Chapter 5: Designing and Evaluating DSS User Interfaces

Contents

<i>I</i> .	Introduction	2
<i>II</i> .	User Interfaces: An Overview	2
III.	User Interface Styles	4
IV.	ROMC Design Approach	7
<i>V</i> .	Building the DSS User Interface	
VI.	Comments on Design Elements	
VII.	Guidelines for Dialog and User Interface Design	12
VIII	I. Factors Influencing User Interface Design Success	13
IX.	Discussion and Commentary	
Х.	Audit Questions	16
Que	estions for Review	16
Que	estions for Discussion	
Inte	rnet Exercises	17
XI.	References	17
XII.	DSS User Interface Web References	
XIII	I. Appendix 5.1 User Centered Design Guidelines	

I. Introduction

An effective user interface is an important component of any type of Decision Support System, but it is especially important for systems that will be used directly by managers. In a Decision Support System, the user interface is sometimes called the dialogue or "frontend" component. Some might ask "why is the user interface or dialogue component of a DSS so important?" Research indicates that the easier it is to use a DSS interface, then the more "user friendly" most people will consider the system and hence the greater the chance that managers will actually use the DSS. The user interface is what managers see when they work with a DSS.

Does the user interface of a DSS differ from the user interface for any other computerized system? What guidelines can improve DSS user interfaces? Managers should be the primary users of DSS so a DSS user interface should help managers interact with the system. When a DSS user interface is complex or difficult to use a staff person may need to assist in using a DSS. So some user interfaces can increase the operating costs of a DSS and probably limit its use. Can we avoid this outcome? If so, how? This chapter presents the basics of user interface design and the technologies that are used to create effective user interfaces. The focus in the chapter is on how display screens should look and how a DSS user interface, design elements, guidelines for dialog and user interface design, and factors influencing user interface success.

The goal of user interface design is developing screen layouts and interfaces that are easy to use and that are visually attractive (cf., Galitz, 1985). Both the intended users of a Decision Support System and information systems designers need to participate actively in designing and evaluating DSS user interfaces. Let's examine the concept of a user interface.

II. User Interfaces: An Overview

Many Decision Support Systems users have limited computing expertise. Most of these users do **NOT** want to learn a command language interface like Structured Query Language (SQL) that may be used by an expert or by a more technically-oriented user. According to Bennett (1986), for a non-technical user the design of an appropriate DSS user interface is the most important determinant of the success of a decision support implementation. So what is a user interface?

A user interface is what managers see and use when they interact with a DSS. More specifically, a user interface is the set of menus, icons, commands, graphical display formats, and/or other representations that are provided by a software program to allow a user to communicate with and use the program. A graphical user interface (abbreviated GUI which is pronounced "goo ee") provides a user a more or less "picture-oriented" way to interact with computing technology. GUIs remain controversial. Many people argue a GUI is the most user-friendly interface to an information or Decision Support System. Some people disagree strongly with this conclusion. User-friendly is an evaluative term for

a computerized system's user interface. It indicates that users judge the interface as easy to learn, understand, and use.

Also, a user interface refers to the hardware and software that creates communication and interaction between a DSS user and the computer processor. The user interface includes responses and involves an exchange of graphic, acoustic, tactile, and other signs. User interface research is a subset of a field called human-computer interaction (abbreviated HCI). HCI focuses on the study of people, computer technology, and the way each influences the other.

An effective user interface is important because data and graphics displayed on a computer workstation screen provide a context for human interaction and provide cues for desired actions by the user. The user formulates a response to the context and takes an action. Data then passes back to the computer through the interface.

A well-designed user interface can increase human processing speed, reduce errors, increase productivity and create a sense of user control. The quality of the system interface, from the user's perspective, depends upon what the user sees or senses, what the user must know to understand what is sensed, and what actions the user can and in some cases must take to obtain needed results.

MI&DSS Book	Service Services		in state of the st		10 A
Ele Edt View Go Favories	Heb		Colling Colorester		and the second
⇒ . ⇒ . tul Back Forward Up	Cut Care Paste	LAT Und) X C	her Views	
Addem 🔯 C:Wy Documents Wil6DSS	Book				
	Name	See	Type	Modified	
	C dictation		File Folder	8/2/1999 9:44 PM	
	🛄 dss forum materials		File Folder	7/31/1999 7:27 PM	
MT&DSS Book	MIS course materials		File Folder	7/31/1999 7:27 PM	
PILQU35 DOOK	inew chapters		File Folder	7/31/1999 7:37 PM	
	old chapters		File Folder	7/31/1999 7:24 PM	
Select an item to view its	student papers		File Folder	8/1/1999 8:36 PM	
description.	MI&DSSbookoutine	62KB	Netscape Hypertext	8/2/1999 5:10 PM	
	i outine	3KB	Text Document	8/2/1999 18:25 PM	

Figure 5.1 Screen shot of part of a Windows 98 browser type of graphical interface.

