-lingual environment. 1 July 1998. University of Natal – Durban, South Africa

Nielsen, Jakob. *Usability Engineering*. Boston, Academic Press, 1994

Norman, Donald A. *Affordances and Design.* Internet. 1998

Norman, Donald. A. *The Psychology of Everyday Things.* New York, NY: Basic Book, 1988

Raskin, Jef. *The Humane Interface*. Addison-Wesley, 2000

Spolsky, Joel. User Interface Design for Programmers 4: Chapter 4: Affordances and Metaphors. Internet. 18 April 2000

Tognazzini, Bruce. First Principles. Internet. 2000

Web of Culture. Worldsmart: Colors. Internet, 2000

Wilson, Diane. Color Vision Color Deficiency. Internet. 2000

Document Index

L

M

N

O

P

R

S

T

U

V

W