UNIT-3
Fluid Navigation

Introduction

- Navigation is key to successfully operating interactive applications, such as installing a mobile app, filling in a survey, or purchasing a train ticket (task navigation). It is also the key to finding information on a website or browsing social media (web navigation) or to finding the action needed in a desktop application (command menu navigation).

- Navigation harnesses users' ability to rapidly skim choices, recognize what is relevant, and select what they need to realize their intentions. The goal for designers is to enable fluid navigation that allows users to gracefully and confidently get to where they want to go, explore novel possible routes, and backtrack when necessary. Navigation depends on recognition of landmarks that travelers use to guide their choices, which differs greatly from search, which requires users to describe what they want by typing keywords in a blank search box.

- While the search box is the main technique to initiate the process of finding information in vast information spaces (like the internet or digital libraries), navigation techniques such as small or large menus, embedded links, or tool palettes are the workhorses of navigation. Users indicate their choices with a touch, tap, or swipe of the finger or by using a pointing device.

Navigation by Selection

Choices can be presented explicitly, in that there is an orderly enumeration of the items with little extraneous information, or they can be embedded in text or graphics and still be selectable. Embedded links of web pages were first popularized in the Hyperties system (Koved and Shneiderman, 1986), which was used for early commercial hypertext projects and became the inspiration for the hotlinks of the World Wide Web. Highlighted names, places, or phrases became menu items embedded in text that informs users and helps to clarify the meaning of the menu items.

Graphical techniques are a particularly attractive way to present choices while providing context to help users specify what they want. For example, maps can orient users about the geography of the area before users select an item of interest, and calendars or timelines can inform users of availability and constraints before a date or time is selected

- The simplest case of explicit menus is a binary menu for yes/no, true/false choices .Another example of a simple menu is the grid menu popularized by mobile devices, with a small set of icons and labels).

- When users need to make a series of choices (e.g., in a survey or to select parameters of an application), there are well-established methods of presenting choices. Radio buttons support single-item selection from a multiple-item menu, while check boxes allow the selection of one or more items in a menu.

- A multiple-selection menu is a convenient method for handling multiple binary choices, since the user is able to scan the full list of items while deciding Unavailable choices can be grayed out

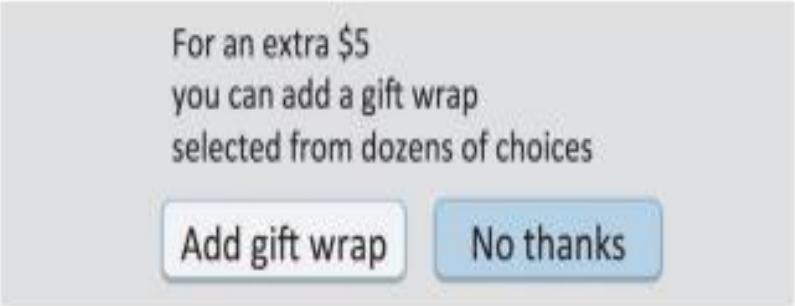


FIGURE 8.1

A simple menu with two choices. A short explanation is provided. Buttons are large enough to be easy to select and have informative labels, and one answer has been highlighted as the most likely answer.

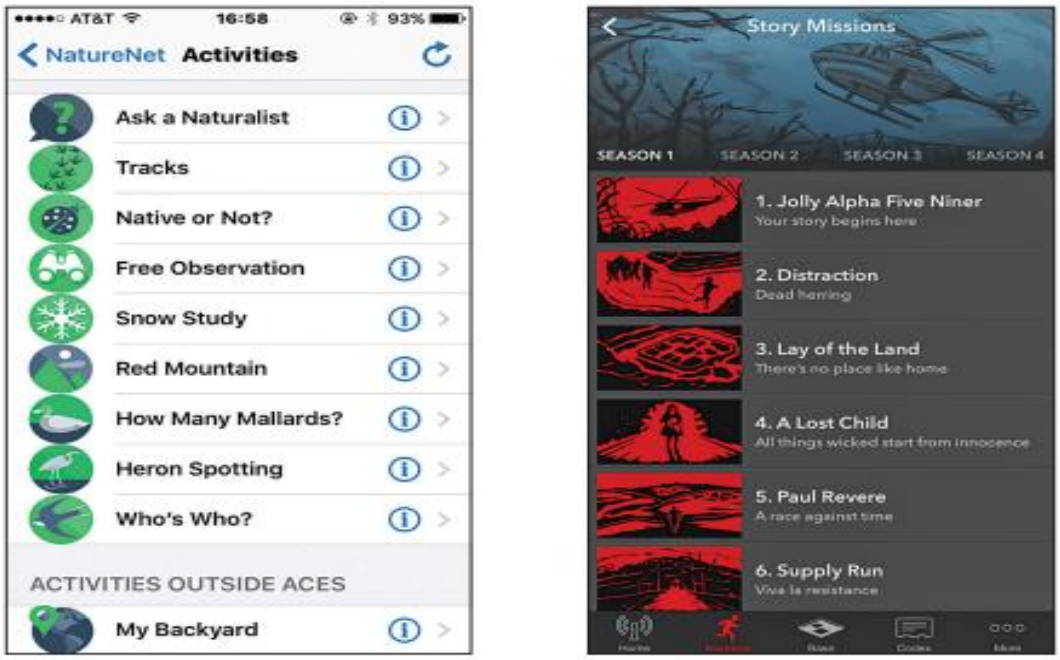


FIGURE 8.2

Two examples of simple menus. On the left, the NatureNet citizen science app shows the nine functions of the main menu. On the right, the Zoombies, Run! app lists the possible missions of Season 1 of the immersive running game and audio adventure.

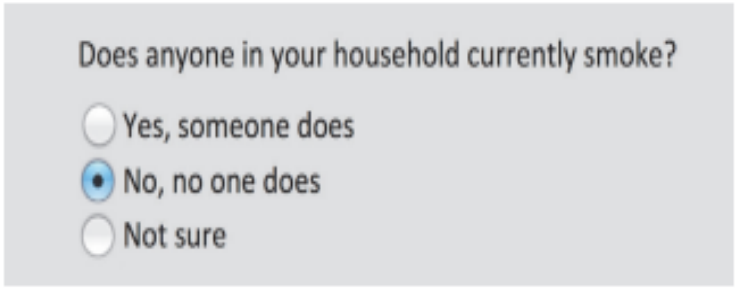


FIGURE 8.3

Three radio buttons constitute a menu that steers users to appropriate information in a health risk assessment website.