To create a well-designed user interface, MIS professionals need to work closely with potential users. Both groups need to be familiar with the following important issues and topics related to building and evaluating a user interface:

- 1. User interface style Is the style or combination of styles appropriate? What styles are used in the user interface?
- 2. Screen design and layout -- Is the design easy to understand and attractive? Is the design symmetric and balanced?
- 3. **The Human-Software interaction sequence** -- Is the interaction developed by the software logical and intuitive? Do people respond predictably to the interaction sequence?
- 4. Use of colors, lines and graphics -- Are colors used appropriately? Do graphics improve the design or distract the user?

- 5. **Information density** -- Is too much information presented on a screen? Can users control the information density?
- 6. Use of icons and symbols -- Are icons understandable?
- 7. Choice of input and output devices -- Do devices fit the task?

Managers and analysts should focus on these seven design issues when they evaluate a DSS prototype or the proposed screens for a DSS. A systematic evaluation of the DSS user interface can substantially improve its usefulness and increase how much it will be used. Let's examine some of these issues in more detail.

III.User Interface Styles

The user interface determines how information is entered in a DSS and how it is displayed by a DSS. The interface also determines the ease and simplicity of learning and using the system. There are four general structures or interface styles that can be used to control interactions with computerized information systems. These styles are 1) command-line interfaces, 2) menu interfaces, 3) point-and-click graphical interfaces, and 4) question-andanswer interfaces. Each style can be used in creating DSS user interfaces. The styles can often be combined usefully in a single application or set of related applications [see Galitz (1985); Shneiderman (1992); and Turban (1993)]. When building a user interface a designer should try to provide multiple ways to perform the same task. For example, a design may include a command-line interface, pull down menus for commands, and keyboard command equivalents. Many input devices including keyboard, mouse, touch pad, and voice inputs can be used to manipulate these four general interface styles.

Command-line Interfaces

Command-line interfaces are the oldest form of computer control. They originated when each command to a program was entered on a punched card. Commands still dominate user interaction with operating systems, including MS-DOS, UNIX, and Linux. In a DSS with a command language style interface, a user enters a command such as "run" or "plot". Many commands are composed of a verb-noun combination (for example "plot sales"). Command-line interfaces require a user to enter a command telling the system what to do next. It is the user's responsibility to know what commands are available and how to phrase those commands with their parameters. Such interfaces can be quite powerful, giving their users detailed control over system operation, but there is a penalty. Command interfaces are hard to learn. Most people never learn more than a fraction of the commands in any command language and make frequent mistakes in command entry. While command entry mistakes can usually be corrected, they create a cost for users and companies in terms of productive time lost and they can make users feel frustrated and incompetent.

Menu Interfaces

In a menu interaction a user selects from a list of possible choices the task or function to be performed. The ordered list of functions or tasks is called a menu. The user makes a choice among items by manipulating an input device or entering a menu item number. Menus

appear in a logical, hierarchical order, starting with a main menu and going to subordinate or sub-menus. Menus can become tedious and time consuming when complex situations are analyzed, since it may take several menus to build or use a system and the user must shift back and forth among the menus. A pull-down menu is a sub-menu that appears as a superimposed drop-down menu on a screen, usually after an entry has been made in a high level menu. Figure 5.2 shows a menu hierarchy in Microsoft Excel. The tools bars with graphical icons also serve as menus.

🗙 Microsoft Excel - Boo	k1	
📔 🔁 Eile Edit View Inser	: F <u>o</u> rmat <u>T</u> ools <u>D</u> ata <u>W</u> indow <u>H</u> elp	
🛛 🗅 🚅 🖬 🎒 🖪 🖤	🕺 🖟 💱 Spelling F7 🛛 😵 Σ 🏂 🛃 🛍	9 🕹
📗 Arial 🔹	10 ▼ : <u>AutoCorrect</u> Look Up Reference	
A1 🔽		
A B 1	C Share Workbook F G Irack Changes Irack Changes Irack Changes Merge Workbooks Irack Changes Irack Changes Protection Irack Changes Irack Changes Irack Changes Irack Changes Irack Changes Merge Workbooks Irack Changes Irack Changes Irack Changes Irack Changes Irack Changes Irack Chang	
9 10 11 12 13 14 15 16	Macro Macros Add-Ins Record New Macro Qustomize Sisual Basic Editor Wizard Mission	

Figure 5.2 An example of a menu hierarchy in Microsoft Excel.

Menu screens are effective because they rely on recognition rather than recall. Working with menus reminds users of available options. The design of menus must consider the conflicting needs of experienced and inexperienced users.

Graphical Interfaces

A graphical user interfaces (GUI) is an interface system in which users have direct control of visible objects. Users point and click to initiate actions rather than enter complex commands. The best known GUI is the Windows 95/98 operating system followed closely by the Macintosh OS. The major GUI elements are windows, icons, pull-down menus, and dialog boxes. A window is an area of the computer screen that behaves as if it was an independent computer terminal. Icons are small pictures that represent windows or actions. Some of the icons frequently used in Microsoft applications are shown in Figure 2. Clicking on an icon initiates opening a window or initiating a command. In the graphical or

object manipulation interface style, the user directly manipulates objects represented as icons (or symbols).



