Exhibit 1014 – Part 4

number of levels decreased. Over the range studied, the authors suggest that a simple function of the number of items on the screen will predict the time, *T*, for a selection:

$$T = k + c^*\log b$$

where k and c are empirically determined constants for scanning the screen to make a choice, and *b* is the breadth at each level. Then, the total time to traverse the menu tree depends on only the depth, *D*, which is

$$D = \log_b N$$

where *N* is the total number of items in the tree. With $N = 4096$ target items and a branching factor of $b = 16$, the depth, $D = 3$, and the total time is $3^*(k + c^*\log 16)$.

Norman and Chin (1988) fixed the number of levels at four, with 256 target items, and varied the shape of the tree structure. They recommend greater breadth at the root and at the leaves, and added a further encouragement to minimize the total number of menu frames needed so as to increase familiarity. In an interesting variation, Wallace et al. (1987) confirmed that broader, shallower trees (4 x 3 versus 2 x 6) produced superior performance, and showed that, when users were stressed, they made 96 percent more errors and took 16 percent longer. The stressor was simply an instruction to work quickly ("It is imperative that you finish the task just as quickly as possible"); the control group received gentler verbal instruction to avoid rushing ("Take your time; there is no rush").

Even though the semantic structure of the items cannot be ignored, these studies suggest that the fewer the levels, the greater the ease of decision making. Of course, display rates, response time, and screen clutter must be considered, in addition to the semantic organization.

Semantic grouping in tree structures Rules for semantic validity are hard to state, and there is always the danger that some users may not grasp the designer's organizational framework. Young and Hull (1982) examined "cognitive mismatches" in the British Prestel Viewdata system (Martin, 1982). The problems that they identified included overlapping categories, extraneous items, conflicting classifications in the same menu, unfamiliar jargon, and generic terms. Based on this set of problems, the rules for forming menu trees might be these:

- *Create groups of logically similar items*: For example, a comprehensible menu would list countries at level 1, states or provinces at level 2, and cities at level 3.

- *Form groups that cover all possibilities*: For example, a menu with age ranges 0–9, 10–19, 20–29, and greater than 30 makes it easy for the user to select an item.
- *Make sure that items are nonoverlapping*: Lower-level items should be naturally associated with a single higher-level item. Young and Hull offered an example of a poorly designed screen with `Places in Britain` and `Regions of England` as overlapping items on the same menu.
- *Use familiar terminology, but ensure that items are distinct from one another*: Choosing the right terminology is a difficult task; feedback from sample users will be helpful during design and testing.

Menu maps As the depth of a menu tree grows, users find it increasingly difficult to maintain a sense of position in the tree, and their sense of disorientation, or of "getting lost," grows. To overcome this sense of disorientation, some menu systems come with a printed index of terms that is easier to scan than is a series of screen displays. The French Minitel system offers a detailed cross-referenced index that, in 1991, was 62 pages long and contained more than ten thousand entries. The CompuServe Information Service's 1991 index contained almost 3000 subjects; it included a diagram, or map, of the first three levels of the tree structure, which contained 43 menus. The PRODIGY information system uses a cascade approach to show its large menu tree (Figure 3.10a and b).

The relative merits of a map and an index were studied in a small menu structure with 18 animals as target items (Billingsley, 1982). In this case, users who had the chance to study an index did somewhat better than a control group that had no special navigation aids. The group with an overall map did substantially better than did both the index and control groups.

	Control	Index	Map
Number of subjects	10.0	8.0	8.0
Mean time per search	35.3	30.7	19.2
Mean choices per search	12.3	8.4	4.7

Menu learning for a three-level, three-item (3 x 3) menu was studied with four forms of training (Parton et al., 1985). The four forms were as follows:

1. *Online exploration:* Subjects could explore the menus online.
2. *Command sequences:* Subjects studied, on paper the 27 paths through the three levels; e.g., `Plans Division`, `Concepts`, `Systems Analyst`.

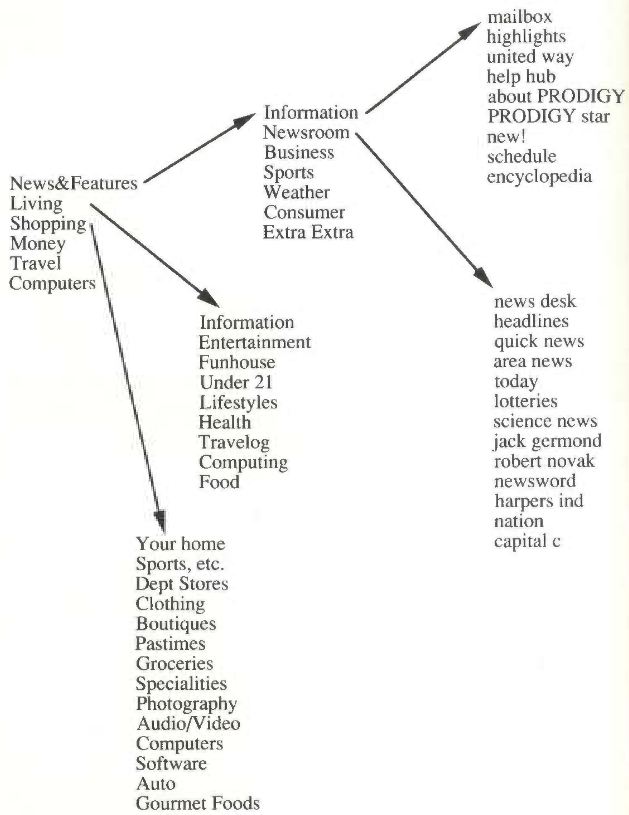


Figure 3.10(a)

The PRODIGY information system. (Figure 3.10a and b: Courtesy of Prodigy Services Company, White Plains, NY.) (a) Partial menu tree.

