

## Chapter 9. User-Computer Dialogues

In this chapter are guidelines for user-computer interactions in general and for the specific dialogue types selected for inclusion in the handbook.

### General Guidelines for User-Computer Interactions

In this section are general guidelines for interactions between the user and a computer.

#### 391. SYMBOLS AND PICTURES:

**Standardized graphic symbols and pictorial representations should be used in addition to text messages for communicating vehicle status information.**<sup>(6)</sup>

*Comment:* When display space is limited and abbreviations are used, their meaning may be less clear than the meaning of a good set of icons. However, given that iconic representations may not always be interpreted in the ways intended by their creator, well-designed experiments should be carried out on a sample representative of the user population along critical dimensions to ensure that any icons used are clearly understood by at least a significant majority of AHS users.

#### 392. MINIMIZING DRIVER WORKLOAD:

**When minimizing driver workload is important, avoid presenting irrelevant status information or imposing tasks that could be accomplished by the roadside or vehicle. Noncritical status messages should be inhibited during periods of high workload.**<sup>(6)</sup>

*Comment:* Presenting information that the driver does not need for an immediate task(s) but that might draw the driver's attention away from those tasks could have negative consequences. Similarly, functions that can be accomplished under automated control should not be allocated to the driver when the driver's workload is high due to other tasking.

#### 393. CHECK-IN PROCEDURES:

**If the driver will be required to configure the vehicle prior to, during, or after check-in, use computer-driven checklists.**<sup>(6)</sup>

*Comment:* Depending upon the extent of the driver's tasking, it may be better to not rely on the driver's memory. A computer-driven checklist, with appropriate highlighting of critical steps (see guideline 394), seems especially appropriate for first time AHS users.

**394. ELECTRONIC CHECKLISTS:**

If electronic checklists are used, emphasize critical steps, put critical items early in the list, and phrase items in terms of the desired status or value being considered (e.g., lights on).<sup>(6)</sup>

**395. INFORMATION ABOUT WHETHER THE SYSTEM OR THE DRIVER IS IN CONTROL:**

Ensure that a driver always has explicit information about whether the driver is in control or the automation is in control. Redundant coding and/or multiple displays should be used for this purpose. When there is a phased transfer of control on a function by function basis (e.g., steering followed by speed control), the automation status of each independent function being transferred should be explicitly annunciated.<sup>(6)</sup>

*Comment:* This is critical so that the driver does not rely on the automation when in fact the vehicle is under manual control.

**396. MANUAL STEERING IN AN EMERGENCY:**

When there is a need for manual steering under emergency conditions, use speech commands that tell the driver the appropriate steering maneuver (e.g., "Turn left"). Visual cues may be used to supplement the speech commands.<sup>(6)</sup>

*Comment:* It is assumed that the automation "knows" where all potential obstacles are before issuing such directions. If that is not the case, directional guidance should not be given.

**397. WHAT INFORMATION TO PRESENT:**

All information essential to decisionmaking should be presented to the user, and only such information.<sup>(31)</sup>

**398. STRUCTURE OF SCREEN FORMATS:**

Screen formats should be designed with a consistent structure that is evident to the user, so that any particular type of information is always presented in the same place and in the same way.<sup>(42)</sup>

**399. PRESENTING INFORMATION TO GUIDE THE USER THROUGH A TRANSACTION :**

Screen formats should be designed so that information intended to guide the user through a transaction is readily distinguishable from other information.<sup>(42)</sup>

**400. SCREEN TITLE:**

Each screen should have a unique title in a highly visible location that is consistent across screens.<sup>(32)</sup>

**401. CHARACTER CASE:**

Text should be in mixed case characters. Exception: Captions, labels, titles, etc., that need to stand out on a screen may be in all upper case characters.<sup>(31)</sup>

**402. CURSOR DESIGN:**

Cursors should be designed to be readily distinguishable from other displayed symbols and information.<sup>(42)</sup>

**403. CURSOR VISIBILITY:**

A cursor should be visible to the user at all times.<sup>(32)</sup>

**404. CURSOR IMAGE QUALITY:**

The cursor should maintain good image quality throughout its entire range of motion within a screen. The position of the cursor should be clearly visible while it is moving. And, the cursor should obscure other characters, not vice versa. Exception: A blinking cursor (e.g., a blinking underline) need not obscure other characters.<sup>(32)</sup>

**405. MESSAGE VOICE:**

Active rather than passive voice should be used in messages (e.g., say “To clear the screen, press CLEAR” rather than “The screen is cleared by pressing CLEAR”).<sup>(42)</sup>

**406. WHEN TO USE ABBREVIATIONS AND ACRONYMS:**

An abbreviation and/or acronym should be used only if a screen does not have sufficient space for the unabbreviated word or if the abbreviation or acronym is used more frequently than the complete word (e.g., AHS instead of automated highway system).<sup>(32)</sup>

**407. HOW TO FORM ABBREVIATIONS:**

If a word must be abbreviated and there is no conventional abbreviation for it, use a single, simple rule (e.g., truncation) to form the abbreviation (e.g., destination becomes abbreviated to dest.). The same rule should be applied throughout to form abbreviations.<sup>(32)</sup>

**408. UNDERSTANDABILITY OF ABBREVIATIONS AND ACRONYMS:**

An abbreviation and/or acronym should be clearly understood by the user population.<sup>(32)</sup>

**409. CODES AND ABBREVIATIONS, AND CONVENTION:**

Codes and abbreviations should conform to conventional usage and use expectations. (Based on reference 42.)