Figure 5.3 Examples of Icons

User interfaces can be enriched with the use of multimedia and hypermedia technologies. Multimedia refers to many media including graphic materials, audio especially sound digitizing, and motion images including motion pictures and animation. Hypermedia describes documents that contain several types of media linked in documents. The World-Wide Web is an example of a hypermedia delivery system. Web documents can include explicit internal and external links, multimedia content, and interactivity with databases.

Question-and-Answer Interfaces

A question and answer interface dialogue begins with the computer asking the user a question. The user answers the question with a phrase or a sentence. A dialogue then occurs between the computer and user. The computer's questions are a function of prior responses of the user and the processing needs of the application. A related interface style is called form interaction; in form interaction style the user enters data or commands into designated spaces (fields) in a form. The headings of the form serve as a prompt for the desired input. A human-computer interaction that is similar to a human-human dialog is referred to as natural language dialogue. The major limitation of using natural language responses is the inability of the computer to really understand unstructured or unanticipated natural language. The programmer must anticipate user answers and program responses.

The following example shows a simple question and answer dialogue:

>dss What is your name?
>user Daniel Power
>dss What is your age?
>user 39 years old
>dss Please enter only a number

A question and answer dialogue is one of the oldest types of interfaces; it is not used as frequently today in building DSS, but it may be revived by improvements in speech recognition technologies. Another new type of interface is called a three-dimensional (3-D) or virtual reality (VR) interface. It is being used in a number of research settings. With a 3-D interface, user interacts with a 3-D computer-generated environment. A user wears a headset and hand position sensor to interact with the 3-D environment. The user can walk

around, grasp and move objects, and in general alter the environment. A 3-D interface may become a viable DSS user interface in the future, but for the next few years managers and DSS analysts should focus on the interfaces discussed in preceding paragraphs. In most DSS more than one interface style will be implemented.

IV. ROMC Design Approach

Sprague and Carlson (1982) presented an approach for designing DSS and especially the user interface called **ROMC**. Their approach has four user-oriented entities: 1) **R**epresentations for conveying information to the user, 2) **O**perations for manipulating data displayed as representations, 3) aids for a user's memory (**M**emory aids), and 4) aids for helping users control the DSS (**C**ontrol aids).

This section describes the four components of Sprague and Carlson's approach and provides examples of each component. ROMC was intended as a process-independent approach for identifying the necessary capabilities of a DSS. It can also serve as a framework for creating screen designs and for building the user interface of a DSS. Considering the four components can improve screen design and layout.

Representations

In a DSS decision-making activities take place in the context of a conceptualization of the information used in the activity. The conceptualization may be an icon, a chart, a map, a text document, a form, a spreadsheet, a picture, a few numbers, or an equation. The conceptualization is a physical representation that helps a decision-maker communicate about the decision with another person.

Representations provide a context in which users can interpret DSS outputs and select DSS operations. Representations also can be used to supply parameters for DSS operations. For example, a point selected on a graph or a map can be linked to a data value, a document or a database query. Also, prioritizing a list of employees may be the primary input for a personnel scheduling DSS. Managers and DSS analysts need to evaluate and choose appropriate representations.

Operations

Operations are specific tasks that a decision-maker can perform with a DSS. For example, a DSS may have operators to gather data, generate a report, retrieve alternatives, rate alternatives, add alternatives, etc. Note that an operation may be used in more than one activity and that there is usually no prespecified ordering of operations. Analysts need to decide how operations will be controlled from the user interface. Will menus be used? Icons? What names will be used for operations?

Memory Aids

Several types of memory aids should be provided in a DSS user interface to support the use of representations and operations. A symbolic link to a data warehouse is a memory aid for decision-makers. Triggers or rules remind a decision-maker that certain operations may need to be performed. A user profile or data filters may make operation of the DSS easier. User established links or command sequences could make the DSS easier for that user to manipulate.

A trigger may invoke an operation automatically or remind the DSS user to invoke the operation. A profile can store initial defaults for using the DSS. Users logs of actions taken and operations invoked are also memory aids, especially if the user can backup and undo actions of replay actions. DSS analysts need to identify needs for memory aids and decide how reminders will be displayed. The help system is an important memory aid that must be designed as part of the user interface. A mouse over label for an icon is also a memory aid.

Control Aids

DSS control aids are intended to help decision-makers use representations, operations and memory aids. Control aids help decision-makers direct the use of the DSS. One type of control aid focuses on the standard conventions for user-system interaction, which are enforced across representations and operations. This type of control aid uniformly displays menus or defines guidelines for icons design and behavior. Some operations are more system oriented than decision process oriented and these operations are also control aids. Edit, delete, and save operations are generic control operations and hence they are also control aids for the DSS. The tools used to create the user interface constrain the control aids. User interface design guidelines should also standardize the "look and feel" of the user interface.