3. *Frames:* Subjects studied online the 13 menu frames, like this one:

```
Plans Division
Concepts
Designs
Proposals
```

4. *Menu map:* Subjects studied online a tree-structured layout of the 13 frames.

The 65 undergraduate subjects had a 12-minute training period followed by a 10-minute work period. The results indicate a strong advantage for those who had the menu map (Table 3.1).

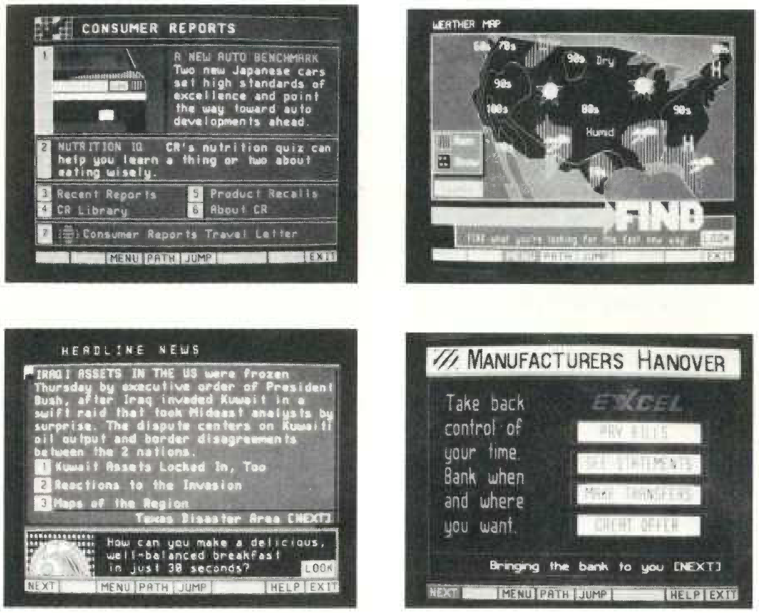


Figure 3.10(b)

Four displays.

Table 3.1

Scores on four dependent variables showed improved performance for subjects who had studied a graphical menu map for a three-level menu. (Source: Parton et al., 1985.)

Variable	Online Exploration	Command Sequences	Frames Map	Menu
Targets found	8.2	4.7	6.5	8.5 n.s.
Average number of menus visited	10.6	20.4	19.6	9.4 p<.10
Recall of tree (max = 27)	10.1	8.4	9.8	16.7 p<.05
Satisfaction (best = 5)	3.6	3.1	2.8	4.8 p<.01

As the tree structure grows, users have greater difficulty in maintaining an overall understanding of the semantic organization. Viewing the structure one menu at a time is like seeing the world through a cardboard tube; it is hard to grasp the overall pattern and to see relationships among categories. Offering a spatial map can help overcome users to this difficulty.

Summary There is no perfect menu structure that matches every person's knowledge of the application domain. Designers must use good judgment for the initial implementation, but then must be receptive to suggested improvements and empirical data. Users will gradually gain familiarity, even with extremely complex tree structures, and will be increasingly successful in locating required items.

3.2.4 Acyclic and cyclic menu networks

Although tree structures are appealing, sometimes network structures are more appropriate. For example, it might make sense to provide access to banking information from both the financial and consumer parts of a tree structure. A second motivation for using networks is that it may be desirable to permit paths between disparate sections of a tree, rather than requiring users to begin a new search from the main menu. These and other conditions lead to network structures in the form of acyclic, or even cyclic, graphs. As users move from trees, to acyclic networks, to cyclic networks, the potential for getting lost increases.

With a tree structure, the user can form a mental model of the structure and of the relationships among the menus. Developing this mental model may be more difficult with a network. With a tree structure, there is a single parent menu, so backward traversals toward the main menu are straightforward. In networks, a stack of visited menus must be kept to allow backward traversals. In a thorough study of 17 subjects using menu networks of 50 frames, Mantei (1982) concluded that "the structure of the user interface . . . causes disorientation if this structure is not obvious to the user."

If networks are used, it may be helpful to preserve a notion of "level," or distance from the main menu. Users may feel more comfortable if they have a sense of how far they are from the main menu.

3.3 Item Presentation 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—then the decision is trivial. Typical bases for sequencing items include these:

- *Time*: Chronological ordering
- *Numeric ordering*: Ascending or descending ordering
- *Physical properties*: Increasing or decreasing length, area, volume, temperature, weight, velocity, etc.

