

# : *What rules should guide DSS user interface design?*

by Dan Power

Editor, DSSResources.com

People must interact with a computerized decision support system to use it. The design of the software interface between the application and the user largely determines whether a given DSS will be used and whether it will be used effectively. Research and past experience suggests "rules" or "guidelines" for building the interface. The software interface should be the central design focus, but it is important to examine broadly the total user experience. The following is a heavily prescriptive, modestly technical discussion of how to develop an interface for a wide variety of computerized decision support systems.

In general, my focus for user interface design is on what a DSS designer should know and do. We must be cautious with a nebulous concept like user experience. Assessing user experience attempts to evaluate and describe the overall experience and satisfaction a person has with a system. The phrase reminds me of the old light bulb joke: "How many Californians are needed to change a light bulb? Answer: "Ten, one person to change the bulb and 9 to share the experience." The following guidelines focus on the role of the person "changing the light bulb."

Ben Shneiderman's (1992) and Jakob Nielsen's research and writings have had a major influence on my thinking about user interface design "rules". Shneiderman has a guidelines and theory approach and Nielsen emphasizes "usability heuristics", both are very prescriptive in their user interface (UI) design approaches.

Shneiderman developed some underlying principles of design that he argues are applicable in most interactive systems. My prior work on user interface design (cf., Power, 2002) started with Shneiderman's work. The following UI design rules are derived heuristically from personal experience, and from writings of Shneiderman (1992, 2005), Gallitz (1985), Talin (1998) and Nielsen.

The following 10 UI design rules are in priority order, starting with the most important rule:

## : *What rules should guide DSS user interface design?*

1) Strive for consistency. This rule 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.

2) 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, and provide simple dialogues (cf., Galitz, 1985, p. 21). In general, information should appear in a natural and logical order. The system should "speak the users' language".

3) Create an aesthetic and minimalist interface design. The interface/display should be pleasing and in many cases the layout should be balanced and proportional. Use familiar interface components when possible. If the purpose of a screen object is not quickly apparent, eliminate it. See rule #2.

4) Provide informative feedback about system status. 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. The DSS should always keep users informed about what is going on!

5) Design interactions to create closure. Sequences of actions/steps should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions should give the user the satisfaction of accomplishment. The power of a dialog/interaction should be appropriate to the capabilities of the users.

6) Anticipate and avoid errors. 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,

## : *What rules should guide DSS user interface design?*

comprehensible mechanisms for handling the error. The user should not have to reenter data or commands, but rather should need to repair only the faulty part. Erroneous actions/commands should leave the system state unchanged, or the system should give instructions about restoring the state. Good design should prevent an error from occurring in the first place. For error-prone situations, the system should present users with a confirmation option before they commit to an action.

7) Permit easy reversal of user 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 DSS options. The unit of reversibility may be a single action, a data entry, or a complete group of actions. Support undo and redo.

8) Support internal locus of control of users. 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, difficulty in obtaining necessary information and a user's inability to produce a desired action create anxiety and dissatisfaction.

9) Provide "accelerators" for frequent users. As the frequency of DSS use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Knowledgeable users often appreciate abbreviations, special keys, hidden commands, and automation facilities. Shorter response times and faster display rates are also important for frequent users. A system must respond to the differing needs of its users. According to Nielsen, an effective "accelerator" speeds up the interaction for the expert user. "Instruction Power" is a measure of the amount of work accomplished by a given instruction to a system (cf., Galitz, 1985, p. 21). Make sure commands and actions have appropriate "instruction power" given the needs and experience of the user.

10) Provide "help" capabilities and appropriate DSS documentation. Although it is best if a DSS can be used without documentation, providing help and documentation remains desirable. Online help information should be easy to search, navigate and understand.

These ten rules can be broken by a knowledgeable DSS designer, but for most of us we should plan

## : *What rules should guide DSS user interface design?*

to implement them in our decision support projects. An effective interface makes the system easier to use, increases use of the system, reduces training costs, and increases satisfaction of users. The more intuitive the user interface, the easier it is to use.

### References

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

Lewis, C., and J. Rieman, Task-Centered User Interface Design: A Practical Introduction, 1994, <http://www.hcibib.org/tcuid/> .

Nielsen, J., "Ten Usability Heuristics,"  
URL [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)

Power, D., Decision Support Systems: Concepts and Resources for Managers, Greenwood/Quorum Books, 2002.

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

Shneiderman, B. and C. Plaisant, Designing the User Interface, Boston: Pearson/Addison-Wesley, 2005.

Talin, "A Summary of Principles for User-Interface Design," Last updated: Friday, August 14, 1998, [http://www.sylvantech.com/~talin/projects/ui\\_design.html](http://www.sylvantech.com/~talin/projects/ui_design.html)

Wikipedia, User interface design, [http://en.wikipedia.org/wiki/User\\_interface\\_design](http://en.wikipedia.org/wiki/User_interface_design)

## : *What rules should guide DSS user interface design?*

Some Web links for User Interface Design and Usability Testing:

Apple Computer Developer's Guide

<http://developer.apple.com/documentation/index.html>

Efficacy of User Interface Design

<http://www.usernomics.com/user-interface-design.html>

Human-Computer Interaction Lab (HCIL) at the University of Maryland

<http://www.cs.umd.edu/projects/hcil/>

IBM's Ease Of Use web

<http://www-03.ibm.com/easy/page/558>

QUIS: The Questionnaire for User Interaction Satisfaction

<http://www.lap.umd.edu/QUIS/index.html>

Webcredible

<http://www.webcredible.co.uk/>

Author: Daniel Power

Last update: 2011-02-14 06:52