V. Building the DSS User Interface

The ROMC framework can be a useful tool for designing the DSS user interface. Also, the ROMC specification of elements along with screen layouts can aid in implementing the actual DSS user interface. Every DSS will have a specific set of representations, operations, memory aids, and control aids. The generality and usefulness of a DSS will depend on the skill of the designers in selecting design elements.

Flow-charting the existing or a desired decision process can help develop the ROMC framework. A decision process flowchart should focus on the inputs, operations and outputs of each decision task. The resulting DSS design will likely follow the flowchart and its sequencing of tasks. The resulting representations may be effective, but the operations and control aids developed from this approach may provide limited flexibility to the decision-maker. Creating prototypes of the DSS screens early in the analysis process, and then eliciting input from potential users can reduce the problem of limited flexibility in the operations of a DSS.

Screen designs and layouts should be aesthetically pleasing. The design does not need to be "artistic" but it should not create a negative impression. Managers and designers should

evaluate a DSS user interface in terms of balance, symmetry, proportion and arrangement. Balance means the design elements are equally weighted on the screen. Symmetry refers to correspondence in size and shape of the design elements. Proportion is a harmonious relation among the parts. Arrangement is the ordering of elements. A balanced, symmetric screen design is the easiest screen layout to create and it is generally pleasing. Working with unbalanced and asymmetric screen designs is much more difficult for most of us. Figure 5.4 provides an example of a simple screen design. Do you think the design is balanced and symmetric?



Web Books Sales Analysis - Fall 1999

Figure 5.4 An example of a simple, balanced screen design.

Keen and Gambino (in Bennett, 1983, p. 168) provide the following suggestions for building a DSS user interface. They believe rapid prototyping and adaptive design is essential; they argue any systems analyst, programmer, or consultant who wants to build Decision Support System must know how to:

a. **Get started**. A DSS application does not come packaged with neat specifications. Start with an initial user interface design. It provides a means of learning from and responding to the user.

b. **Respond quickly**. A DSS must evolve rapidly and designers must learn quickly. The design structure and programming techniques must facilitate evolution and learning.

c. **Pay close attention to user-system interfaces and outputs.** A DSS is a set of relatively simple components that must fit together to permit complex, varied, and idiosyncratic problem solving. The designer needs to get a very detailed understanding of the task to be supported and of the people who carry out the task.

According to Keen and Gambino the "natural sequence and order of priority" in developing a DSS is the following four steps:

1. Design the user interface and dialog.

- 2. Design commands and operations in terms of the users' processes and concepts.
- 3. Define what the user does and sees when a command is invoked.
- 4. Work backward to create the program logic and data management.

An alternative design approach is reviewed by Silver (1991). He notes the following steps are appropriate for developing a DSS user interface:

- 1. Determine who your user is.
- 2. Determine what the user will do with the system.
- 3. Determine what sequence of steps the user must follow to accomplish a task.
- 4. Diagram the steps in item 3 and the decision tree involved. Review them with the user.
- 5. Determine which of these steps require interaction with the system.
- 6. Determine information and decision requirements for each interaction (both system and user).
- 7. Select the categories of dialogue (menus, prompts, forms, etc.).
- 8. Diagram the flow of dialogue, showing all decisions and their information requirements. Review these with the user.
- 9. Design screens.
- 10. Try it, analyze it, simplify it, change it, try it . . .
- 11. Update the decision diagrams.
- 12. Bulletproof the dialogue by asking what happens if the user does something unexpected?

While Silver's list focuses on a broad set of steps; steps 4 to 11 are an iterative design process. Even if a DSS analyst plans to develop a DSS user interface using rapid prototyping, it is important to understand who your DSS user is, what the system will be used for, and what sequence of steps a user will follow. A designer may be able to skip over some of the formal diagramming steps, such as steps 4, 8, and 11, in favor of creating a prototype, but some understanding of the task should be formalized and documented.

VI. Comments on Design Elements

Graphics, including charts, enable the presentation of information in a way that can clearly show the meaning of data and permits users to visualize relationships. The importance of using charts and graphs in communicating numeric data has been recognized for many years. Since the mid-1970s, computer graphics have been used to aid in management decision making. Graphics help managers "visualize" data, relationships, and variances. Common types of computer graphs and charts include time-series charts, bar and pie charts, scatter diagrams, maps, hierarchy charts, and flow charts. Managers, business analysts, and corporate staff use computer generated graphics in reports, presentations, performance tracking, scheduling, control, planning, modeling, and design. It is important to use graphics in the DSS user interface, especially in data displays. An end user tool like Excel has a wizard that helps create charts. Some of the charts available in Excel are shown in Figure 5.5.



Figure 5.5 Some Excel Chart types.