Many cases have no natural ordering, and the designer must choose from such possibilities as:

- *Alphabetic sequence of terms*
- *Grouping of related items* (with blank lines or other demarcation between groups)
- *Most frequently used items first*
- *Most important items first:* Importance may be difficult to decide and may vary among users

Card (1982) experimented with a single 18-item vertical permanent menu of text-editing commands such as `INSERT`, `ITALIC`, and `CENTER`. He presented subjects with a command, and they had to locate the command in the list, move a mouse-controlled cursor, and select the command by pressing a button on the mouse. The menu items were sequenced one of three ways: alphabetically, in functional groups, and randomly. Each of four subjects made 86 trials with each sequencing strategy. The mean times were as follows:

Strategy	**Time per trial**
alphabetic	0.81 seconds
functional	1.28 seconds
random	3.23 seconds

Since subjects were given the target item, they did best when merely scanning to match the menu items in an alphabetic sequence. The performance with the functional groupings was remarkably good, indicating that subjects began to remember the groupings and could go directly to a group. In menu applications where the users must make a decision about the most suitable menu item, the functional arrangement might be more appealing. Users' memory for the functionally grouped items would be likely to surpass their memory for the alphabetic or random sequences. The poor performance that Card observed with the random sequence confirms the importance of considering alternative item presentation sequences.

With a 64-item menu, the time for locating a target word was found to increase from just over 2 seconds for an alphabetic menu to more than 6 seconds for a random menu (McDonald et al., 1983). When the target word was replaced with a single-line definition, the 109 subjects could no longer scan for a simple match and had to consider each menu item carefully. The advantage of alphabetic ordering nearly vanished. User reaction time went up to about 7 seconds for the alphabetic and about 8 seconds for the random organization. Somberg and Picardi (1983) studied user reaction times in finding to which category a target word belonged in a five-item menu. Their three experiments revealed a significant and nearly linear relationship between the user's reaction time and the serial position of the correct category in the menu. Furthermore, there was a significant increase in reaction time if the target word was unfamiliar, rather than familiar.

If frequency of use is a potential guide to sequencing menu items, then it might make sense to vary the sequence adaptively to reflect the current pattern of use. Unfortunately, adaptations can be disruptive, increasing confusion and undermining the users's learning of menu structures. In addition, users might become anxious about other changes occurring at any moment. Evidence against such changes was found in a study in which a pull-down list of food items was resequenced to ensure that the most frequently selected items migrated toward the top (Mitchell and Shneiderman, 1988). Users were clearly unsettled by the changing menus, and their performance was better with static menus. Evidence in favor of adaptation was found in a study of a telephone book menu tree that had been restructured to make frequently used telephone numbers more easily accessible (Greenberg, 1985). However, this study did not deal with the issue of potentially disorientating changes to the menu during usage. To avoid disruption and unpredictable behavior, it is probably a wise policy to allow users to specify when they want the menu restructured.

3.4 Response Time and Display Rate

A critical variable that may determine the attractiveness of menu selection is the speed at which users can move through the menus. The two components of speed are system *response time*, the time it takes for the system to begin displaying information in response to a user selection, and *display rate*, the rate in characters per second at which the menus are displayed (see Chapter 7).

Deep menu trees or complex traversals become annoying to the user if system response time is slow, resulting in long and multiple delays. With slow display rates, lengthy menus become annoying because of the volume of text that must be displayed. In positive terms, if the response time is long, then designers should create menus with more items on each menu to reduce the number of menus necessary. If the display rate is slow, then designers should create menus with fewer items to reduce the display time.

If the response time is long and the display rate is low, menu selection is unappealing, and command-language strategies, in spite of the greater memory demands on the users, become more attractive.

With short response times and rapid display rates, menu selection becomes a lively medium that can be attractive even for frequent and knowledgeable users.

In five studies with 165 adult users of a videotext system, response-time delay pairs (0 versus 10 seconds, 10 versus 15 seconds, and 3 versus 7 seconds) did not yield a statistically significant difference in the preference or performance measures tested (Murray and Abrahamson, 1983). The authors' interpretation was that "inexperienced videotext users are relatively immune to a wide range of constant values of system delay." Other studies have also found that novice users are often pleased with slower response times. However, the large variations in individual performance may have obscured the usual preference for faster response times. Murray and Abrahamson found a significant effect that indicated that large variations in response time led to slower user-response rates.

3.5 Moving Through Menus Quickly

Even with short response times and high display rates, frequent menu users may become annoyed if they must make several menu selections to complete a task. There may be some advantage to reducing the number of menus by increasing the number of items per menu, where possible, but this strategy may not be sufficient. As response times lengthen and display rates decrease, the need for shortcuts through the menus increases.

Instead of creating a command language to accomplish the task with positional or keyword parameters, the menu approach can be refined to accommodate expert and frequent users. Three approaches have been used: allow typeahead for known menu choices, assign names to menus to allow direct access, and create menu macros that allows users to assign names to frequently used menu sequences.

