

Designing the User Interface

Strategies for Effective

*Human-Computer
Interaction*



Third Edition

Ben Shneiderman

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

Library of Congress Cataloging-in-Publication Data

Shneiderman, Ben.

Designing the user interface : strategies for effective human
-computer-interaction / Ben Shneiderman. -- 3rd ed.

p. cm.

Includes bibliographical references and index.

ISBN 0-201-69497-2

1. Human-computer interaction. 2. User interfaces (Computer
systems) I. Title.

QA76.9.H85S54 1998

004'.01'9--dc21

96-37974

CIP

Access the latest information about Addison-Wesley titles from our World Wide
Web site: <http://www.awl.com/cseng>

Chapter opener illustrations from art provided by Mark Kostabi. Reproduced
with permission.

Cover art © Boris Lyubner/SIS

Copyright © 1998 by Addison Wesley Longman, Inc.

Reprinted with corrections, March 1998.

All rights reserved. No part of this publication may be reproduced, stored in a
retrieval system, or transmitted, in any form or by any means, electronic,
mechanical, photocopying, recording, or otherwise, without the prior written
permission of the publisher. Printed in the United States of America.

6 7 8 9 10-MA-01009998

developers, in spite of limited success thus far. Natural-language interaction usually provides little context for issuing the next command, frequently requires *clarification dialog*, and may be slower and more cumbersome than the alternatives. Still, where users are knowledgeable about a task domain whose scope is limited and where intermittent use inhibits command-language training, there exist opportunities for natural-language interfaces (discussed at the end of Chapter 8).

Blending several interaction styles may be appropriate when the required tasks and users are diverse. Commands can lead the user to a form fillin where data entry is required, or menus can be used to control a direct-manipulation environment when a suitable visualization of actions cannot be found.

2.5 Principle 2: Use the Eight Golden Rules of Interface Design

Later chapters cover constructive guidance for design of direct manipulation, menu selection, command languages, and so on. This section presents underlying principles of design that are applicable in most interactive systems. These underlying principles of interface design, derived heuristically from experience, should be validated and refined.

1. *Strive for consistency.* This rule is the most frequently violated one, but following it can be tricky because there are many forms of consistency. Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent color, layout, capitalization, fonts, and so on should be employed throughout. Exceptions, such as no echoing of passwords or confirmation of the delete command, should be comprehensible and limited in number.
2. *Enable frequent users to use shortcuts.* As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, special keys, hidden commands, and macro facilities are appreciated by frequent knowledgeable users. Short response times and fast display rates are other attractions for frequent users.
3. *Offer informative feedback.* For every user action, there should be system feedback. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial. Visual presentation of the objects of interest provides a con-

venient environment for showing changes explicitly (see discussion of direct manipulation in Chapter 6).

4. *Design dialogs to yield 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 operators 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.
5. *Offer error prevention and simple error handling.* As much as possible, design the system such that users cannot make a serious error; for example, prefer menu selection to form fillin and do not allow alphabetic characters in numeric entry fields. If users make an error, the system should detect the error and offer simple, constructive, and specific instructions for recovery. For example, users should not have to retype an entire command, but rather should need to repair only the faulty part. Erroneous actions should leave the system state unchanged, or the system should give instructions about restoring the state.
6. *Permit easy reversal of actions.* As much as possible, actions should be reversible. This feature relieves anxiety, since the user knows that errors can be undone, thus encouraging exploration of unfamiliar options. The units of reversibility may be a single action, a data-entry task, or a complete group of actions such as entry of a name and address block.
7. *Support internal locus of control.* Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Surprising system actions, tedious sequences of data entries, inability or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction. Gaines (1981) captured part of this principle with his rule *avoid acausality* and his encouragement to make users the *initiators* of actions rather than the *responders* to actions.
8. *Reduce short-term memory load.* The limitation of human information processing in short-term memory (the rule of thumb is that humans can remember "seven-plus or minus-two chunks" of information) requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided.

These underlying principles must be interpreted, refined, and extended for each environment. The principles presented in the ensuing sections