**410. AVAILABILITY OF MEANINGS OF ABBREVIATIONS AND ACRONYMS:**

When abbreviations and/or acronyms are used, they and the complete words they represent should be easily accessible to the user (e.g., in an on-line glossary).<sup>(32)</sup>

**411. WORDING OF MESSAGES:**

When wording labels, prompts, and other messages, terminology familiar to users should be used.<sup>(42)</sup>

**412. WORDING IN PROMPTS AND CONTROL ENTRIES:**

The wording in prompts should be consistent with the wording in control entries (e.g., say “To cancel a transaction, press CANCEL” rather than “To abort a transaction, press CANCEL”).<sup>(42)</sup>

**413. AFFIRMATIVE WORDING:**

Prompts and other messages should be stated in the affirmative (e.g., say “You must enter a destination before entering the check-in area” rather than “You cannot enter the check-in area until you enter a destination”).<sup>(31)</sup>

**414. ORDER OF INFORMATION IN PROMPTS:**

Information in a prompt should be presented in chronological order (e.g., say “Press CHG LANE and then indicate direction” rather than “Indicate the direction of the turn after pressing CHG LANE”). Information in a prompt should be presented in the order in which the user will need it (e.g., say “To change lanes, press CHG LANE” rather than “Press CHG LANE to change lanes”).<sup>(31)</sup>

**415. FORMAT OF PROMPTS:**

White space, justification, and visual cues should be used to format prompts. For example:

**Poor:** To go back one screen, press BACK; to go to the main menu, press MAIN.

**Improved:** To go back one screen, press BACK.  
To go to the main menu, press MAIN.<sup>(31)</sup>

**416. TERMINOLOGY:**

Use consistent terminology in labels, prompts, etc. (Based on reference 31.)

**417. DESTRUCTIVE COMMANDS:**

The user should be required to take some explicit action to confirm a potentially destructive command (e.g., “Delete . . . .”) before the computer will execute it.<sup>(42)</sup>

**418. UNDOING THE LAST ACTION:**

Provide a simple means for the user to undo his/her last action (e.g., an UNDO key).<sup>(31)</sup>

**419. HELP FUNCTION:**

The user should be provided with a context-sensitive help function. A single, standard action that *is* always available should be provided to request the help.<sup>(13,31)</sup>

**420. BROWSING THROUGH HELP:**

The user should be allowed to browse through on-line help screens to gain familiarity with system functions and operating procedures<sup>(13)</sup>

**421. CANCELING AN OPERATION:**

Provide a means for the user to cancel an operation that is in progress.<sup>(31)</sup>

**422. SCROLLING AND PAGING:**

A user should be able to move through a set of linked screens by either scrolling or paging:

- a. Scrolling involves continuous movement (vertical or horizontal) within a set of linked screens. Users perceive two types of scrolling: moving text, in which the information on the screen appears to move behind a fixed window; and panning, in which a window appears to move in front of the information on the screen.
- b. Paging involves discrete movement within a set of linked screens; the unit of movement is typically one page.<sup>(32)</sup>

**423. PREFERRED SCROLLING METHOD:**

When scrolling is used, only one method should be implemented. Panning is preferred to moving text.<sup>(32)</sup>

**424. SCROLL INCREMENT:**

The scroll increment should be one line vertically or one character horizontally. The rate of scrolling should allow the user to scan the information as it moves past on the screen.<sup>(32)</sup>

**425. DIRECTION OF SCROLLING:**

The direction that information will be scrolling should be evident to the user before the user begins to scroll.<sup>(32)</sup>

**426. DIRECTION OF PAGING:**

The direction that information will be paging should be evident to the user before the user begins to page.<sup>(32)</sup>

**427. MOVEMENT OF INFORMATION WHEN PAGING:**

When paging, movement of information should be discrete, with no display of intermediate pages between the starting page and the selected page.<sup>(32)</sup>

**428. SYSTEM-DELAY MESSAGES:**

When system functioning requires the user to stand by, a message such as “Working” should be displayed until user interaction is again possible. When the delay is likely to exceed 15 s, a message should inform the user about the cause of the delay and approximately how long it will last. For delays exceeding 60 s, a count-down display should show delay time remaining.<sup>(13,64)</sup>

**429. INTEGER NUMBERS:**

Integer numbers should be right justified.<sup>(31)</sup> For example:

Poor:           10  
                  100  
                  1,000

Improved: 10  
                  100  
                  1,000

#### 430. REAL NUMBERS:

Real numbers should be decimal aligned.<sup>(31)</sup> For example:

Poor:           24.8  
                  123.56  
                  1.49

Improved:      24.8  
                  123.56  
                  1.49

#### 431. LEADING ZEROS:

Avoid leading zeros when they are not necessary and/or not standard (e.g., the leading zeros are standard in 24-h clock times, such as 0130).<sup>(31)</sup>

### Function Keys Dialogue

#### DEFINITION:

There are two types of function keys, as follows:

- a. Dedicated, which are hard keys that perform only a single function. The name of the function (e.g., ENTER) is on the key top.
- b. Nondedicated, which are hard keys that perform different functions depending upon the mode the system is in or whether some other key is pressed in combination with the function key. As an example of the latter, a typical alphanumeric keyboard has a key that types 5 when pressed alone and % when pressed in combination with the SHIFT key. The  $\left[ \begin{array}{c} \% \\ 5 \end{array} \right]$  key is a nondedicated function key (and the SHIFT key is a dedicated function key).

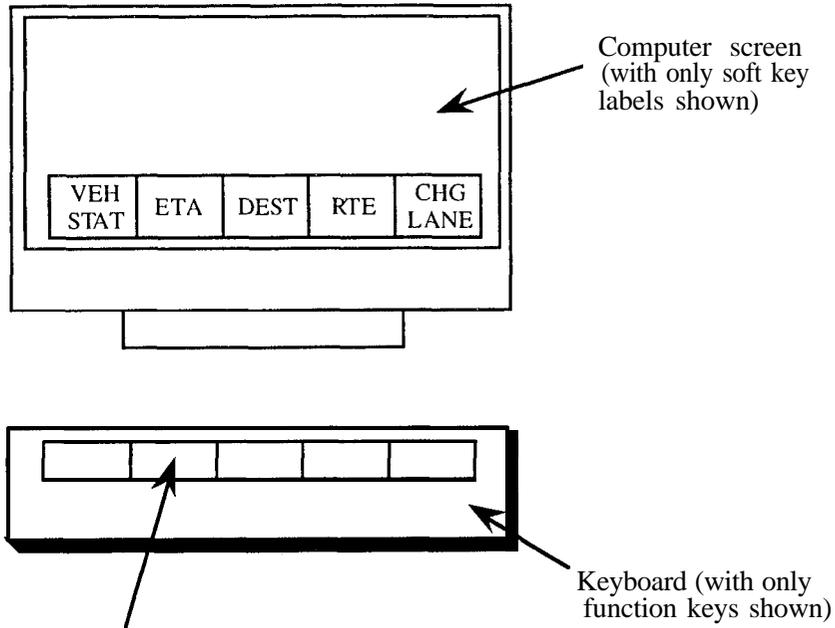
Nondedicated function keys have either no labels or generic labels (e.g., F1, F2). Their current function may be displayed as a “soft” key label on an associated computer screen (see figure 29). (Based on reference 31.)

#### 432. WHEN TO USE:

Consider function keys for tasks requiring only a limited number of control entries, or for use in conjunction with other dialogue types as a ready means of accomplishing critical entries that must be made quickly without syntax error.<sup>(42)</sup>

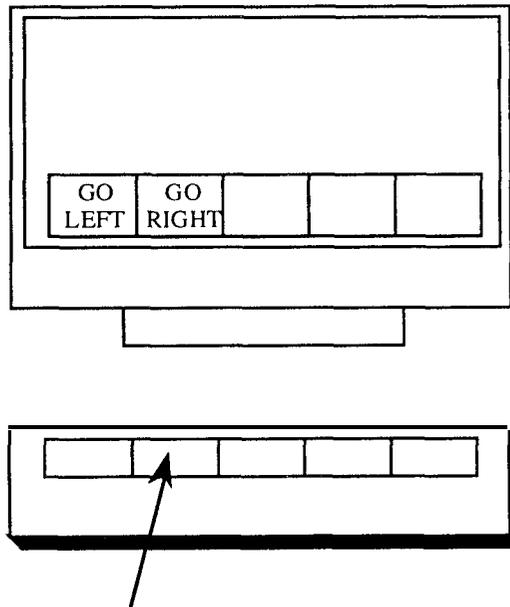
#### 433. IF A KEY IS USED FOR MORE THAN ONE FUNCTION:

If a key is used for more than one function, the function that is currently available should always be indicated to the user.<sup>(42)</sup>



Pressing this key will allow the user to determine the estimated time of arrival at the destination.

(a) Soft labels for nondedicated function keys.



Pressing this key will indicate to the system that the user wants to move into the lane to his/her right.

(b) The same nondedicated function keys as in (a), but with different soft key labels (indicating their new functions).