3.5.1 Menus with typeahead—the BLT approach

A natural way to permit frequent menu users to speed through the menus is to allow *typeahead*. The user does not have to wait to see the menus before choosing the items, but can type a string of letters or numbers when presented with the main menu. For example, in the document-printing package in Section 3.2, the user could type `T2N` to get printing at the terminal, double spacing, and no page numbering. The IBM Interactive

System Productivity Facility (ISPF) has numbered choices and allows typeahead with a decimal point between choices (for example, `1.2.1`). Typeahead becomes important when the menus are familiar and response time or display rates are slow, as in many voice-mail systems. Most telephone-inquiry and electronic-mail systems allow the experienced user to enter a string of keypresses at any point in the session.

If the menu items are identified with single letters, then the concatenation of menu selections in the typeahead scheme generates a command name that acquires mnemonic value. To users of a photo-library search system that offered menus with typeahead, a color slide portrait quickly became known as a CSP, and a black-and-white print of a landscape became known as a BPL. These mnemonics come to be remembered and chunked as a single concept. This strategy quickly became known as the *BLT approach*; after the abbreviation for a bacon, lettuce, and tomato sandwich.

The attraction of the BLT approach is that users can gracefully move from being novice menu users to being knowledgeable command users. There are no new commands to learn, and as soon as users become familiar with one branch of the tree, they can apply that knowledge to speed up their work. Learning can be incremental; users can apply one-, two-, or three-letter typeahead, and then explore the less familiar menus. If users forget part of the tree, they simply revert to menu usage.

The BLT approach requires a more elaborate parser for the user input, and handling nonexistent menu choices is a bit more problematic. It is also necessary to ensure distinct first letters for items within each menu, but not across menus. Still, the typeahead or BLT approach is attractive because it is powerful, is simple, and allows graceful evolution from novice to expert.

3.5.2 Menu names for direct access

A second approach to support frequent users is to use numbered menu items and to assign names to each menu frame. Users can follow the menus or, if they know the name of their destination, they can type it in and go there directly. The CompuServe Information Service has a three-letter identifier for major topics, followed by a dash and a page number. Rather than working their way through three levels of menus at 30 characters per second, users know that they can go directly to `TWP-1`, the start of the subtree containing today's edition of *The Washington Post*. Similarly, PRODIGY users can `JUMP` to the `WEATHER` by typing those words.

This strategy is useful if there is only a small number of destinations that each user needs to remember. If users need to access many different portions of the menu tree, they will have difficulty keeping track of the destination names. A list of the current destination names is necessary to ensure that designers create unique names for new entries.

An empirical comparison of the learnability of the typeahead and direct-access strategies demonstrated an advantage for the latter (Laverson et al., 1987). Thirty-two undergraduates had to learn either path names (typeahead) or destination names (direct access) for a four-level menu tree. The direct-access names proved to be significantly faster to learn and were preferred. Different tree structures or menu contents may influence the outcome of similar studies.

3.5.3 Menu macros

A third approach to serving frequent menu users is to allow regularly used paths to be recorded by users as *menu macros*. In other words, users can define their own commands. A user can invoke the macro facility, traverse the menu structure, and then assign a name. When the name is invoked, the traversal is executed automatically. This mechanism allows individual tailoring of the system and can provide a simplified access mechanism for users with limited needs.

3.6 Menu Screen Design

Little experimental research has been done on menu-system screen design. This section contains many subjective judgments, which are in need of empirical validation (Table 3.2).

Table 3.2

Menu selection guidelines that have been distilled from practice, but that still require validation and clarification.

Menu Selection Guidelines

- Use task semantics to organize menus (single, linear sequence, tree structure, acyclic and cyclic networks)
- Prefer broad and shallow to narrow and deep
- Show position by graphics, numbers, or titles
- Use item names as titles for trees
- Use meaningful groupings of items
- Use meaningful sequencing of items
- Make items brief, begin with keyword
- Use consistent grammar, layout, terminology
- Allow typeahead, jumpahead, or other shortcuts
- Allow jumps to previous and main menus
- Consider online help, novel selection mechanisms, response time, display rate, and screen size

3.6.1 Titles

Choosing the title for a book is a delicate matter for an author, editor, or publisher. A more descriptive or memorable title can make a big difference in reader responses. Similarly, choosing titles for menus is a complex matter that deserves serious thought.

For single menus, a simple descriptive title that identifies the situation is all that is necessary. With a linear sequence of menus, the titles should accurately represent the stages in the linear sequence. For the menus in the document-printing package (Section 3.2), the titles might be `Printing location`, `Spacing control`, and `Page numbering placement`. Consistent grammatical style can reduce confusion. If the third menu were titled `How do you want page numbering to be done?` or `Select page numbering placement options`, many users would be unsettled. Excess verbiage becomes a distraction. Brief noun phrases are often sufficient.

For tree-structured menus, choosing titles is more difficult. Such titles as `Main menu` or topic descriptions as `Bank transactions` for the root of the tree clearly indicate that the user is at the beginning of a session. One potentially helpful rule is to use the exact words in the high-level menu items as the titles for the next lower-level menu. It is reassuring to users to see an item such as `Business and financial services` and, after it has been selected, a screen that is titled `Business and financial services`. It might be unsettling to get a screen titled `Managing your money`, even though the intent is similar. Imagine looking in the table of contents of a book and seeing a chapter title such as "The American Revolution," but, when you turn to the indicated page, finding "Our early history"—you might worry about whether you had made a mistake, and your confidence might be undermined.