Let's summarize some guidelines for graphical displays. Communicate only one major message on each chart or screen. Use an action heading in appropriate sized fonts for screen and chart headings. A line chart is appropriate for displaying time-related information, but analysts need to be careful to use appropriate labels and to avoid adding dissimilar quantities. Bar charts are more appropriate for comparing individual data values. Pie charts help show how the whole breaks down into component parts. You should limit the number of components or pieces in a pie chart to five or fewer.

Color is often recommended as a means of enhancing a user-interface design. Appropriate use of color can enhance the aesthetics of an interface for most people. Color can call attention to extreme or exceptional data values, help users differentiate among items on a chart, and convey information quickly. For example, research indicates blue creates a sense of trust; green means "go" or "all clear;" red indicates danger. In general, the following guidelines related to the use of color are appropriate for DSS:

- 1. Do not allow color to be the only way your system conveys any information. Augment the use of color with other cues that can be used by people who cannot see the color difference. Include numerical values in addition to color codes, provide cross-hatching on top of color, or make sure that the colors you choose are perceived as substantially lighter or darker than each other.
- 2. Where your computer hardware and software permit, allow the user to customize an application's use of color. Changing colors can compensate for some people's color vision deficiencies. In some systems, colors that cover an area, such as a region on a map, can be replaced by monochrome patterns such as dots, stripes, and cross-hatching.
- 3. Use light pastel colors in screen designs. They create fewer annoying reflections than do dark colors. This is especially true in an office environment with fluorescent lights. As a result, try to use light colors to

cover large areas of the screen. Reserve darker colors for smaller "spot" usage.

Human-computer interaction is an important design issue. An interaction issue, unique to DSS and often not adequately considered, is the type and amount of guidance, called decisional guidance, that a DSS provides its users in the decision-making process (cf., Silver, 1991).

Decisional guidance provided by a system can be unintended or inadvertent. For example, users tend to select the first or last items from menus. Putting a frequently used capability in the middle of a menu may not be planned. Decisional guidance can also be planned and deliberate. Designers can intentionally build guidance mechanisms into a system after determining that a particular decision approach or process is better than what many users would come upon by chance. This type of planned, process guidance is distinct from the typical on-line help facility, which focuses on guidance in the mechanical aspects of operating the system. On-line help assumes that the user has already decided what to do, but does not know how to do it. Process guidance assume the user need direction in using the system.

Some DSS give the developer more of an opportunity to provide decisional guidance than do others. A DSS that does not provide for many discretionary user judgments during its use cannot benefit greatly from such a guidance facility. A DSS that lets users choose among several decision methods, several alternative models, several ways to cross tabulate a set of data, several forecasting techniques, or even alternative sequences of activities does provide an opportunity to provide decisional guidance.

VII. Guidelines for Dialog and User Interface Design

This section is based on Ben Shneiderman's (1992) research and writings. He has developed some underlying principles of design that he argues are applicable in most interactive systems. These underlying principles of interface design are derived heuristically from experience.

Strive for consistency. This principle is the most frequently violated one, and yet is the easiest one to correct and avoid. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout. Exceptions should be comprehensible and limited in number. A system should look, act, and feel the same throughout.

Provide frequent users shortcuts. As the frequency of use increase, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Frequent knowledgeable users appreciate abbreviations, special keys, hidden commands, and macro facilities. Shorter response times and faster display rates are other attractions for frequent users. A system must respond to the differing needs of its users.

Provide informative feedback. For every user action, there should be some system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial.

Design dialogs to create closure. Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the user the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions. The power of the dialog and commands should be appropriate to the capabilities of the users. Power is a measure of the amount of work accomplished by a given instruction to a system (cf., Galitz, 1985, p. 21).

Provide simple error recovery. Design the system so the user cannot make a serious error. If an error is made, the system should detect the error and offer simple, comprehensible mechanisms for handling the error. The user should not have to retype the entire command, but rather should need to repair only the faulty part. Erroneous commands should leave the System State unchanged, or the system should give instructions about restoring the state.

Permit easy reversal of actions. A user's actions should be reversible. This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

Support internal locus of control. Experienced users want to feel that they are in charge of the system and that the system responds to their actions. Surprising system actions, tedious sequences of data entries, incapacity or difficulty in obtaining necessary information and the inability to produce the action desired all build anxiety and dissatisfaction.

Reduce information load. Information load is a measure of the degree to which a user's memory is being used to process information on the display screens. It is a function of the task being performed, a person's familiarity with the task, and the design of the user interface itself. The limitation of human information processing in short-term memory requires that displays be kept simple and that sufficient training time be allotted for learning commands and sequences of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided. Designers can reduce information load by providing graphic rather than alphanumeric displays, format displays to correspond to users' immediate information requirements, use words that are easy to understand, provide simple dialogues (cf., Galitz, 1985, p. 21).