**Figure 29. Soft labels for nondedicated function keys.**

**434. WHEN FUNCTION KEY ACTUATION DOES NOT PROVIDE OBVIOUS FEEDBACK:**

When function key actuation does not result in any immediately observable response, provide the user with some other form of computer acknowledgment.<sup>(42)</sup>

**435. INDICATING ACTIVE KEYS:**

If some function keys are active and some are not, indicate the current subset of active keys in some noticeable way.<sup>(42)</sup>

**436. UNNEEDED KEYS:**

When a function key is not needed for any current transaction, temporarily disable it under computer control.<sup>(42)</sup>

**437. CONTINUOUSLY AVAILABLE FUNCTIONS:**

When a function is continuously available, assign that function to a dedicated function key.<sup>(42)</sup>

**438. CONSISTENCY IN KEY ASSIGNMENTS:**

If a function is assigned to a particular key in one transaction, assign that function to the same key in other transactions.<sup>(42)</sup>

**439. USE OF QUALIFIER KEYS:**

The use of qualifier keys (e.g., SHIFT, CONTROL) should be minimized. When they are used, they should be used consistently throughout the system.<sup>(31)</sup>

**440. RELATIONSHIP BETWEEN SOFT KEY LABELS AND HARD FUNCTION KEYS:**

Spatial relationships between soft function key labels (on the screen) and their associated hard function keys should be preserved; a horizontal layout is preferred (i.e., both the labels and the hard keys should be laid out from left to right).<sup>(31)</sup>

**441. GENERIC LABELING OF HARD KEYS ASSOCIATED WITH SOFT KEY LABELS:**

When the hard function keys associated with soft function key labels have generic labels on them (e.g., F1, F2):

- a. The labels on the hard keys should also appear as part of the soft function key labels (see figure 30).
- b. The order of the labels on the screen should correspond to the order of the hard keys.<sup>(31)</sup>

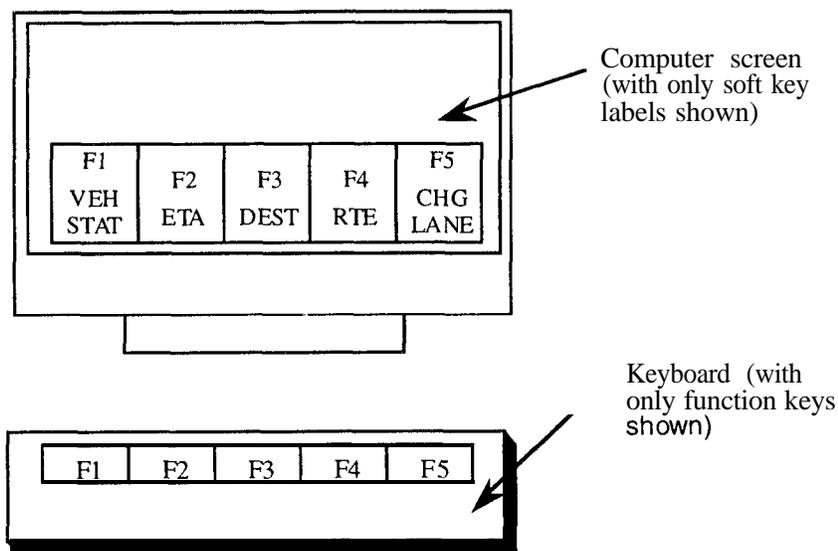


Figure 30. Correct spatial relationship between soft function key labels and associated hard function keys.

**442. NUMBER AND USE OF FUNCTION KEYS:**

There should be a limited number of function keys, and they should be used to perform highly frequent actions.<sup>(32)</sup>

**443. FUNCTION KEY LABELS:**

Each function key should be labeled clearly and uniquely to describe its function.<sup>(32)</sup>

**444. HARD KEYS AND USE OF GENERAL LABELING GUIDELINES:**

Labels on hard function keys should conform to the applicable guidelines in chapter 8 of this handbook.

**445. SOFT KEY LABELS AND USE OF DISPLAYS GUIDELINES:**

Soft function key labels should conform to the applicable guidelines in chapters 5 and 6 of this handbook.

**446. RESULTS OF PRESSING A FUNCTION KEY:**

Pressing a function key should result in a single action that does not change with repeated key presses.<sup>(32)</sup>

**447. PRESSING A FUNCTION KEY IN A SEQUENCE OF UNRELATED KEY PRESSES:**

Pressing a function key in a sequence of key presses unrelated to the function should result in a message asking the user whether he/she really intended to select that function; it should not result in the action normally produced by the function key unless the user responds positively to the question.<sup>(32)</sup>

## **Menu Selection Dialogue**

### **DEFINITIONS:**

Menu selection provides a user initiated transaction sequence that permits the user to select a control option from a display of several choices. There are three main types of menus:

- a. A static menu is one that is displayed continuously until the user makes a selection.
- b. A pop-up menu is not displayed until the user selects it, at which point it appears to pop up from the bottom of the screen.
- c. A pull-down menu is not displayed until the user selects it, at which point it appears to pull down from the top of the screen.<sup>(32)</sup>

Only static menus are discussed in this handbook.