Using menu items as titles may encourage the menu author to choose items more carefully so that they are descriptive in two contexts.

A further concern is consistency in placement of titles and other features in a menu screen. Teitelbaum and Granda (1983) demonstrated that user think time nearly doubled when the position of information, such as titles or prompts, was varied on menu screens.

In networks of menus, titles become even more important as a guidepost because the potential for confusion is greater. If menu items are made to match the title, then several menus in a network may have the same items. It is satisfying to find the item `Electronic mail` in several menus, but unsettling to find menus with variant terms such as `Electronic mail`, `Sending a note to another user`, and `Communicating with your colleagues`.

3.6.2 Phrasing of menu items

Just because a system has menu choices written with English words, phrases, or sentences, it is not guaranteed to be comprehensible. Individual words may not be familiar to some users, and often two menu items may appear to satisfy the user's needs, whereas only one does. This enduring problem has no perfect solution. Designers can gather feedback from colleagues, users, pilot studies, acceptance tests, and user-performance monitoring. The following guidelines may seem obvious, but we state them 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 `Bike tours`, `Train tours to national parks`, and `Cruise ship tours`.
- *Use consistent and concise phrasing*: The collection of items should be reviewed to ensure consistency and conciseness. Users are likely to feel more comfortable and 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 left:* Try to write menu items so that the first word aids the user in recognizing and discriminating among items. Users scan menu items from left to right; if the first word indicates that this item is not relevant, they can begin scanning the next item.

3.6.3 Graphic layout and design

The constraints of screen width and length, display rate, character set, and highlighting techniques strongly influence the graphic layout of menus. Presenting 50 states as menu items was natural for the Domestic Information Display System built by NASA on a large screen with rapid display rate. On the other hand, the CompuServe Information Service, which must accommodate microcomputer users with 40-column displays over 30-character-per-second telephone lines, used the main menu page shown in Figure 3.11. An improved menu with greater breadth and more distinctive terms was introduced in 1985 (Figure 3.12). As users move down the tree, they find the page numbers always displayed at the upper right, a title, numbered

Figure 3.11

Early version of CompuServe main menu. The items are not sufficiently distinctive; for example, users would have a hard time deciding where to look for programs to assist them with home checkbook management. (Courtesy of CompuServe, Incorporated, Columbus, OH.)

```
COMPUSERVE                    PAGE CIS-1

   COMPUSERVE INFORMATION SERVICE

1 HOME SERVICES
2 BUSINESS & FINANCIAL
3 PERSONAL COMPUTING
4 SERVICES FOR PROFESSIONALS

5 USER INFORMATION
6 INDEX


ENTER YOUR SELECTION NUMBER,
OR H FOR MORE INFORMATION.
```

choices, and instructions. This consistent pattern puts users at ease and helps them to sort out the contents. Menu designers should establish guidelines for consistency of at least these menu components:

- *Titles*: Some people prefer centered titles, but left justification is an acceptable approach, especially with slow display rates.
- *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, down the columns).

Figure 3.12

Revised CompuServe main menu with more items and more distinctive separation among items. Compare to Figure 3.11. (Courtesy of CompuServe, Incorporated, Columbus, OH.)

```
CompuServe                                TOP

 1 Instructions/User Information
 2 Find a Topic
 3 Communications/Bulletin Bds.
 4 News/Weather/Sports
 5 Travel
 6 The Electronic MALL/Shopping
 7 Money Matters/Markets
 8 Entertainment/Games
 9 Home/Health/Family
10 Reference/Education
11 Computers/Technology
12 Business/Other Interests

Enter choice number !
```

- *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 an unacceptable choice, the message should appear in a consistent position.
- *Status reports*: Some systems indicate which portion of the menu structure is currently being searched, which page of the structure is currently being viewed, or which choices must be made to complete a task. This information should appear in a consistent position.

Consistent formats help users to locate necessary information, focus users' attention on relevant material, and reduce users' anxiety by offering predictability.

In addition, since disorientation is a potential problem, techniques to indicate position in the menu structure can be useful. In books, different fonts and typeface indicate chapter, section, and subsection organization. Similarly, in menu trees, as the user goes down the tree structure, the titles can be designed to indicate the level or distance from the main menu. If different fonts, typefaces, or highlighting techniques are available, they can be used beneficially. But even simple techniques with only upper case characters can be effective; for example,

```
***************************
*        MAIN MENU        *
***************************
```

followed by

```
* * * HOME SERVICES * * *
```

followed by

```
- - NEWSPAPERS - -
```

followed by

```
New York Times
```

gives a clear indication of progress down the tree. When traversal back up the tree or to an adjoining menu at the same level is done, the user has a feeling of confidence in the action.