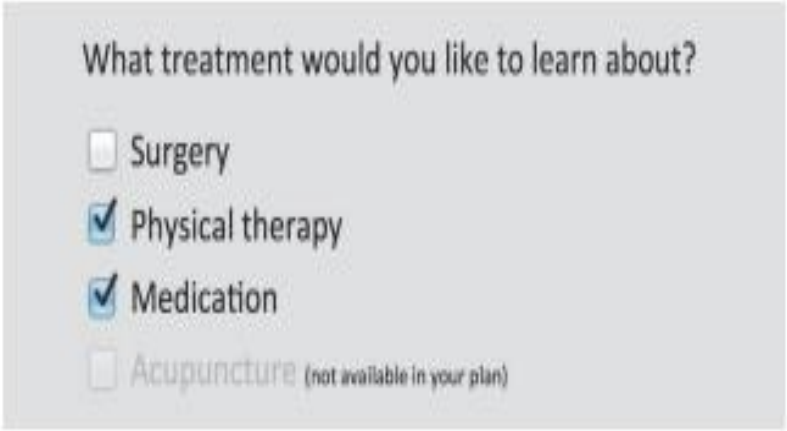


FIGURE 8.4

Check boxes allow users to indicate their preferences about treatment they would like to discuss. Feedback is provided by a check mark. Unavailable choices can be grayed out.

Menu bars, pop-up menus, toolbars, palettes, and ribbons

- Menu bars are typically found at the top of the each application or both at the top and on the side of the screen. Common items in desktop or tablet applications are File, Edit, View, and Help, and menus that follow this order will seem familiar to most users. Clicking on a menu title brings up a list

of related items, and users can then make a selection by moving the pointing device over the items (which respond by highlighting) and clicking on the desired choice. Since positional constancy is such a strong principle, when an item is not available for selection, it is important to gray it out rather than removing it from the list.

- The increasing ease of creating custom widgets allows designers to create endless variants of the original menu bars. Preserving readability and ensuring that users will be able to identify menus as such are important goals when creating these new designs. Many rely on multiple menu bars, placing menus at the top but also on the side and bottom of the screen or webpage.

- When placed on the side, submenus can open in place using an accordion menu style expansion, or to the side. Accordion menus work well when the submenus have few items and do not force users to scroll too far to collapse the accordion, but accordions may also increase user disorientation when the indenting scheme is unclear or the menu structure is more than two or three levels deep.

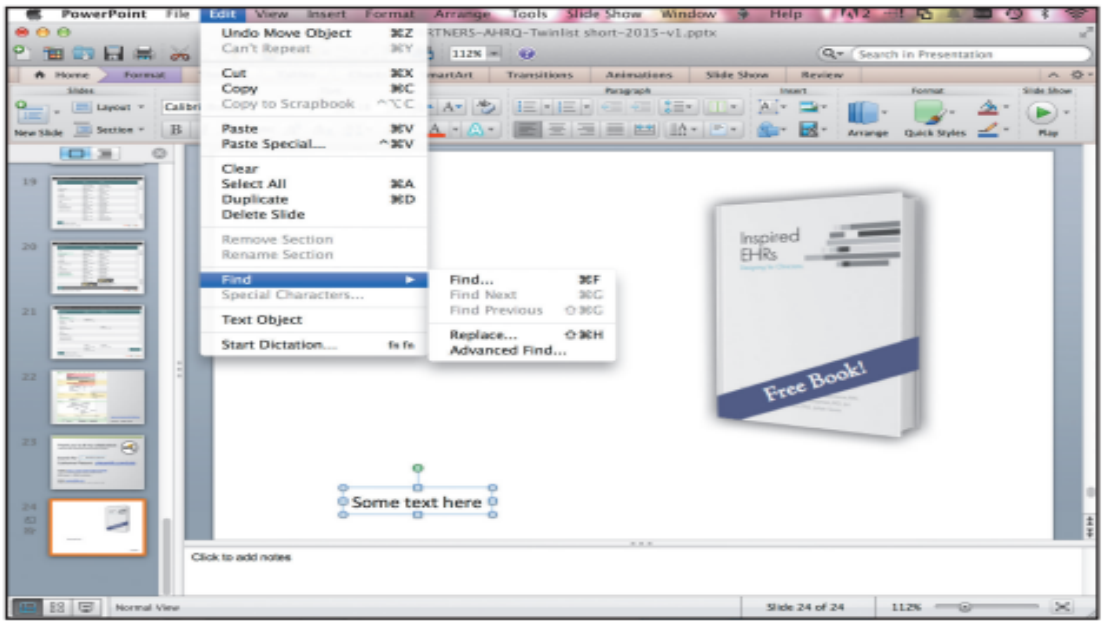


FIGURE 8.5
On the top menu bar of Microsoft PowerPoint, the Edit cascading *pull-down menu* (also called *pulled-right*) is open, followed by the Find menu. The menus allow users to explore the functions of the application. To facilitate discovery and learning, icons and keyboard shortcuts are indicated on the right of the menu items (for example, ⌘C for Copy or ⌘F for Find). A small black triangle indicates that selection of the menu item will lead to a submenu. Three dots (...) indicate that the selection will lead to a dialog box. Partially hidden behind the Edit menu, the application *ribbon* is visible, revealing the large number of choices available in the selected tab (Format).

- The limited screen space of mobile devices leads designers to strive to limit the number of menu items. To leave more room for content, most or all menu items can be moved into a separate screen that is accessible from a main menu icon, sometimes called the hamburger menu icon for its shape and which can be placed on every screen

- Toolbars, iconic menus, and palettes can offer many actions that users can select with a click and apply to a displayed object

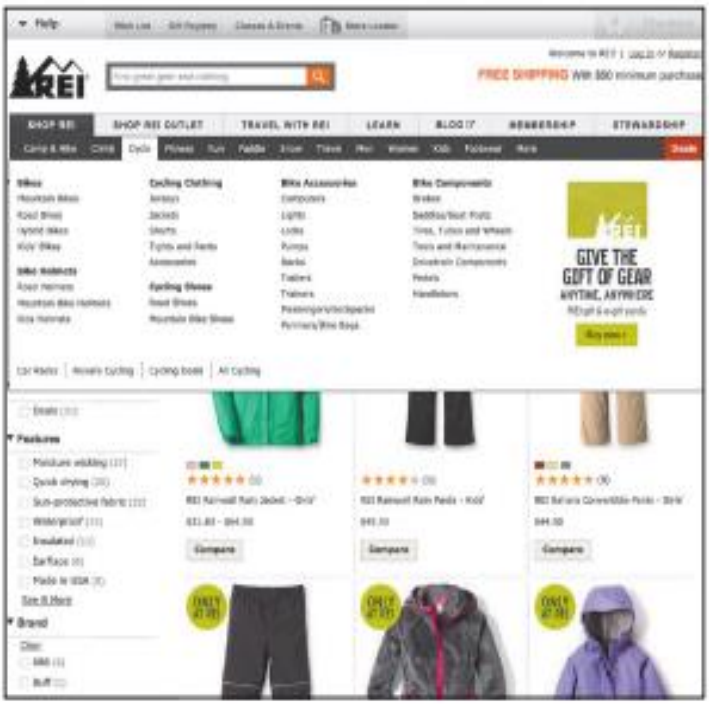


FIGURE 8.6

In the REI website, the categories for "Cycle" are expanded all at once below the top menu, showing 34 items organized in a meaningful hierarchy as a large menu.

