MODULE 3: DESIGNS OF HUMAN-COMPUTER INTERACTION UNIT

1: DESIGN GUIDELINES, RULES AND PRINCIPLES

Table of contents

1.0 Introduction
2.0 Objectives
3.0 Main Content
3.1 Design guidelines
3.1.1 Selection of design guidelines
3.1.2 Expertise experience versus guidelines
3.1. 3 Monitoring guidelines and prototype testing
3.1.4 Translation of selected guidelines into design rules
3.2 Design rules
3.2.1 Types of design rules
3.2.2 Documentation, implementation and evaluation of design rules
3.2. 3 Using design rules
3.2.4 The Shneiderman's 8 Golden Rules
3. 3 Design principles
3. 3.1 Norman's 7 design principles
3. 3.2 Design principles formulated to support usability:
4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 Further Reading/References

1.0 INTRODUCTION

This unit offers guidelines for design of user interface software in six functional areas: data entry, data display, sequence control, user guidance, data transmission, and data protection.

The guidelines are proposed here as a potential tool for designers of user interface software. Guidelines can help establish rules for coordinating individual design contributions; can help to make design decisions just once rather than leaving them to be made over and over again by individual designers. It can also help to define detailed design requirements and to evaluate user interface software in comparison with those requirements.

The design of user interface software will often involve a considerable investment of time and effort. Design guidelines can help ensure the value of that investment.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Select design guidelines
- Know how to monitor guidelines and carry out prototype tests
- Understand the concept of translating selected guidelines into design rules

- Know the importance of documenting design rules
- Explain various types of design rules
- Know how to use design rules
- Learn some design principles

3.0 MAIN CONTENT

3.1 DESIGN GUIDELINES

Guidelines are more suggestive and general. There are two types of guidelines, they are:

i. Abstract guidelines or principles that are applicable during early design life cycle activities

ii. Detailed guidelines otherwise called style guides that are applicable during the later system life cycle activities

Understanding justification for guidelines aids in resolving conflicts

3.1.1 SELECTION OF DESIGN GUIDELINES

Not all of the guidelines can be applied in designing any particular system. For any particular system application, some of the guidelines will be relevant and some will not. Design guidelines must be generally worded so that they might apply to many different system applications. Thus generally-worded guidelines must be translated into specific design rules before they can actually be applied.

The process of selecting relevant guidelines for application and translating them into specific design rules is referred to here as "tailoring". Who will do this guidelines tailoring? It should be the joint responsibility of system analysts and human factors specialists assessing design requirements, of software designers assessing feasibility, and of their managers. It may also be helpful to include representatives of the intended system users in this process, to ensure that proposed design features will meet operational requirements.

Once all relevant guidelines have been identified, a designer must review them and decide which ones actually to apply.

There are two reasons why a designer might not wish to apply all relevant guidelines.

First, for any given application, some guidelines may conflict, and the designer must therefore choose which are more important.

Second, budgetary and time restrictions may force the designer to apply only the most important guidelines -- those that promise to have the greatest effect on system usability.

3.1.2 EXPERTISE EXPERIENCE VERSUS GUIDELINES

Guidelines cannot take the place of expertise experience.

An experienced designer, one skilled in the art, might do well without any guidelines. An inexperienced designer might do poorly even with guidelines. Few designers will find time to read an entire book of guidelines. If they do, they will find it difficult to digest and remember all of the material. If guidelines and/or the rules derived from guidelines are to be helpful, they must be kept continually available for ready reference.

Guidelines cannot take the place of expert design consultants, or at least not entirely. A design expert will know more about a specific topic than can be presented in the guidelines. An expert will know what questions to ask, as well as many of the answers. An expert will know how to adapt generally-stated guidelines to the specific needs of a particular system design application. An expert will know how to trade off the competing demands of different guidelines, in terms of operational requirements.

3.1. 3 MONITORING GUIDELINES AND PROTOTYPE TESTING

For maximum effectiveness, guideline tailoring must take place early in the design process before any actual design of user interface software. In order to tailor guidelines, designers must have a thorough understanding of task requirements and user characteristics. Thus task analysis is a necessary prerequisite of guidelines tailoring.

The result of guidelines application will be a design for user interface software that may incorporate many good recommendations. However, even the most careful design will require testing with actual users in order to confirm the value of good features and discover what bad features may have been overlooked. Thus prototype testing must follow initial design, followed in turn by possible redesign and operational testing.