VIII. Factors Influencing User Interface Design Success

According to Larson (1982), user interface design success is influenced by 11 factors. Some of them overlap with Shneiderman's (1992). System factors include DSS execution time; the system's versatility; quality of help provided; adaptability; and uniformity of the commands and interface. Human factors include the: learning time for the DSS; ease of recall; errors made by end users; concentration required; fatigue from using the system; and the fun the user has while using the system. Larson elaborates on each of the factors that are determined, or at least affected, by the DSS user interface.

DSS execution time. How long does it take the end user to perform his or her task? Some fraction of task time is determined by actual DSS execution time. DSS execution time, while developers have some control over it by choosing software tools and hardware that are fast enough to provide the desired responsiveness, is not a user interface issue. Proper user-interface design can, however, minimize wasted time on the user's part.

Versatility. Can the system be used to perform a variety of tasks? The DSS must be versatile enough to accommodate the full range of tasks for which decision-makers will want to use it. Once a system becomes widely used, tasks related to its original purpose--but distinct from that purpose--will often arise. While it is reasonable for these new tasks to require new development work, it should be possible to incorporate them into the existing user-interface framework.

Quality of Help provided. Does the system provide help when the end user has trouble? On-line help facilities are becoming the norm for DSS. More and more development tools make it easy to incorporate on-line help into a system. Wherever possible, help should be context sensitive. The Help facility should recognize what the user is trying to do, or at least what screen the user is looking at, and provide help that is tailored as closely as possible to the current need. Expert systems have unique requirements in this area since they are often expected to explain to their users why they are asking for a particular data item or how they reached a particular conclusion.

Adaptability. Does the system adjust to the end user's level of competence as he or she becomes more experienced? Does it tailor itself to the habits and styles of different users? It may be difficult or impractical for the system to be truly self-tailoring in this sense. It is easier, and may be sufficient, to let the experience user select an "expert user" mode in which prompts are minimized. In a graphical user-interface environment, it is helpful to provide keyboard equivalents for commonly used mouse-and-menu commands, as some users prefer mice and others prefer keystrokes.

Uniformity of commands and interface. Are the commands of this system identical to equivalent commands of other systems? People can learn any idiosyncratic or esoteric interface it they want to. What is difficult is learning (and remembering) two, three, or more interfaces, be they esoteric or simple, and switching frequently among them. A DSS developer must be aware of other systems that the intended DSS users are familiar with. Also, using familiar terminology is a factor in making commands look uniform.

Learning time of the DSS. How long does it take a novice to learn the system? A design that provides for rapid learning must take into account what the user knows and how the user's mind fits that knowledge together.

Ease of recall. How easy is it for an end user to recall how to use the system after he or she has not used it for some time? This is a more important factor for DSS than for data

processing systems because managers often return to a DSS after a long interval of nonuse. Some systems, for example, are intended to help with decisions that recur predictably on an annual basis. Eleven months might elapse from a manager's last use of the system on one annual cycle until his or her first encounter with it on the next. A user interface that facilitates recall will reduce the time it takes to "get back up to speed" each year.

Errors. How many errors does the DSS user make, and how serious are those errors? The most serious errors lead to wrong decisions as a result of system misuse. Following closely behind are errors that corrupt a corporate database. Then come errors that bring down ("crash") the computer, followed finally by errors that waste the user's time but have no other bad effects. Fortunately, most user errors will be in the last category. Understanding the users' usual decision-making process can help minimize errors. Design the system to make errors less likely, or at least to make recovering form them as easy, quick and painless as possible.

Concentration required. How many things must an end user keep in mind while using the system? Most people have difficulty keeping more than six or seven active facts in mind at any one time. One way to reduce the memory load is to label screens and output with the parameters of the current scenario: "Profit Projections: 6% inflation, 10% sales growth, Model 47 shipments start 4/94..."

Fatigue. How quickly does the user tire while using the system? Physical fatigue--or, more seriously, repetitive strain injuries such as carpal tunnel syndrome--is seldom a factor with DSS because usage frequency is not high enough to lead to such problems. However, mental fatigue can occur. Minimize mental fatigue by keeping the necessary concentration required within the capabilities of the users and by asking for information in the sequence in which users would normally use it.

Fun. Does the end user enjoy using the DSS system? This does not mean "funny" error messages or jokes on the screen. Such "humor" grows stale quickly. It means you should keep users informed about what the system is doing, warn them of time-consuming operations, provide progress displays or reports as long operation is being carried out, and generally try to minimize frustrations that come from using an uncooperative system.

IX. Discussion and Commentary

The user interface is a major component of all Decision Support Systems. It facilitates communication between a system and its users. The user interface is a critical component of a DSS that creates unique design and development problems. This chapter briefly examined and evaluated the current status of our knowledge about user interface design for computer-aided Decision Support Systems. Managers are rarely comfortable with command line or simple menu interfaces. It appears that a sophisticated graphical interface can increase a manager's use of computerized decision aids and management support software.

In general, Decision Support Systems users seem to prefer easy to use, functional interfaces that use meaningful design conventions and standards. "Cute" user interfaces with funny graphics lose their appeal quickly and complex interfaces increase training costs and increase the burden to recall commands and conventions for the user.