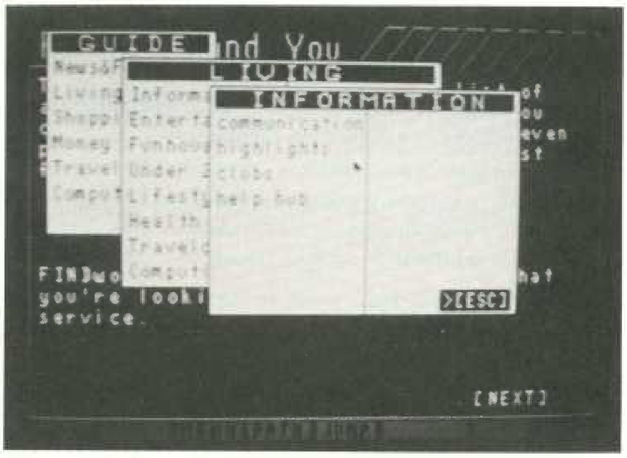


Figure 3.13

Cascade of menus from PRODIGY Information Service. (Courtesy of Prodigy Services Company, White Plains, NY.)

With linear sequences of menus, the users can be given a simple visual presentation of position in the sequence by the use of a *position marker*. In a computer-assisted instruction sequence with 12 menu frames, a position marker just below the menu items might show progress. In the first frame, the position marker was +---------, in the second frame it was -+---------, and in the last frame it was ----------+. The users can gauge their progress, and can see how much remains to be done. The position marker served to separate the items from the instructions in a natural way, and the position was indicated in a nonobtrusive manner.

With graphic user interfaces, many possibilities exist for showing successive levels of a tree-structured menu or progress through linear sequences. A common approach is to show a cascade of successive menu boxes set slightly lower than and slightly to the right of the previous items (Figure 3.13). For pull-down menus, *walking menus* are perceptually meaningful, but can present a motor challenge to move the cursor in the appropriate direction (Figure 3.14).

With rapid high-resolution displays, more elegant visual representations are possible. With enough screen space, it is possible to show a large portion of the menu map, and to allow users to point at a menu anywhere in the tree. Graphic designers or layout artists may be useful consultants in design projects.

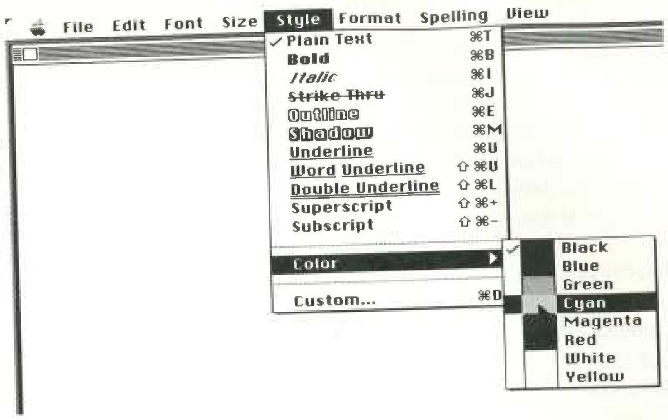


Figure 3.14

Walking menus are a motor challenge to users who must move the cursor down to the proper item and then carefully to the right to produce the submenu. This example from Claris's MacWrite II shows selection of colors for text. (Courtesy of Claris Corp., Santa Clara, CA.)

3.7 Selection Mechanisms

At first glance, choosing the menu-selection mechanism appears to be a minor design decision that can be made quickly, so that the design team can get on to more important matters. On the other hand, the selection mechanism is the central aspect of the menu system for most users (pointing devices, such as the mouse, are covered in Section 3.8). For keyboard-oriented systems this issue might be simplified to this question: Should the designer use numbers or letters for indicating menu items?

The arguments in favor of *numbered items* are that there is a clear sequencing of items, and that even nontypists can find the numbers on the keyboard. In some systems, numeric keypads are the only input device. Sequential numbering is satisfying because the user can see quickly how many items there are, and visual scanning is aided by the natural numeric ordering. As users scan down the items, they can use the numbers as a guide to make sure that they review each choice. When menu items have a natural numeric sequence—for example, the twelve months of the year, the chapters of a book, or the days of the week—numbered choices are appealing.

The disadvantage of numbers is that, when there are more than 10 items, two keypresses are required to make a selection. Another problem with using numbers only is that, if there are standard menu items such as `HELP` or

BACK TO MAIN MENU, then these items may have a different number on each screen. If there is no natural numbering of menu items, then the numbering may be misleading, somehow indicating preference for item 1. Attaching numbers to a group of colors or of bank-loan plans may mislead the user into believing that there is some hidden sequencing or preference.

If lettered items are used for menu items, then there is the choice between ABCDEF...lettering (sequential) and meaningful letter choices (mnemonic). Sequential lettering is similar to numbering, but 26 choices are available before two keypresses are required. There is some evidence that there is less likelihood of a keying error with letters than with numbers, because the letters are more spread out on the keyboard. It may be a bit more tricky for someone unfamiliar with a typewriter keyboard to locate the proper letter, but this problem does not appear to be a serious hindrance. Mnemonic lettering for menu items is appealing because the congruence between the description of the item and the keypress can build user confidence in the task. For example, it makes sense that T is for TRANSFER and W is for WITHDRAWAL.