Indeed, testing is so essential for ensuring good design that some experts advocate early creation of an operational prototype to evaluate interface design concepts interactively with users, with iterative design changes to discover what works best. But prototyping is no substitute for careful design. Prototyping will allow rapid change in a proposed interface; however, unless the initial design is reasonably good, prototyping may not produce a usable final design.

Considering the system development process overall, guidelines application will not necessarily save work in user interface design, and in fact may entail extra work, at least in the initial stage of establishing design rules. But guidelines application should help produce a better user interface. Because guidelines are based on what is known about good design, the resulting user interface is more likely to be usable. Certainly the common application of design rules by all designers working on a system should result in more consistent user interface design. And the single objective on which experts agree is design consistency.

3.1.4 TRANSLATION OF SELECTED GUIDELINES INTO DESIGN RULES

Because guidelines are intended for use on a variety of systems, they are worded in general terms. Before a guideline can actually be applied it must be translated into specific design rules. For instance, a guideline which states that displays should be consistently formatted might be translated into design rules that specify where various display features should appear, such as the display title, prompts and other user guidance, error messages, command entries, etc.

Any guideline can have different possible translations. A guideline which states that each display should be uniquely identified could be translated into a design rule that display titles will be bolded and centered in the top line of the display. Or it could be translated into a design rule that display titles will be capitalized in the upper left corner of the display.

What would happen if guidelines were not translated into design rules, but instead were given directly to interface designers? If designers do not decide as a group what design rules will be used, then each designer will decide separately in the course of applying guidelines. The result will surely be an inconsistent design.

3.2 DESIGN RULES

3.2.1 Types of design rules Rules based on principles:

These comprise abstract design rules, rules based on low authority, and those based on high generality Rules based on standards

These are specific design rules from high authority but with limited application Rules derived from guidelines

These are rules of lower authority but of more general application

3.2.2 DOCUMENTATION, IMPLEMENTATION AND EVALUATION OF DESIGN RULES

After design rules have been specified for each selected guideline, those rules should be documented for reference by software designers and others involved in system development. Documentation of agreed rules, subject to periodic review and revision as necessary, will help coordinate the design process. Documented rules can then be applied consistently for a given application. With appropriate modifications, rules adopted for one application might later be used for other applications.

In the course of design, it may be determined that a particular design rule cannot be used. Therefore, some means must be provided to deal with exceptions. If a design rule is not appropriate for one particular display, then an exception can be made by whoever has been appointed to make such decisions. But if a design rule cannot be implemented at all, perhaps due to other design constraints, then all designers for that particular system must be notified, and perhaps another design rule must be substituted.

Finally, after the design is complete, it must be evaluated against the original design requirements to ensure that all design rules have indeed been followed. To help in the exception process and in the evaluation process, it may be useful to assign different weights to the various rules, indicating which are more important than others. Such weighting will help resolve the trade-offs that are an inevitable part of the design process.

3.2. 3 USING DESIGN RULES

Since design rules suggest how to increase usability, they may differ in generality and authority, therefore there exist various standards that guarantee uniformity of application. Some of those existing standards are:

- The standards set by national or international bodies to ensure compliance by a large community of
  Designers. These standards require sound underlying theory
  particularly on this slowly changing technology.
- Hardware standards: These are more common than software standards. They are of high authority and low level of detail
- ISO 9241 standards that define usability as the effectiveness, the efficiency and the satisfaction with which users accomplish tasks
- "Broad brush" design rules
- Useful check list forgood design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen's 10 Heuristics (see Chapter 9)
  - Shneiderman's 8 Golden Rules
  - Norman's 7 Principles

There are Golden rules and heuristics governing designs
These are regarded as "Broad brush" design rules that provide a useful check list for good design. A better design is achieved using these than using nothing!
The different collections include: the Nielsen's 10 Heuristics rules, the Shneiderman's 8 Golden Rules and the Norman's 7 Principles.

3.2.4 The Shneiderman's 8 Golden Design Rules

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialogs to yieldclosure*
5. *Offer error prevention and simple errorhandling*
6. *Permit easy reversal ofactions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*

3.3 DESIGN PRINCIPLES

3. 3.1 the Norman's 7Design Principles

1. Use both knowledge in the world and knowledge in the head.
2. Simplify the structure of tasks.
3. Make things visible: bridge the gulfs of Execution and Evaluation.
4. Get the mappings right.
5. Exploit the power of constraints, both natural and artificial.
6. Design for error.
7. When all else fails, standardize.