Shortcuts and gestures for rapid interaction

- For rapid selection, keyboard shortcuts (also sometime called hotkeys, such as Ctrl-C on PCs or -C on Macs for Copy) are essential for expert users using desktop computers. Users can memorize the keystrokes for the menu items they use often and thus speed up the interaction considerably. The first letter of the command is often used for the shortcut to favor memorability, but caution is required to avoid collisions. If at all possible, shortcuts should be used consistently across applications; for example, Ctrl-S on a PC or -S on a Mac is usually used for Save and Ctrl-P or -P for Print. Keyboard shortcuts should be indicated next to their corresponding menu items and in the tooltip of the menu icons.

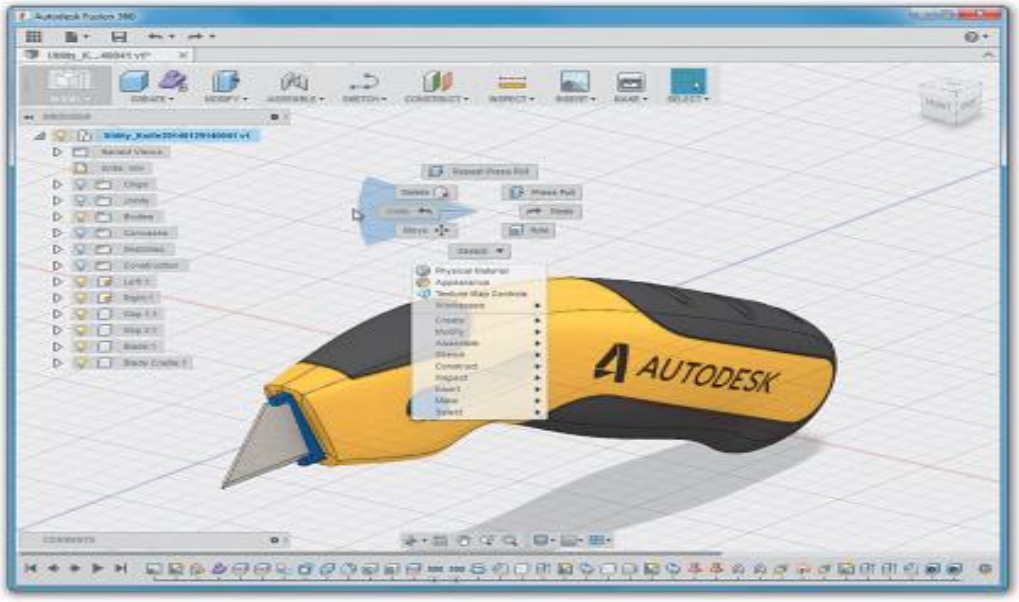


FIGURE 8.8
Fusion 360TM, an AutodeskTM 3D Computer Aided Design tool, allowed an engineer to design a utility knife. A click on the background of the image brings a pop-up marking menu with eight context-dependent menu items arranged in a circle (as well as a conventional linear menu below it). Sliding the mouse to the left selects the Undo command, now highlighted by a pie-shaped gray background. When the click + move is done rapidly, the menu itself doesn't appear on the screen, allowing rapid command selection via simple gestures (http://www.autodesk.com/products/fusion-360).

Examples of Common Gestures and Their Effects

Gestures can speed interaction, and their directness is compelling, but they are hard to discover. Gestures may have different actions when applied on an object, on the background, or toward the edge of the screen, which can be frustrating when applied inadvertently; therefore, it is important to ensure easy reversal of actions. Consistent application of gestures remains an issue.

- Tap: select
- Long press: varied, from magnified cursor (iOS) to showing a tooltip (Windows 8)
- Double tap: varied (e.g., zoom [iOS])
- Small swipe: varied (e.g., move location or order of objects, reveal a delete button)
- Large swipe: usually scroll
- Rapid swipe or fling: fast scroll with inertia
- Pinch and spread: zoom in and out

FastTap allows users to select commands by combining a thumb tap (to display the menu) and an index finger tap to select (Gutwin et al., 2014). As users learn the location of menu items relative to their thumb, they can select rapidly before the menu is even displayed. Allowing users to customize the gestures may help users remember them and provide better accessibility than pre-defined gestures, but users have limited understanding of the recognizer's ability to recognize gestures they propose, often leading to poorly recognized gestures

Long lists

Sometimes the list of menu items may be longer than the 30 to 40 lines that can reasonably fit on a display. One common solution is to create a tree-structured menu sometimes the desire to limit the interface to one conceptual menu is strong—for example, when users must select a state from the 50 states in the United States or a country from an extensive list of possibilities. Typical lists are alphabetically ordered, but categorical lists may be useful. The principles of menu-list sequencing apply.

- Scrolling menus, combo boxes, and fisheye menus
- Sliders and alphasliders
- Two-dimensional mega menus

Scrolling menus, combo boxes, and fisheye

Scrolling menus display the first portion of the menu and an additional menu item, typically an arrow that leads to the next set of items in the menu sequence. The scrolling (or paging) menu might continue with dozens or thousands of items. Allowing users to type the letter "M" to scroll directly to the first word starting with the letter "M" will reduce manual scrolling, but this feature is not always discovered. Similarly, typing M twice can move to the second word starting with "M". Combo boxes make this option more evident by combining a scrolling menu with a text-entry field. Users can type in leading characters to scroll quickly through the list. Another alternative is the fisheye menu, which displays all of the menu items on the screen at once but shows only items near the cursor at full size; items further away are displayed at a smaller size.

Sliders and alphasliders

When the available choices are continuous numerical values, a slider is a natural choice to allow the selection of a single value. Ranges of values can also be selected with double-sided (range) sliders. Users select values by using a finger or pointing device to drag the slider thumb (scroll box) along the scale (see Fig. 1.7). When greater precision is needed, the slider thumb can be adjusted incrementally by clicking on arrows located at each end of the slider. A similar technique that allows users to select a name or category among even large numbers of ordered items is an alphaslider.

FIGURE 8.9

An alphaslider (also called an item slider) in the Spotfire visualization tool from Tibco. The alphaslider allows users to select one item from a large number of categorical items and rapidly step through the other items (http://spotfire.tibco.com).