**448. WHEN TO USE:**

Consider menu selection for tasks that involve a choice among a constrained set of alternative actions, that require little entry of arbitrary data, where users may have little training, and where computer response is relatively fast.<sup>(42)</sup>

**449. WHEN MENU SELECTION IS THE PRIMARY MEANS OF SEQUENCE CONTROL:**

When menu selection is the primary means of sequence control, and especially if choices must be made from extensive lists of displayed control options, option selection by direct pointing (e.g., using a touch screen) should be permitted.<sup>(42)</sup>

**450. MENU TITLE:**

A menu should have a brief title indicating the nature of the choice to be made.<sup>(32,42)</sup>

**451. MENU LIST FORMAT:**

When multiple menu choices are displayed in a list, each choice should be displayed on a new line, i.e., format the list as a single column. **Exception 1:** Displaying choices in several columns may be considered where shortage of display space dictates a compact format; if there are only a few choices, they might be displayed in a single row.

**Exception 2:** An exception could be made for hierarchic menus, where a high level menu might be shown in the left column of a display, accompanied by a lower level menu in the right column, the choices of which change to reflect whatever selection is currently made from the high level menu.<sup>(42)</sup>

**452. ACTUATION WHEN SELECTION IS BY POINTING:**

If menu selection is accomplished by pointing (e.g., with a touch screen), provide for dual actuation: The first action designates (positions a cursor at) the desired choice, the second action makes an explicit control entry.<sup>(42)</sup>

**453. COMMAND AREA WHEN SELECTION IS BY CODE ENTRY:**

When menu selection is accomplished by code entry, provide a standard command entry area (window) where the user enters the selected code. Locate that entry area in a fixed location on all screens.<sup>(42)</sup> (See figure 31.)

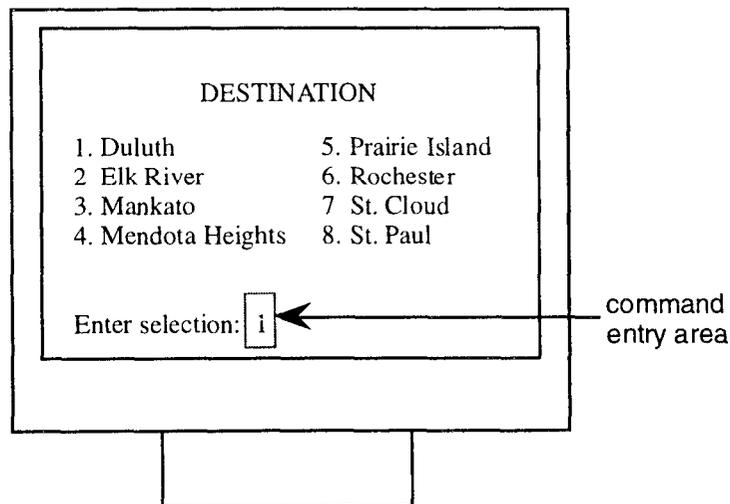


Figure 31. Use of a standard command entry area for selection codes.

**454. CODE PRESENTATION WHEN SELECTION IS BY CODE ENTRY:**

When menu selection is accomplished by code entry, the code associated with each choice should be shown in a consistent, distinctive manner (e.g., use an equal sign, as in “n = next page”).<sup>(42)</sup>

**455. NULL ENTRY WHEN CONTROL IS BY KEYED COMMAND OR OPTION CODE:**

When control is accomplished by keyed command or option code entry, if there is a default for a null control entry, that default should be indicated to the user (e.g., “To see more choices, press ENTER”).<sup>(42)</sup>

**456. WORDING OF CHOICES:**

The wording of menu choices should consistently represent commands to the computer rather than questions to the user (e.g., say “Print” rather than “Print?”).<sup>(42)</sup>

**457. WHEN SELECTION IS BY LETTER CODE:**

If letter codes are used for menu selection, those letters should be used consistently in designating choices from one transaction to another.<sup>(42)</sup>

**458. WHEN SELECTION IS FROM A LONG LIST:**

When menu selection must be made from a long list, and not all choices can be displayed at once, provide a hierarchic sequence of menu choices rather than one long multipage menu. Exception: Where a long list is already structured for other purposes, such as a list of destinations, it might be reasonable to require the user to scan multiple display pages to find a particular choice. Even in such cases, however, an imposed structure for sequential access may prove more efficient, as when a user can enter preliminary letters to access a long alphabetic list (e.g., typing “wa” to go directly to Washington).<sup>(42)</sup>

**459. CURSOR PLACEMENT WHEN SELECTION IS BY POINTING:**

When menu selection is by pointing, the computer should place the cursor automatically at the first listed choice. When menu selection is by code entry, place the cursor in the command entry area (see figure 31).<sup>(42)</sup>

**460. INDICATION OF CURRENT POSITION IN A HIERARCHIC MENU:**

When hierarchic menus are used, some indication of current position in the menu structure should be displayed to the user. A menu tree showing the hierarchy should be included in the user’s manual.<sup>(13,42)</sup>

**461. RETURNING TO THE NEXT HIGHER LEVEL IN A HIERARCHY:**

When hierarchic menus are used, only one, simple control action should be required to return to the next-higher-level menu.<sup>(42)</sup>

**462. FEEDBACK ABOUT CHOICE SELECTION:**

Feedback that a menu choice has been selected should be provided by making that item perceptually distinct.<sup>(32)</sup>