3. 3.2 DESIGN PRINCIPLES FORMULATED TO SUPPORT USABILITY :

Principle of Learnability: This is the ease with which new users can begin effective interaction and achieve maximal performance
Principle of Flexibility: These are the multiplicity of ways the user and system exchange information Principle of Robustness: This is the level of support provided the user in determining successful achievement and assessment of goal-directed behavior

The Principles of learnability are broken down into:
Predictability: This is determining effect of future actions based on past interaction history and its operation visibility
Synthesizability: This is assessing the effect of past actions, its immediate and its eventual honesty Familiarity: This is how prior knowledge applies to new system and how easy one can guess its affordance Generalizability: This is extending specific interaction knowledge to new situations Consistency: This concerns the likeness in input and output behavior arising from similar situations or task objectives

Principles of flexibility comprise:
Dialogue initiative: This is the freedom from system imposed constraints on input dialogue and it compares the system against the user pre-emptiness.

Multithreading: This is expressing the ability of the system to support user interaction for more than one task at a time. It also looks at the concurrent and interleaving multimodality.

Task migratability: This is passing responsibility for task execution between user and system Substitutivity: This allows equivalent values of input and output to be substituted for each other. It compares representation multiplicity and equal opportunity

Customizability: This is the modifiability and adaptability of the user interface by user or the modifiability and adaptivity of the user interface by the system.

Principles of robustness are made up of:

Observability: This is the ability of the user to evaluate the internal state of the system from its perceivable representation. It considers the browsability, the defaults, the reachability, the persistence, and the operation visibility.

Recoverability: This concerns the ability of the user to take corrective action once an error has been recognized. It looks at the reachability, the forward and backward recovery and the commensurate effort. Responsiveness: This is how the user perceives the rate of communication with the system and how stable is the response.

Task conformance: This explains the degree to which system services support all of the user's tasks, the task completeness and its adequacy.

4.0 Conclusion

The goal of interaction design is to design for maximum usability

Design rules comprise the principles of usability which look at the general understanding of the design, the standards and guidelines which set the direction for design, and the design patterns that capture and reuse design knowledge.

In designing computer-based information systems, special attention must be given to software supporting the user interface.

5.0 Summary

Guidelines are more suggestive and general. There are two types of guidelines; Abstract guidelines and detailed guidelines.

Understanding justification for guidelines aids in resolving conflicts.

Before a guideline can be applied, it must be translated into specific design rules. Those rules should be documented for reference by software designers and others involved in system development.

There are Golden rules and heuristics governing designs

For maximum effectiveness, guideline monitoring must take place early in the design process before any actual design of user interface software.

Guidelines cannot take the place of expertise experience.

There should be early creation of an operational prototype to evaluate interface design concepts interactively with users.

There are certain design principles of learnability, flexibility, and robustness that are formulated to support usability.

6.0 Tutor Marked Assignment

1. Mention those responsible for selecting relevant guidelines for application and translation into design rules. Indicate specific area of responsibility for each professional.
2. Why is it necessary to translate selected guidelines into design rules?
3. What are the advantages derivable from documenting design rules, why is the evaluation of the design necessary?
4. When is it most appropriate to monitor design guidelines and carry out prototype testing and why?
5. Distinguish between rules based on principles, those based on standards and those derived from guidelines.
6. What are those design principles formulated to support usability?
7. Mention any 6 of the Shneiderman's 8 Golden rules that govern interactive designs.

7.0 Further Readings / References

- C. Brown: Human-Computer Interface design guidelines. Ablex, 1989.
- W. Galitz: Handbook of screen format design. QED, 1989.
- C. Gram, G. Cockton (eds.): Design principles for interactive software. Capman & Hall, 1996.
- D. Hix, R. Hartson: Developing user interfaces. Wiley, 1993.
- ISO 9241 (Part 10: Dialogue principles, Part 12: Presentation of information, Part 14: Menu dialogues, Part 15: Command dialogues, Part 16: Direct manipulation dialogues, Part 17: Form fill-in dialogues)
- D. Mayhew: Principles and guidelines in software user interface design. Prentice, 1992.