Two-dimensional mega menus

Alternatively, menus that fill all the available space might be used. Two-dimensional mega menus give users a good overview of the choices, reduce the number of required actions, and allow rapid selection. The ease of scrolling on touch screens has encouraged designers to make heavy use of scrollable two-dimensional menus in webpage design

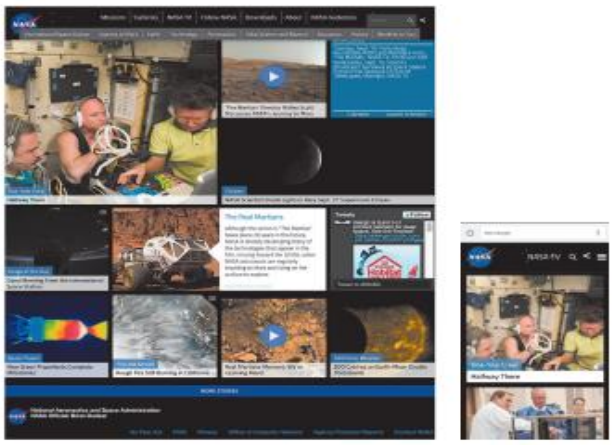


FIGURE 8.10

The NASA website consists of a large scrollable 2-dimensional menu. Below the main menu, each square or rectangle is a large button. Scrolling gives access to dozens of items easily updated and rearranged. This adaptive grid design scales down nicely to the small displays. On the right, the same page is displayed on an Android phone. The grid now appears as a single column of items.

Linear versus simultaneous presentation

Often, a sequence of interdependent menus can be used to guide users through a series of choices. For example, a pizza-ordering interface might include a linear sequence of menus in which users choose the size (small, medium, or large), thickness (thick, normal, or thin crust), and finally toppings. Other familiar examples are online examinations that have sequences of multiple-choice test items, each made up as a menu, or wizards (a Microsoft term) that steer users through software installation by presenting a sequence of menu options. Linear sequences guide users by presenting one decision at a time and are effective for novice users performing simple tasks. They may be the only possible option for a small display. Simultaneous menus present multiple active menus (also called filters) on a screen at the same time and allow users to enter choices in any order. They require more display space; however, experienced users performing complex tasks benefit from simultaneous menus. Faceted search menus are a very powerful application of simultaneous menus now used extensively in online shopping, library catalogs, and other database searches.

FIGURE 8.12

Awwwards.com gives awards to a large number of websites, which are tagged. A tag index at the top of the page displays all the tags sorted by total count. The counts are indicated in parenthesis. The green-colored tags are the popular tags that have been selected more often (which most likely will lead to even more selection).

Small Displays

While most designs adapt fairly easily from desktop displays to the larger tablets (once the design has been reviewed for touchability), small displays make most desktop designs impractical, and dumbed-down designs are very likely to fail. Small displays require a radical rethinking of what functionalities should be included and often lead to novel interface and menu designs specifically adapted to particular devices and applications. The smaller the screen, the more temporal the interface becomes (all the way to entirely linear audio interfaces when no display is available).

For example, linear sequences of menus are possible, while simultaneous menus are much harder to fit in. On tiny devices (such as watches or fitness wearables), a deck of card menu can be used, where each single tap advances to the next choice and a long press or two-finger press may select the item to access more information. Animated attention-catching ticker menus have also been used. Users don't need to manually scroll or page through the menu items, and with a single touch they can stop the scrolling and select an item in view.

Design considerations for small displays.

- Simplify: Less is more.

• Strive to reduce or eliminate data entry.

• Learnability is key.

• Consider use frequency and importance.

• Plan for interruptions.

• Use of contextual information.

• Make clear what is selectable and what is not.

• Leave room for scroll and swipe gestures to avoid inadvertent actions.

• Consider relegating less important functions to other platforms.

FIGURE 8.14
Small devices have very focused functionalities and few selectable areas. Discoverability is often an issue.

Here are some key considerations and strategies for working with small displays in HCI:

- ✓ Content Prioritization: Given the limited screen real estate, it's crucial to prioritize the most important information and actions. Designers need to make decisions about what content to display prominently and what can be hidden behind secondary layers or accessed through user interactions.

- ✓ Adaptive Layouts: Responsive and adaptive design techniques are essential for optimizing content layout on small displays. Content should be arranged in a way that makes the best use of available space, adapting to different orientations and screen sizes.

- ✓ Touch and Gestures: Small displays are often touch-enabled, so designing intuitive touch interactions is vital. Designers should consider the size of touch targets, gesture recognition, and avoid overcrowding the interface with small interactive elements.
  Minimizing Clutter: Cluttered interfaces can be overwhelming on small displays. Designers should aim for simplicity, minimizing unnecessary elements and using whitespace effectively to enhance readability and reduce visual noise.
- ✓ Navigation: Navigation patterns need to be carefully thought out. Traditional navigation bars might take up too much space, so alternative navigation solutions like tab bars, slide-out menus, or gesture-based navigation can be more suitable.
- ✓ Visual Hierarchy: Establishing a clear visual hierarchy helps users understand the importance of different elements on the screen. Proper use of typography, color, and contrast can guide users' attention effectively.
- ✓ Contextual Information: Consider providing contextual information that adapts to the user's current task or location. This can help reduce the need for excessive navigation and allow users to access relevant information more efficiently.
- ✓ Progressive Disclosure: Complex interactions can be broken down into simpler steps using progressive disclosure. This approach helps users focus on one task at a time and prevents overwhelming them with too much information at once.
- ✓ Consistency: Maintaining consistency in design elements, interactions, and visual language across different parts of the application ensures a smoother user experience and reduces confusion.
- ✓ Feedback and Affordances: Providing clear feedback for user actions helps users understand the outcome of their interactions. Visual cues, animations, and micro-interactions can enhance the perceived responsiveness of the interface.
- ✓ User Testing: Regular usability testing with representative users is crucial for identifying pain points and areas of improvement. Testing on the actual devices with small screens can uncover issues that might not be apparent in design mockups.
- ✓ Optimization for Performance: Small displays often come with limitations in processing power and graphics capabilities. Optimizing the performance of your application is essential to ensure smooth interactions and fast response times.

Content Organization

- Meaningful grouping and sequencing of choices, along with careful editing of titles and labels and appropriate layout design, can lead to easier-to-learn menus and increased navigation speed. In this section, This section reviews the content organization issues and provides guidelines for design. This area of design has been heavily researched in the context of traditional menus for desktop applications, but most results are useful for website and phone

application designs (Krug, 2014). Web pages act as large menus where items are the embedded links or buttons that can be used to navigate to another page

Structure and breath versus depth

- When a collection of items grows, designers can form categories of similar items, creating a tree structure. Some collections can be partitioned easily into mutually exclusive groups with distinctive identifiers. For example, the products in an online grocery store can be organized into categories such as produce, meat, dairy, cleaning products, and so on. Produce can then be organized into vegetables, fruits, and nuts, while dairy is organized into milk, cheese, yogurt, and so on.

Rules for forming menu trees.
Grouping menu items in a tree such that they are comprehensible to users and match the task structure is sometimes difficult. The problems are akin to putting kitchen utensils in order; steak knives go together and serving spoons go together, but where do you put butter knives or carving sets? Problems include overlapping categories, extraneous items, conflicting classifications in the same menu, unfamiliar jargon, and generic terms.