A good DSS user interface can be built using a number of development environments. A Web-Based "thin client" interface can be as powerful and as easy to use as a more traditional client/server interface with a "thick client" installed on each user's computer. But, in any development environment extensive time and energy needs to be spent on designing and evaluating the interface. A good interface needs to be planned and designed.

Use a checklist for evaluating a prototype or proposed DSS interface. The user centered design guidelines in Appendix 5.1 provide a starting point for creating a checklist. Check the transparency of the DSS, the resiliency, the orientation and navigation, the impact on productivity, the impact on data integrity, and the control capabilities. Also, the design guidelines and factors influencing user interface design success discussed in this chapter should be incorporated in an evaluation checklist

The use of graphics and other visual information displays seems to appeal to many managers. GUI seems here to stay. Overall, if DSS builders improve user interfaces, then the usability of a new DSS and its value to decision-makers should increase.

X. Audit Questions

- 1. What software development products are primarily used in your company?
- 2. Does your company have user interface guidelines for systems?
- 3. Does your company involve potential users in the design and development of new applications?
- 4. Are users satisfied with the user interfaces of current DSS?

Questions for Review

- 1. What is a user interface?
- 2. What are the various user interface styles?
- 3. What steps should be followed in building a DSS User Interface?
- 4. What factors influence the success of a DSS user interface?
- 5. What is the ROMC approach? What is an example of each ROMC entity?
- 6. What are five important guidelines for designing a user interface? Why is each especially important?

Questions for Discussion

- 1. What user interface style do you prefer? Why? Is a GUI better for building DSS?
- 2. Why are graphics and charts important in DSS?

- 3. Examine a DSS user interface. Can you identify the ROMC entities?
- 4. What is meant by the statement "From the user's perspective, the dialog is the system"? Do you agree with this statement? Discuss.
- 5. User interface design guidelines standardize the "look and feel" of a user interface. Why is this important?
- 6. Should a DSS provide shortcuts for frequent users?

Internet Exercises

- 1. Search the Web to find sites on human-computer interaction and user interface design.
- 2. Visit Dragon Systems (<u>http://www.dragonsystems.com/</u>) and find out about Dragon Naturally speaking continuous speech recognition software.

XI. References

Bennett, J. L. (1986) "Tools for Building Advanced User Interfaces" IBM System Journal, No. 3/4.

Galitz, W.O. Handbook of Screen Format Design. (2nd edition) Wellesley, MA: QED Information Sciences, Inc., 1985.

Keen, Peter G. W. and Michael S. Scott Morton. Decision Support Systems: An Organizational Perspective. Reading, MA: Addison-Wesley, Inc.

Larson, J. End User Facilities in the Nineteen Eighties. IEEE Computer Society, Los Alamitos, CA., 1982.

Shneiderman, B. Designing the User Interface: Strategies for Effective Human-Computer Interaction. (2nd edition) Reading, MA: Addison-Wesley, 1992.

Silver, M.S. "Decisional Guidance for Computer-based Decision Support," MIS Quarterly, 15, no. 1 (March 1991), pp. 105-122.

Sprague, R.H. and E. D. Carlson. Building Effective Decision Support Systems. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1982.

Turban, E. Decision Support and Expert Systems: Management Support Systems (Third Edition). New York: Macmillan Publishing Co., 1993.

Whitten, J. L., Bentley, L. D. and Barlow, V. M. Systems Analysis and Design Methods (3rd ed.) Burr Ridge, IL: Irwin, 1994.

XII. DSS User Interface Web References

HCI Index at URL http://is.twi.tudelft.nl/hci/

Human Computer Interaction at URL http://hydra.bgsu.edu/HCI/

Human Computer Interaction Resources at URL http://www.ida.liu.se/labs/aslab/groups/um/hci

Web Style Guide at URL http://info.med.yale.edu/caim/manual/contents.html

Version 1.0 was completed 2/10/1997. Version 2.0 was completed 8/10/1999. Version 3.0 was completed January 5, 2000. Last updated September 7, 2000. Please request permission prior to quoting from this chapter.

XIII. Appendix 5.1 User Centered Design Guidelines

Please review and evaluate the following design guidelines. Why are they called user centered design guidelines? Do the guidelines make sense? Are they complete? Are they too detailed? Will they help a manager evaluate a DSS user interface? Or will they be of greater use to a DSS designer/builder?

The guidelines are based on a web document prepared at NASA called HCI Guidelines at URL http://groucho.gsfc.nasa.gov/Code 520/Code 522/Documents/HCI Guidelines.

A.1 Transparency: Create Low Profile Software.

- 1. Have the user interface design focus on the decision task, e.g., approving loan applications, monitoring key results metrics, allocating resources.
- 2. Have interface styles reflect the users point of view and conception of what is being done, not designer's point of view.
- Present only information relevant to the user's decision tasks. Do not present extraneous information on the screen.
 Use system capabilities because they enhance user task accomplishment, not because they can be used. For example, color or blinking text should be used appropriately.