Of course, there are mixed strategies. Some systems, such as CompuServe, use numbers for the primary menu items and letters for generic functions, such as M to get to the previous MENU and H to get to HELP information. This approach solves some problems and helps to clarify the grouping of menu items. Perlman (1984) found user think times to be lowest with mnemonic letter items and highest with sequential (and therefore nonmnemonic) letter items. Numbered items produced a middle level of user think time.

3.7.1 Typeahead selections

We cannot make the design decision without looking at the larger issue of tasks that require several menu selections. If a sequence of menus is to be viewed, the mnemonic-lettering approach gains substantially because the user can remember sequences such as TCS, for "Transfer from Checking to Saving," more easily than 253. If the user can type these selection letters before seeing the full menu, then the mnemonic-lettering approach becomes a command language for the frequent user. This typeahead approach (Section 3.5.1) is powerful, since it makes the same system appealing to novices and frequent users. Furthermore, it facilitates the graceful evolution from novice to expert—users type ahead only as much as they can remember, and then examine the next menu.

3.7.2 System evolution

Another advantage of mnemonic lettering is that, as items are added to menus, there is no need to renumber the other choices. Mnemonic lettering does have the problem of collisions—that is, of more than one choice with the same first letter. Collisions present a serious concern, but often an acceptable alternate term can be found. If not, then using more than one letter of the term may be necessary.

3.7.3 Mixed letters and numbers

If numeric data entry is to be made on some menu or data-entry screens, then the lettered-item approach will be advantageous since the typeahead command string may be more comprehensible. For example, the direction to deposit $40.00 in savings account 38847 might be entered as `D40.00S38847`, which is more appealing than `340.00638847`. On the other hand, if the data entry is for alphabetic strings, then the numbered approach might yield a more comprehensible command string. The alternation of letters and numbers helps to break a string into more meaningful chunks.

3.7.4 Arrow-key movement of highlight bar

Instead of typing a choice, users can move a highlight bar to the intended item. The cursor could be moved by the arrow keys, tab key, or space bar. This approach is appealing to novice users for single screen selections, even though there may be more keystrokes and the RETURN key must be pressed. There is a great sense of satisfaction in being able to move the highlight bar among the items. The menu item is highlighted clearly on the screen and in the user's mind, user confidence is high, and screen space is conserved since item labels are not needed. Brightness, underscoring, boxes, color, or reverse video can be used to indicate the item that has been selected. Of course, this approach does not lend itself to typeahead schemes.

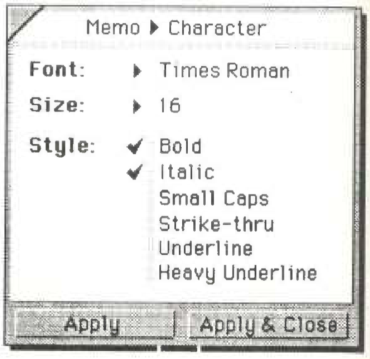
3.8 Graphical User-Interface Menu Features

With modern graphical user interfaces, menu selection by mouse clicks becomes the common method of working with pull-down and pop-up menus plus the ubiquitous dialog boxes containing radio buttons, check boxes, text-entry fields, and scrollable lists of choices (Figure 3.15). The key issues are how to show where selection is possible (the *affordance*), which item has been selected (*highlighting*), whether *de-selection* is possible, and how *activation* is invoked. The mouse, touchscreen, or stylus for pointing at sets of buttons or on scrollable lists are widely appreciated for being rapid and direct (see Chapter 5). The graphical environment allows more information to be conveyed in the menu through use of, for example, the actual fonts in a font selection menu or the actual colors in a color-selection menu. Items

Figure 3.15

Dialog box from the PenPoint stylus-based user interface shows typical graphical user interface features: scrolling menus for character font and size, plus a set of six check boxes for the style. Buttons on the bottom apply the selections and close the dialog box. Close triangle is in the upper left. (Courtesy of GO Corp., Foster City, CA.)

Character Options for PenPoint Text Component



not available in the current context can be grayed out to hold their place in the menu while showing their current unavailability. Spatial placement of items can also be helpful in showing relationships among items and in guiding users through a sequence of selections. Boxes around items, white spaces, varying font sizes, and use of color can help to organize the display. Iconic menus are discussed in Section 5.4.

Graphical approaches will continue to gain adherents where the available technology supports this style of design. Three-dimensional affordances and lighting models with shading are eye-catching and novel, but they risk being distracting and taking more screen space (see Chapter 9).

3.9 Embedded Menus

All the menus discussed thus far might be characterized as *explicit menus* in that there is an orderly enumeration of the menu items with little extraneous information. In many situations, however, the menu items might be *embedded* in text or graphics and still be selectable.