- Use task semantics to organize menus.
- Limit the number of levels (i.e., prefer broad–shallow to narrow–deep).
- Create groups of logically similar items: e.g., Level 1: countries, Level 2: states, Level 3: cities.
- Form groups that cover all possibilities: e.g., age ranges: [0–9] [10–19] [20–29] and [>= 30].
- Make sure that items are non-overlapping: e.g., use "Concerts" and "Sports" over "Entertainment" and "Events".
- Arrange items in each branch by natural sequence (not alphabetically) or group related items.
- Keep ordering of items fixed (or possibly duplicate frequent items in dedicated sections of the menu).

Tree-structured menu systems have the power to make large collections of data available to novice or intermittent users. If each menu has 10 items, a menu tree with four levels has the capacity to lead an untrained user through a collection of 10,000 destinations. That number would be excessively large for a word processor but is realistic in a newspaper, a library, or an enterprise web portal.

Terminology from the user's task domain can help orient the user: Instead of using a title that is vague and emphasizes the computer domain, such as "Main Menu Options", use terms such as "Friendlibank Services" or simply "Games". Menus using large indexes, such as library subject

headings or comprehensive business classifications, are challenging to navigate, making search a valuable alternative

The depth, or number of levels, of a menu tree depends in part on the breadth, or number of items per level. If more items are put into the main menu, the tree spreads out and has fewer levels.

Sequence, phrasing, and layout Sequence

Sequence: Once the items in a menu have been chosen, the designer is still confronted with the choice of presentation sequence. If the items have a natural sequence—such as days of the week, chapters in a book, or sizes of eggs—the decision is trivial. Many cases have no task-related ordering, though, so the designer must choose from either alphabetic order, grouping of related items, and most frequently used items first. Categorical organization is generally preferable over alphabetical. Using frequency of use does speed up selection of the topmost items, but the loss of a meaningful ordering for low-frequency items may be disruptive, so it is best limited to small lists

- A sensible compromise is to extract three or four of the most frequently selected items and put them near the top while preserving the order of the remaining items. This split-menu strategy proved appealing, statistically significantly improved performance, and has been adopted by commercial software. Adaptable menus (i.e., providing users with control over the sequence of menu items) is an attractive alternative to adaptive menus that adapt automatically. One study compared the Microsoft Word version using adaptive menus with a variant providing users with the ability to switch between two modes of operation: the normal full-featured mode and a personal mode that users could customize by selecting which items were included in the menus (McGrenere et al., 2007). Results showed that participants were better able to learn and navigate through the menus with the personally adaptable version.

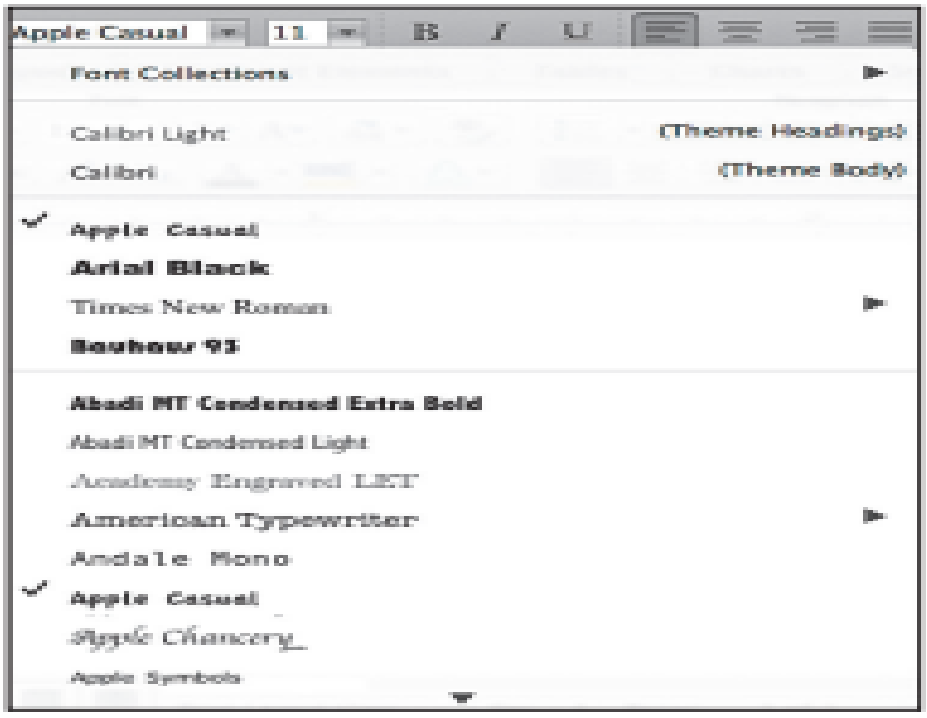


FIGURE 8.15
Example of adaptive split menus in Microsoft Office. A font-selection menu lists the theme fonts and then the recently used fonts near the top of the menu (as well as in the full list), making it easier to quickly select the popular fonts. A thin line separates the sections.

- **Phrasing:** For single menus, a simple descriptive title that identifies the situation is all that is necessary. For tree-structured menus, choosing titles is more difficult. One helpful rule is to use the words used for the menu items as the titles for the submenu or next pages. For example, it is reassuring to users to find that when they select “Business and financial services”, they are shown a display that is titled “Business and financial services”. It might be unsettling to get a display titled “Managing your money”, even though the intent is similar. For webpages, a distinctive short title displayed as browser tab label will help users return to the page after they visit other tabs. A distinctive icon improves the tab label as well.

The following directives may seem obvious but are listed here because they are so often violated:

• Use familiar and consistent terminology. Carefully select terminology that is familiar to the designated user community and keep a list of these terms to facilitate consistent use.

• Ensure that items are distinct from one another. Each item should be distinguished clearly from other items. For example, “Slow tours of the countryside”, “Journeys with visits to parks”, and “Leisurely voyages” are less distinctive than are “Bike tours”, “Train tours to national parks”, and “Cruise-ship tours”.

• Use consistent and concise phrasing. Review the collection of items to ensure consistency and conciseness. Users are likely to feel more comfortable and to be more successful with “Animal”, “Vegetable”, and “Mineral” than with “Information about animals”, “Vegetable choices you can make” and “Viewing mineral categories”.

• Bring the keyword to the fore. Try to write menu items such that the first word aids the user in recognizing and discriminating between items—use “Size of type” instead of “Set the type size”. Then, if the first word indicates that this item is not relevant, users can begin scanning the next item.

Layout: While the layout of applications and websites can be assisted by the use of templates and website management tools, designers who establish guidelines for consistency across dozens or hundreds of screens will reduce users’ anxiety by offering predictability (see Section 3.2). The following elements can be included:

• Titles. Some people prefer centered titles, but left justification is also acceptable.

• Item placement. Typically, items are left justified, with the item number or letter preceding the item description. Blank lines may be used to separate meaningful groups of items. If multiple columns are used, a consistent pattern of numbering or lettering should be used (for example, it is easier to scan down columns than across rows).

• Instructions. The instructions should be identical in each menu and should be placed in the same position. This rule includes instructions about traversals, help, or function-key usage.