**463. FEEDBACK ABOUT COMMANDED-ACTION COMPLETION:**

If the completion of an action commanded by selection of a menu choice has a result that is visible to the user, that result is sufficient feedback to the user. If completion of the action has no visible result, feedback should be provided by a message in a message area. After the completion of all actions commanded by a menu item, the menu should be removed from the screen.<sup>(32)</sup>

**464. HIERARCHY DEPTH VS. BREADTH:**

Menu hierarchy depth should be minimized at the expense of breadth (see table 33).<sup>(31)</sup>

**Table 33. Maximum menu hierarchy breadth.**

User/Task Variables	Maximum Optimal Menu Breadth <sup>1</sup>
Choice items are complex and/or choice items cannot be grouped.	≤10 choice items per screen
Choice items are not complex and choice items can be grouped, but users are infrequent/casual.	11 to 20 choice items per screen
Choice items are not complex and choice items can be grouped, and users are frequent/expert.	≥21 choice items per screen

<sup>1</sup> Within the ranges given, actual breadth and depth should be determined by the most natural categorization of items into a hierarchy.

**465. INACTIVE CHOICES:**

Inactive menu choices should be “grayed out” (i.e., shown in lightened characters) and unselectable.<sup>(31)</sup> (See figure 32.)

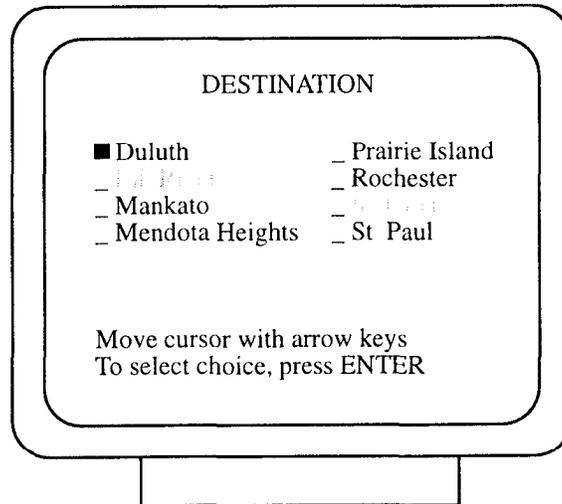


Figure 32. Inactive menu choices are “grayed out.”

#### 466. GENERAL RULES FOR CHOICES:

Menu choices should be brief, consistent in grammatical style (e.g., all start with verbs) and placement (e.g., make **HELP** always the last item on a menu), and matched with corresponding menu titles (e.g., if the menu item selected is “Destinations,” the screen to which it leads should be titled “**DESTINATIONS**” as opposed to “**AVAILABLE CITIES**”).<sup>(31)</sup>

#### 467. ORDERING OF CHOICES:

Menu choice labels should be ordered according to the following methods, listed with the preferred first:

- a. Convention.
- b. Frequency of use.
- c. Order of use.
- d. Categorical or functional groups.
- e. Alphabetical order.

The specific method chosen will depend on the user and on task variables.<sup>(31)</sup>

#### 468. WHEN NUMBERS ARE USED WITH CHOICES:

If numbers are used to code menu choices, they should start with 1.<sup>(31)</sup>

#### 469. WHEN CHARACTER CODES ARE USED WITH CHOICES:

If character codes are used to code menu items (e.g., “d” for destination),

the menu items should be left justified in a column and their associated codes should be aligned in a column to the left of the menu items.<sup>(31)</sup>

**470. WHEN A CURSOR IS USED TO SELECT A CHOICE:**

When a cursor is used to select a menu choice, a separate control action should be required to accept the selection. (Based on reference 42.) (See figure 33.)

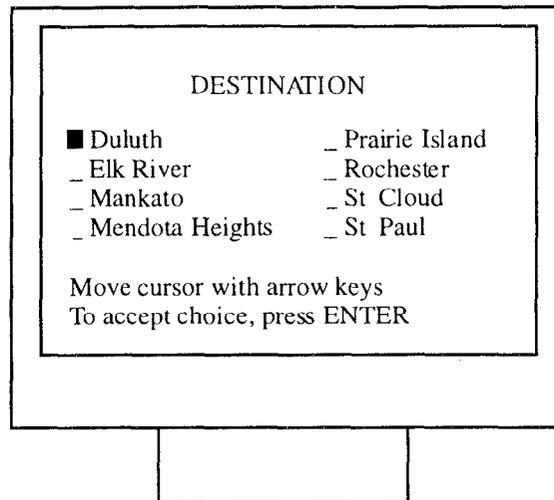


Figure 33. Use of a separate control action to accept a choice selected by a cursor.

**471. MENU SELECTION SHORT CUTS:**

The user should be able to go directly to a specific menu by typing ahead (entering all choices from all successive menus in a particular pathway all at once, without viewing each screen) or by entering the name of (or some appropriate mnemonic for) the menu to which the user wishes to go (e.g., type “select city” to go directly to that menu). When typing ahead, if some punctuation other than spacing is needed to separate the entries, a single, standard symbol should be used throughout (e.g., *l*).<sup>(31,42)</sup>

*Comment:* If the user is allowed to directly enter the menu name (or some short form thereof), it might be a good idea to have an intelligent system that can deal with spelling errors by either accepting “reasonable” alternatives to the correct spelling and/or by learning the user’s idiosyncracies and accepting those on subsequent attempts.