A.2 Transparency: Design Intuitive Software.

- 1. Use abbreviations, mnemonics, codes, and acronyms based on normal language usage, specific job related terminology, or a known logic, do not use an arbitrary selection.
- 2. Design the DSS to take advantage of what the user already knows.
- 3. Make terminology for labeling, commands, messages, and prompts consistent with the user's frame of reference. A term should mean what a user thinks it means.
- 4. Design icons to directly represent the associated object or action.
- 5. Design the DSS to do what the user would naturally or naively guess it should do.

A.3 Transparency: Design Predictable Software.

- 1. Maintain visual consistency as well as action consistency.
- 2. Predictable software results from consistency.
- 3. Maintain consistency in the display, labeling terminology, system control and abbreviations.
- 4. Design the DSS so the user is able to predict how it will respond to actions.

B.1 Resiliency: Accept minor deviations in responses.

B.2 Resiliency: Anticipate User Actions and Needs.

C.1 Orientation and Navigation: Provide Orientation in the System.

- 1. Make sure the user knows: Where she is, What she can do there, How she can leave a page or the system.
- 2. Give each screen and window a descriptive title, placed in a consistent location.
- 3. Provide cues to identify the currently displayed page and the total number of pages in a multi-page display.
- 4. When appropriate, provide a system map to show the user where s/he is in the system.

C.2 Orientation and Navigation: Consider Possible User Actions.

- 1. Provide a general list of basic control options, always available, to serve as a home base or consistent starting point for control entries.
- 2. Make applicable menus and control options available to the user at all times.

C.3 Orientation and Navigation: Exiting.

- 1. Provide the user a means to log-off a DSS by a single action (e.g., menu option, command input).
- 2. Require a confirmation to exit without saving changes.

D.1 Productivity: Simplicity.

- 1. Provide an overall design and specific features that take job requirements and decision tasks into consideration and that supports job accomplishment.
- 2. Avoid the use of acronyms and abbreviations.
- 3. Require recognition rather than recall memory where possible.
- 4. Use units of measurement familiar to the user. Do not require the user to transform units of measurement.

D.2 Productivity: Screen Layout.

- 1. To support critical tasks, keep screen density as low as possible (for warning and emergency messages, preferably less than 25% of the screen space).
- 2. Maintain consistent display formatting within the system.
- 3. Use highlighting, blinking, reverse video, etc. to draw attention to specific screen elements -- but do not overuse such devices (Do not use blink coding for text that must be read).
- 4. Use colors for coding and emphasis, but use it sparingly (no more than 4 colors on a screen).
- 5. Display only task related information and place all data related to one task (operation/transaction) on a single screen.

D.3 Productivity: Feedback.

- 1. Highlight data, a message, a menu item, an icon, or other display structure as feedback to acknowledge that the user has selected the item.
- 2. Provide users with information about the current system status as it affects their work (for example, printing delays, inoperable peripherals, and processing delays due to system load).
- 3. When the completion of a command results in a consequence that is not visible to the user, provide a feedback message that describes the actions resulting from the command in simple, direct, positive language.
- 4. When the completion of a command results in a consequence that is easily perceptible to the user, do not provide additional feedback.
- 5. If a process is time-consuming and causes the screen and input devices to be locked, display and update a progress message.

E.1 Integrity: Dangerous Operations.

- 1. Maintain the integrity of DSS data.
- 2. Build protection around dangerous operations; permit the user to undo things that have been done.
- 3. Require users to confirm that they want to perform a critical, potentially hazardous, or potentially destructive command before execution.

E.2 Integrity: Help Function

1. Provide on-line Help -- summary information initially, with more detailed explanations available on request.

- 2. Permit the user to enter Help at any point and use a simple, standard action for the user to request Help.
- 3. Permit the user to browse Help topics.
- 4. Provide an easy means of returning to the task after accessing Help.

F.1 Control: System Control.

- 1. Make sure the user feels in control of a decision support session. The user should not feel controlled by the computer.
- 2. Give the user multiple means for doing things and let the user, not the computer, set the pace.
- 3. Provide for simple command language control of a DSS by advanced users.

F.2 Control: Entry.

- 1. Use command keystrokes where speed of command input is important. Make other dialog techniques available as appropriate.
- 2. Require the user to enter any particular data only once; have the system access that data if needed thereafter.
- 3. Design any data entry transactions and associated displays so that the user can stay with one method of entry and not have to shift to another.

F.3 Control: Feedback Messages.

- 1. Permit the user to request a more detailed explanation of feedback.
- $2. \quad {\rm Use \ neutral \ wording \ in \ feedback \ messages \ -- \ do \ not \ make \ them \ funny \ or \ negative.}$
- 3. Make messages brief. Do NOT use the word "error".
- 4. Design the DSS so users are unlikely to make "errors".