• Error messages. If the users make unacceptable choices, the error messages should appear in a consistent position and should use consistent terminology and syntax. Graying out unacceptable choices will help reduce errors

Audio Menus

- Audio menus found in interactive voice response (IVR) systems (Lewis, 2010) are useful when hands and eyes are busy, such as when users are driving or testing equipment and are ubiquitous in phone surveys or services and public-access situations that need to accommodate blind or vision-impaired users, such as information kiosks or voting machines.

- With audio menus, instruction prompts and lists of options are spoken to users, who respond by using the keys of a keyboard or phone or by speaking. While visual menus have the distinct advantage of persistence, audio menus have to be memorized. Similarly, visual highlighting can confirm users' selections, while audio menus have to provide a confirmation step following the selection. As the list of options is read to them, users must compare each proposed option with their goal and place it on a scale from no match to perfect match.

- To reduce dependence on short-term memory, it is preferable to describe the item first and then give the number. A way to repeat the list of options and an exit mechanism must be provided (preferably by detecting user inaction). Complex and deep menu structures should be avoided. A simple guideline is to limit the number of choices to three or four to avoid memorization problems, but this rule should be re-evaluated in light of the application. For example, a theater information system will benefit from using a longer list of all the movie titles rather than breaking them into two smaller, arbitrarily grouped menus. Dial-ahead capabilities allow repeat users to skip through the prompts.

Here's a guide to designing audio menus in HCI:

- ✓ Clear Structure and Organization: Just like visual menus, audio menus should have a clear hierarchical structure. Group related options together and organize them logically.
- ✓ Use a consistent and predictable naming convention for menu items to help users understand and remember their options.
- ✓ Natural Language and Voice Commands: Allow users to interact with audio menus using natural language and voice commands. This can make the interaction more intuitive and user-friendly.
- ✓ Provide users with examples of voice commands they can use to navigate the menu.
- ✓ Limited Menu Depth: Avoid creating menus with too many levels of nesting. Deep hierarchies can be confusing and frustrating to navigate using audio alone.
- ✓ Auditory Cues for Navigation: Use distinct auditory cues (e.g., tones, chimes, spatial audio) to indicate transitions between menu levels and selections. Auditory cues help users understand the navigation flow.
- ✓ Contextual Feedback: Provide audio feedback to confirm user selections and actions. Users should hear confirmation of their chosen option or menu selection.
- ✓ Avoid Information Overload: Present information in manageable chunks to avoid overwhelming users with too much audio information at once.

- ✓ Progressive Disclosure: Similar to visual interfaces, consider using progressive disclosure in audio menus. Present high-level options first, and then offer more detailed options as users navigate deeper.
- ✓ Short and Clear Prompts: Use concise and clear prompts to guide users through the menu navigation process. Avoid lengthy or complex instructions.
- ✓ Hands-Free Interaction: One of the main advantages of audio menus is their hands-free nature. Ensure that users can easily navigate and make selections without requiring visual or physical interaction.
- ✓ Error Handling and Recovery: Design the audio menu to gracefully handle errors, such as misinterpretations of voice commands. Provide users with alternative options or allow them to repeat commands.
- ✓ Testing and Iteration: Conduct user testing with a diverse group of users, including those with varying levels of familiarity with voice interfaces and audio menus.
- ✓ Gather feedback and iteratively improve the audio menu design based on user insights.
- ✓ Personalization and User Profiles: Consider offering personalization options, such as allowing users to customize the menu structure or save frequently used options.
- ✓ Training and Onboarding: If your audio menu system includes advanced features or commands, offer an onboarding process or tutorial to guide users through the available functionality.
- ✓ Compatibility and Accessibility: Ensure that the audio menus are compatible with a range of devices and platforms, and that they adhere to accessibility guidelines for users with disabilities.

Form Fill-in and Dialog Boxes

Selection is effective for choosing an item from a set of choices, but if the entry of names or numeric values is required, typing becomes more attractive. When many fields of data are necessary, the appropriate interaction style is form fill-in. The combination of form fill-ins, menus, and custom widgets such as calendars or maps supports rapid navigation for a vast array of applications from airline-ticket booking to triage of new patients in the emergency room.

Form fill-in:

- There is a paucity of empirical research on form fill-in, but several design guidelines have emerged from practitioners (Jarrett and Gaffney, 2008). Software tools simplify design, help to ensure consistency, ease maintenance, and speed implementation, but even with excellent tools, the designer must still make many complex decisions. The elements of form fill-in design include the following:

- Meaningful title. Identify the topic and avoid computer terminology.

- Comprehensible instructions. Describe the user's tasks in familiar terminology. Be brief; if more information is needed, make a set of help screens available to the novice user. A useful rule is to use the word "type" for entering information and the word "press" for special keys such as the

Tab, Enter, or cursor movement (arrow) keys. Since “Enter” often refers to the special key with that name, avoid using it in the instructions (for example, do not use “Enter the address”; instead, stick with “Type the address”). Once a grammatical style for instructions is developed, be careful to apply that style consistently.

- Label the fields. Place the label in a consistent location (e.g., top or left of the field). A less desirable location is to place labels inside the fields, using a grayedout font. It saves space, but the labels disappear as soon as users start typing, requiring users to remember what is needed, which often leads to errors.

- Limit data entry. Make sure all fields are really needed. Carefully set default values (e.g., use the current location). This is particularly important for small displays .for example, using only the zip code instead of the city