When we designed a textual database about people, events, and places for a museum application, it seemed natural to allow users to retrieve detailed information by selecting a name in context (Koved and Shneiderman, 1986). Selectable names were highlighted, and users could move a reverse-video bar that jumped among highlighted names by pressing the four arrow keys (Figure 3.16). They made selections by pressing ENTER, and they obtained a new article title plus the option of returning to the previous article title. The names, places, phrases, or foreign-language words were menu items embed-

WASHINGTON, DC: THE NATION'S CAPITAL PAGE 2 OF 3

Located between **Maryland** and **Virginia**, Washington, DC embraces the **White House** and the **Capitol**, a host of **government offices** as well as the **Smithsonian museums**.
Designed by **Pierre L'Enfant**, Washington, DC is a graceful city of broad boulevards, **national monuments**, the rustic **Rock Creek Park**, and the **National Zoo**.
First-time visitors should begin at the **mall** by walking from the **Capitol** towards the **Smithsonian museums** and on

---

SMITHSONIAN MUSEUMS: In addition to the familiar castle and popular Air & Space Museum there are 14 other major sites.
SEE ARTICLE ON "SMITHSONIAN MUSEUMS"

BACK PAGE NEXT PAGE RETURN TO "NEW YORK CITY" EXTRA

Figure 3.16

Embedded menus in this early version of Hyperties improved comprehensibility over numbered menu lists and lowered anxiety for novice users. A reverse-video selector box initially covers the NEXT PAGE command. Users move the selector box over highlighted references or commands, and then select by pressing ENTER. A touchscreen version allows selection by merely touching the highlighted reference or command. (Created in 1983–1985 by Human–Computer Interaction Laboratory, University of Maryland, College Park, MD; distributed and refined by Cognetics Corporation, Princeton Junction, NJ.)

ded in meaningful text that informed users and helped to clarify the meaning of the items. Subsequent implementations used mouse selection or touchscreens.

Embedded menus have emerged in other applications. Air-traffic control systems allow selection of airplanes in the spatial layout of flight paths to provide more detailed information for controllers. Geographic display systems allow selection of cities or zooming in on specific regions to obtain more information (Herot, 1984). In these applications, the items are icons, text, or regions in a two-dimensional layout.

Hypertext (Section 11.4) program browsers allow software engineers to explore programs by simply pointing and clicking on variables to get the data declaration, or on function invocations to get the function definition (Seabrook and Shneiderman, 1989). Many spelling checkers use the embedded-menu concept by highlighting possibly misspelled words in the context of the words' use. The author of the text can move a cursor to a highlighted word and request alternate corrected spellings; or can type in the correctly spelled word.

Embedded menus permit items to be viewed in context and eliminate the need for a distracting and screen-wasting enumeration of items. Contextual display helps keep the users focused on their tasks and on the objects of interest. Items rewritten in list form may require longer descriptions (of the items) and may increase the difficulty of making selections because of confusion arising from cross-referencing between the menu and the context.

3.10 Form Fillin

Menu selection is effective for choosing an item from a list, but some tasks are cumbersome with menus. If data entry of personal names or numeric values is required, then keyboard typing becomes more attractive. The keyboard may be viewed as a continuous single menu from which multiple selections are made rapidly. When many fields of data are necessary, the appropriate interaction style might be called *form fillin*. For example, the user might be presented with a purchase-order form for ordering from a catalog, as in Figure 3.17.

The form-fillin approach is attractive because the full complement of information is visible, giving the users a feeling of being in control of the dialog. Few instructions are necessary, since this approach resembles famil-

```
Type in the information below, pressing TAB to move the cursor,
and press ENTER when done.

   Name: ______________________     Phone: (___) ___-____

Address: ______________________

   City: ______________________     State: __  Zip Code: _____

Charge Number: ____ ____ ____ ____

    Catalog                      Catalog
    Number        Quantity       Number       Quantity

    _______       _______        _____        _____
    _______       _______        _____        _____
    _______       _______        _____        _____
    _______       _______        _____        _____
```

Figure 3.17

A form-fillin design for a department store.

iar paper forms. On the other hand, users must be familiar with keyboards, use of the TAB key to move the cursor, error correction by backspacing, field-label meanings, permissible field contents, and use of the ENTER key. Form fillin must be done on displays, not on hardcopy devices, and the display device must support cursor movement.

An experimental comparison of database update by form fillin and by a command-language strategy demonstrated a significant speed advantage for the former (Ogden and Boyle, 1982); eleven of the 12 subjects expressed a preference for the form-fillin approach.

3.10.1 Form-fillin design guidelines

There is a paucity of empirical work on form fillin, but a number of design guidelines have emerged from practitioners (Galitz, 1980; Pakin and Wray, 1982; Brown, 1986). Many companies offer form-fillin creation tools, such as Hewlett-Packard's VPLUS, IBM's ISPF, DEC's FMS, Ashton-Tate's dBASE, and Borland's Paradox. 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 (Table 3.3).

The elements of form fillin design include the following:

- *Meaningful title*: Identify the topic and avoid computer terminology.
- *Comprehensible instructions*: Describe the user's tasks in familiar terminology. Try to be brief; if more information is needed, make a set of help screens available to the novice user. In support of brevity, just

Table 3.3

Form fill-in guidelines distilled from practice, but in need of validation and clarification.

Form Fillin Design Guidelines

- Meaningful title
- Comprehensible instructions
- Logical grouping and sequencing of fields
- Visually appealing layout of the form
- Familiar field labels
- Consistent terminology and abbreviations
- Visible space and boundaries for data-entry fields
- Convenient cursor movement
- Error correction for individual characters and entire fields
- Error messages for unacceptable values
- Optional fields marked clearly
- Explanatory messages for fields