## Question and Answer Dialogue

### **DEFINITION:**

A question and answer dialogue involves a computer-initiated sequence of transactions between the user and the system that provides explicit prompting in performing task and control activities.<sup>(32)</sup>

### **472. WHEN TO USE:**

Consider a question and answer dialogue for routine data entry tasks, where data items are known and their ordering can be constrained, where users will have little or no training, and where computer response is expected to be moderately fast.<sup>(42)</sup>

### **473. NUMBER OF QUESTIONS TO BE DISPLAYED SIMULTANEOUSLY:**

In a question and answer dialogue, display each question separately; do not require the user to answer several questions at once.<sup>(42)</sup>

### **474. WHEN A SERIES OF QUESTIONS IS INTERRELATED:**

When a series of computer posed questions is interrelated, display answers to previous questions when those will provide context to help a user answer the current question.<sup>(42)</sup>

### **475. SPACE FOR ANSWERING THE QUESTION, AND CURSOR LOCATION:**

Space for answering the question should be provided closely following the question mark. Exception: When additional information needed for the answer is provided following the question, the answer space should follow that additional information. In either case, the cursor should be placed at the start of the answer area.<sup>(32)</sup>

### **476. CONTEXTUAL INFORMATION FOR ANSWERING A QUESTION:**

The system should provide the user with contextual information required for answering the question (e.g., if the answer must be a gap, the question should be followed by "ft"). The answer area should follow the contextual information.<sup>(32)</sup>

### **477. FEEDBACK FOR COMMANDS:**

When a question and answer dialogue is used to communicate commands and the completion of the action commanded has a result that is visible to the user, that result is sufficient feedback to the user. If the completion of the action has no visible result, feedback should be provided by a message in a message area.<sup>(32)</sup>

**478. TRANSACTION TITLE:**

There should always be a transaction title on the screen (e.g., **INPUT DESTINATION**). If appropriate to provide context, use subtitles also.<sup>(31)</sup>

**479. GRAMMATICAL FORM FOR QUESTIONS:**

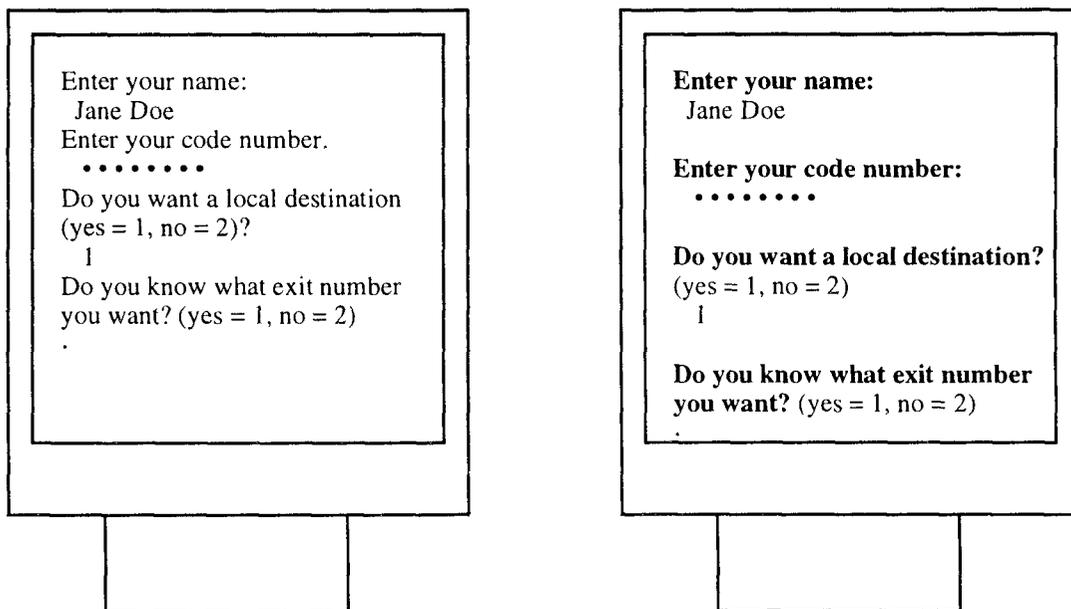
Questions should be stated in a consistent grammatical form (e.g., do not say “Your destination” at one point and “Enter your destination” at another).<sup>(31)</sup>

**480. USE OF NEGATIVES IN QUESTIONS:**

Negatives should be avoided in questions (e.g., do not say “Which sites do you not want to visit?”).<sup>(31)</sup>

**481. DISTINGUISHING AMONG DISPLAYED INFORMATION:**

Visual cues and white space should be used to distinguish questions, prompts, instructions, and user inputs.<sup>(31)</sup> (See figure 34).



Poor interface: Questions, prompts, and user inputs not clearly separated.

Improved interface: White space and visual cues used to separate questions and prompts from each other and from user inputs.