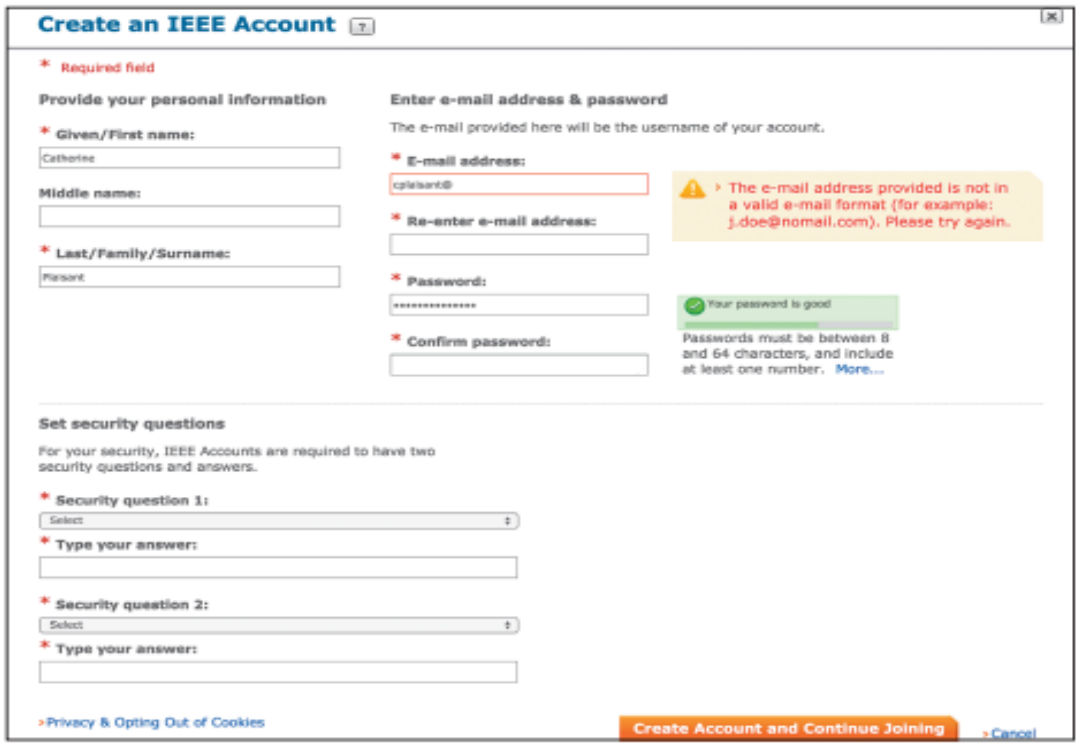


FIGURE 8.16

This form fill-in allows users to enter information when joining the IEEE Society. Fields are grouped meaningfully, and field-specific rules such as password requirements are provided next to the fields. The information is validated as it is provided (as opposed to when the form is submitted), and error messages explain how to correct problems (http://www.ieee.org).

- Explanatory messages for fields. Information about a field (e.g., “Your e-mail address will be the user name of your account”) or its permissible values should appear in a standard position, such as next to or below the field, preferably using a different font and style.

- Error prevention. Where possible, prevent users from entering incorrect values. For example, in a field requiring a whole number, do not allow the user to enter letters or decimal points.
- Error recovery. Summarize errors at the top of the page. Highlight errors in the form. If users enter unacceptable values, indicate permissible values for the field; for example, if the zip code is entered as 28K21 or 2380, the message might be “Zip codes should have 5 digits”.
- Immediate feedback. Immediate feedback about errors is preferable. When feedback can be provided only after the entire form has been submitted, the location of the field needing correction should be made clearly visible (for example, by displaying the error message in red next to the field in addition to general instructions at the top of the form).
- Logical grouping and sequencing of fields. Related fields should be adjacent and should be aligned with blank spaces for separation between groups. The sequencing should reflect common patterns—for example, city followed by state followed by zip code.
- Visually appealing layout of the form. Alignment creates a feeling of order and comprehensibility. For example, the field labels “Name”, “Address”, and “City” can be right-justified so that the data-entry fields are vertically aligned. This layout allows the frequent user to concentrate on the entry fields and to ignore the labels.
- Familiar field labels. Common terms should be used. If “Home Address” were replaced by “Domicile”, many users would be uncertain or anxious about what to enter.
- Consistent terminology and abbreviations. Prepare a list of terms and acceptable abbreviations and use the list diligently, making additions only after careful consideration. Instead of varying between such terms as “Address”, “Employee Address”, “ADDR.”, and “Addr.”, stick to one term, such as “Address”.
- Visible space and boundaries for data-entry fields. Users should be able to see the size of the field and to anticipate whether abbreviations or other trimming strategies will be needed. An appropriately sized box can show the maximum field length.
- Convenient cursor movement. Provide a mechanism for moving the cursor between fields using the keyboard, such as the Tab key or cursor-movement arrows.
- Required fields clearly marked. For fields that must be filled in, the word “Required” or some other indicator (e.g., *) should be visible. Optional fields should follow required fields whenever possible.
- Privacy and data sharing information. Users will be anxious sharing their personal information and want to know how the data will be used and who will have access to it.
- Accessibility. For example, make sure the forms are navigable with a screen reader.

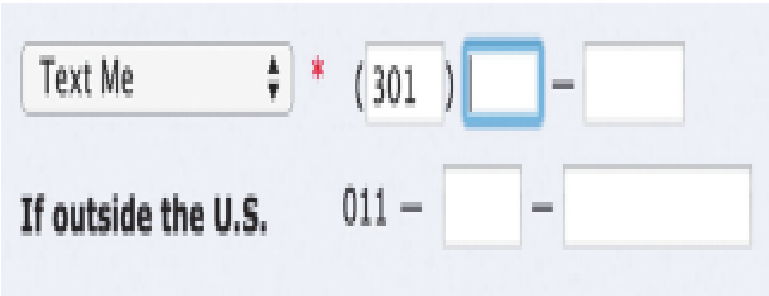
- Completion signal. It should be clear to the users what they must do when they are finished filling in the fields.

Format-specific fields

- Using custom widgets and direct-manipulation interaction techniques can facilitate data entry and reduce errors. Calendars can be used to enter dates, seating maps can help users select airplane seats, and menus using photographs might clarify choices of pizza style.

- Apps for touch screen devices need to open the keyboard with the appropriate preset; for example, when a number is requested, the numerical keyword should appear by default. For e-mail addresses, the "@" and "." buttons need to be shown. For URLs, the ":" and "/" will be handy. Alphabetic fields are customarily left-justified on entry and on display. Numeric fields may be left-justified on entry but then become right-justified on display. When possible, avoid entry and display of leftmost zeros in numeric fields (with zip codes being an exception). Numeric fields with decimal points should line up on the decimal points. Pay special attention to such common fields as these:

- *Telephone numbers.* Offer a form to indicate the subfields:



Be alert to special cases, such as the addition of extensions or the need for nonstandard formats for international numbers. When the user has typed all the needed digits of a field, the cursor should jump to the leftmost position of the next field.

- *Dates.* Providing a pop-up graphical calendar showing the current month will reduce the number of errors in some cases, but users may still want to type in the numbered field if moving the calendar to the correct date requires a large number of clicks (e.g., to enter a date of birth). Different formats for dates are appropriate for different tasks, and European rules differ from American rules. An acceptable standard may never emerge. Instructions need to show an example of correct entry. For example:

```
Date: _ _/_ _/_ _ _ _ (04/22/2016 indicates April 22, 2016)
```

For many people, examples such as this one are more comprehensible than abstract descriptions like *MM/DD/YYYY*.

- *Times.* Even though the 24-hour clock is convenient, many people in the United States find it confusing and prefer a.m. and p.m. designations.
- *Dollar amounts (or other currency).* The currency sign should appear on the screen so users enter only the amount. If a large number of whole-dollar amounts are to be entered, users might be presented with a field such as

```
Deposit amount: $_____.__
```

- Passwords: When asked to type a password, users also need a means to retrieve or change the password if they have forgotten it, but it is also important to avoid malicious use of that functionality
- CAPTCHAs:A CAPTCHA (acronym for Completely Automated Public Turing test to tell Computers and Humans Apart) requires users to type text presented graphically to be illegible to computers. Including an audio option is necessary to make the CAPTCHA accessible to users with visual impairments. Newer versions observe user behavior with the CAPTCHA to predict whether a human or a robot is interacting

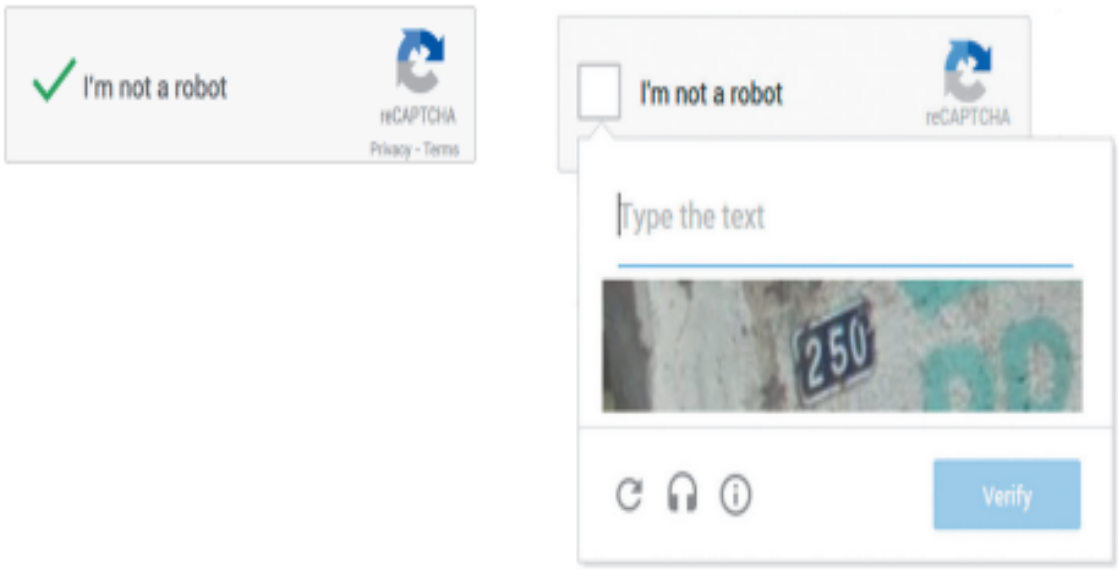


FIGURE 8.17

Google introduced a new reCAPTCHA in 2014. Observing the interaction, it predicts whether a human or a robot is clicking on the box but presents a more difficult CAPTCHA when in doubt. An audio version can play hard-to-understand words instead of the visual hard-to-read text.

Dialog boxes

- Many tasks are interrupted to request users to select options, perform limited data entry, or review alerts and error messages. The most common solution is to provide a dialog box . Dialog-box design combines menu-selection and form fill-in issues with additional concerns about consistency across potentially hundreds of dialog boxes and relationships with other items on the screen. A guidelines document for dialog boxes will help strive for consistency. Dialog boxes should have meaningful titles to identify them, and the titles should have consistent visual properties. Dialog boxes are often shaped and sized to fit each situation, but distinctive sizes or aspect ratios may be used to signal errors, confirmations, or components of the application.

Here are some important considerations when designing dialog boxes in HCI:

- ❖ Clarity and Conciseness: Dialog boxes should convey their purpose clearly and concisely. Use descriptive titles and messages that provide users with a clear understanding of what is happening and what action is required.
- ❖ Consistency: Maintain a consistent visual and interaction style for your dialog boxes to ensure that users can quickly recognize and understand them. Consistency in layout, color scheme, and typography helps create a seamless user experience.

- Alignment with User Workflow: Dialog boxes should align with the user's workflow and context. Avoid interrupting tasks unnecessarily and ensure that dialog boxes appear when users expect them.
- Appropriate Timing: Dialog boxes should appear at the appropriate time in the user's interaction. Avoid displaying them too frequently or too early, as this can disrupt the user's experience.
- Avoiding Overuse: Use dialog boxes sparingly. Overusing them can lead to user frustration and cognitive overload. Reserve them for situations that require immediate attention or user input.
- Clear Call to Action: Dialog boxes typically contain buttons for users to take specific actions (e.g., "OK," "Cancel," "Yes," "No"). Ensure that these buttons have clear labels that indicate the expected outcome of each action.
- Default Actions: When appropriate, set a default action for the most common response to a dialog box. This can help users proceed quickly without needing to make a selection.
- Modal vs. Modeless: Modal dialog boxes require users to interact with them before returning to the main interface, while modeless dialog boxes allow users to interact with both the dialog and the main interface concurrently. Choose the appropriate type based on the urgency and impact of the dialog's content.
- Feedback and Validation: Provide feedback when users interact with dialog box elements. For example, indicate that a required field has been left empty or that a certain input format is incorrect.
- Resize and Scalability: Ensure that dialog boxes are resizable and can accommodate varying amounts of content without sacrificing usability. This is especially important for accommodating translations and different screen sizes.
- Error Messages: Use dialog boxes to display error messages or alerts when something goes wrong. Clearly communicate the issue and provide suggestions for resolution if possible.
- Accessibility: Ensure that dialog boxes are accessible to users with disabilities. Use appropriate contrast, keyboard navigation, and screen reader compatibility.
- Testing: User testing is crucial for validating the effectiveness of your dialog boxes. Observe how users interact with the dialog boxes, gather feedback, and iterate on the design based on the findings.
- Exit and Dismissal: Provide a clear and easily accessible way for users to dismiss or close the dialog box. Users should always have a way to exit without taking an action if they choose.

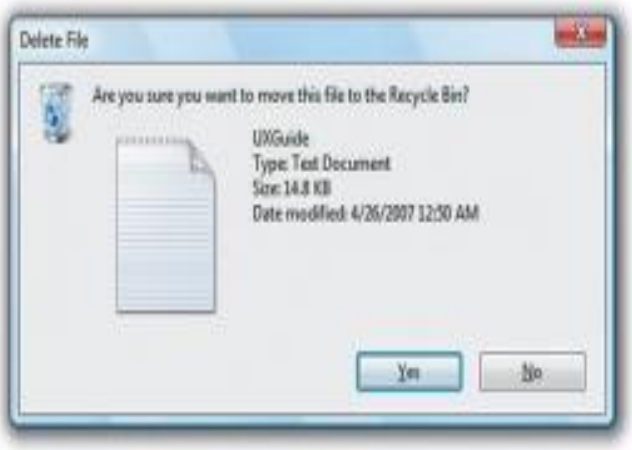


FIGURE 8.18

This dialog box includes a binary menu with two choices ("Yes" and "No"). The blue highlighting on Yes indicates that this selection is the default and that pressing Return will select it. Specific keyboard shortcuts can be made available. Escape closes the dialog box. Typing the letter "N" will select No, as indicated by the underlined letter "N".

FIGURE 8.19

This dialog box is used to alert clinicians who try to prescribe the drug Warfarin because it increases the risk of bleeding in patients already on aspirin. Several possible actions are proposed. Overriding the alert is possible but requires confirmation by clicking a check box. Because of the severity of the alert, this is a modal dialog box and requires immediate action.