**Figure 34. Use of white space and visual cues to distinguish questions and prompts.**

**482. NUMBER OF CONTROL ACTIONS TO RESPOND TO A QUESTION:**

The number of control actions needed to respond to a question should be minimized (e.g., don't require the user to type "state" when "s" will suffice). (Based on reference 31.)

**483. USE OF CAPITAL LETTERS IN USER INPUTS:**

The system should not require capital letters unless absolutely necessary (e.g., don't require "S" for state). And, if the user does enter a capital letter where it is not required, the system should accept it (i.e., the system should typically not be case sensitive). (Based on reference 31.)

**System Responsiveness**

**484. RESPONSE TIME SPECIFICATIONS:**

Response times for various tasks should conform to the specifications shown in table 34. (See references 13,31,32,42,64.)

**Table 34. System response times.**

Activity	Maximum Response Time	Optimum Response Time
From key depression until a positive response (e.g., key click)	.1 s	Instantaneous to .1 s
From key depression until appearance of the appropriate character	.2 s	-
System activation	3 s	.5 to 1 s
Error feedback following completion of user input	2 to 4 s	.25 to 1 s <sup>1</sup>
Request for simple service (e.g., a routine, single-step operation, such as a request for a page from a menu)	2 s	.25 to .5 s
Request for the next page in a multi-page document	.5 to 1 s	.25 to .5 s
Request to scan a page	-	.5 s
Request for complex service (where a number of operations must be performed)	5 s	2 s
Request to load a program or restart	15 to 60 s	<30 s
Response to a simple status inquiry (e.g., request one category of information about an unambiguously identified object)	2 s	.25 s

*(Table continued on next page.)*

<sup>1</sup> For type-ahead entries with experienced users, error messages should be displayed as quickly as possible. <sup>(42)</sup>

**Table 34. System response times (continued).**

Activity	Maximum Response Time	Optimum Response Time
Response to a complex inquiry in table form (i.e., a request that requires collecting and displaying data on the basis of logical relationships among categories)	2 to 4 s	.25 s
Request for information on the next procedure (i.e., the user is in a conversational interaction with the system, requesting the next in a computer aided or guided task)	<5 s	2 s

**485. RESPONSE TO USER CONTROL ACTIONS:**

The computer should acknowledge every control action immediately; for every action by the user there should be some apparent reaction from the system.<sup>(42)</sup>

**486. RESPONSE TIME VARIABILITY:**

Variability of response times should be kept to a minimum. Total response time deviation should not exceed 50 percent of the mean response time (e.g., if the mean response time is 4 s, the total variation should be limited to 2 s, thus producing a range of response times of 3 to 5 s).<sup>(32)</sup>

**Error Handling**

**487. ERROR MESSAGE CONTENT:**

When the computer detects an entry error, display an error message to the user stating what is wrong and what can be done about it (e.g., say “Code format not recognized; enter two letters followed by three digits” rather than “Invalid input”). Error messages should be specific and understandable (e.g., say “Maximum trip length is 500 miles” rather than “Invalid entry”).<sup>(31,42)</sup>

**488. ERROR MESSAGE WHEN A SMALL SET OF ALTERNATIVES IS AVAILABLE:**

If an entry must be made from a small set of alternatives, an error message that is displayed in response to a wrong entry should indicate the correct alternatives.<sup>(42)</sup>

**489. TONE OF ERROR MESSAGES:**

Wording for error messages should be neutral. Do not imply that the

user is to blame, or personalize the computer, or attempt to make the message humorous (e.g., say “Entry must be a number” rather than “Sorry, I can’t accept a nonnumeric entry”).<sup>(42)</sup>

**490. WHEN MULTIPLE ERRORS ARE DETECTED:**

When multiple errors are detected in a combined user entry, the user should be notified that multiple errors have been found. The complete error messages can then be displayed independently.<sup>(42)</sup>

**491. TIMING OF ERROR MESSAGES:**

The computer should display an error message only after a user has completed the entry.<sup>(42)</sup>

**492. CURSOR POSITIONING FOLLOWING AN ERROR:**

In addition to providing an error message, the cursor should be positioned at the point of the error and the error highlighted.<sup>(31,42)</sup>

**493. WHAT TO DISPLAY FOLLOWING AN ERROR:**

When an entry error has been detected, continue to display the erroneous entry as well as the error message, until the correction is made.<sup>(42)</sup>

**494. USER REQUIREMENT IN ERROR CORRECTION:**

Following error detection, require the user to re-enter only that portion of the entry that is not correct.<sup>(42)</sup>

**495. REMOVING AN ERROR MESSAGE:**

After the error has been corrected, the error message should be removed. Do not continue to display a message that is no longer applicable.<sup>(42)</sup>

**496. USE OF EXCLAMATION POINTS IN ERROR MESSAGES:**

Error messages should not contain exclamation points (e.g., do not say “Maximum trip length is 500 miles!”).<sup>(31)</sup>

**497. USE OF HOSTILE WORDS:**

Error messages should not contain hostile and/or violent words (e.g., “Fatal error,” “Process killed”).<sup>(31)</sup>