

SUPPORTING NOVICE APPLICATION USERS IN LEARNING BY TRIAL
AND ERROR AND READING HELP

OSCAR DANIEL ANDRADE

Department of Computer Science

APPROVED:

David G. Novick, Ph.D., Chair

Nigel G. Ward, Ph.D.

Christian A. Meissner, Ph.D.

Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

Copyright

by

Oscar D. Andrade

2009

to my

MOTHER
GRANDPARENTS
and SISTER

with love and gratitude

PREVIEW

SUPPORTING NOVICE APPLICATION USERS IN LEARNING BY TRIAL
AND ERROR AND READING HELP

by

OSCAR DANIEL ANDRADE, BS

THESIS

Presented to the Faculty of the Graduate School of
The University of Texas at El Paso
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Department of Computer Science
THE UNIVERSITY OF TEXAS AT EL PASO
December 2009

UMI Number: 1473849

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.

UMI[®]

Dissertation Publishing

UMI 1473849

Copyright 2010 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.

ProQuest[®]

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Acknowledgements

In no particular order, my sincerest thanks go to: Guadalupe Rivera, Félix Arellano Flores, Roberto Fernandez, Gerardo Ortiz, Javier Ortiz, Ben Molina, Dr. Walter Fisher, Dr. Novick for spotting me from the crowd, Kathleen Raphael, Drs. Carl Jackson and Osvaldo Mendez, Drs. Ceberio and Longpré, Coach Johnson, Jorge Bañuelos, Dr. Karen Ward, Dr. Nigel Ward for showing me my calling, Dr. Rodrigo Romero, Drs. Santiago Ibarreche and John Bretting, David Batten, Dr. Josefina Lujan, Dr. Bill Farnsworth, Edgar and Eduardo Morales, Rafael Escalante, Shreyas Karkhedkar, Irbis Gallegos, Mark Williamson, Aramis Martinez and Nathaniel Bean, my research associate and co-author. Much thanks to former classmates, UTEP co-workers and to Interactive Systems Group colleagues; it is a personal journey but others walk along the same path.

More than anything, thank you Linda. The message lives on.

Abstract

This thesis examines three related topics in analyzing user preferences for troubleshooting application usability problems and the kinds of issues that influence such preferences. The goal of this thesis is to propose (1) ways in which users can be supported to learn to use applications and (2) a model to rewrite help such that users are able to adapt its contents dynamically.

The consensus of documentation research is that users rarely use help, usually preferring to muddle through. To increase use of help, tutorials for novice users could be changed from guided presentations toward using the application's help system. To determine whether this approach would increase users' use of help when they encountered problems with an application, I developed an alternative, help-based tutorial introduction to Microsoft Publisher. I compared the behaviors of users introduced to Publisher with the help-based tutorial with the behaviors of users who learned from a traditional tutorial. A balanced study of 22 novice users of Publisher suggests that using a help-based tutorial leads to significantly greater use of help when users encounter problems. However, the data also suggest that the increased use of help may not lead to more effective task performance.

A post-experiment survey showed that the subjects did not find help useful; and while they still used help at least occasionally, they used other problem-solving strategies. I analyzed recordings of the subjects, to identify (1) transition patterns among problem-solving approaches, and (2) the frequency of these transitions. The analysis indicates that people switch frequently between consulting help and exploring the interface. Switching between problem-solving approaches appears to be an effective way of succeeding in tasks. Applications and their help systems can be better designed to support users who switch between help and non-help approaches to solving problems.

Another common complaint about help is that the contents are pitched at the wrong level of technical detail for users, but the "right" level differs among them. Building on a prior definition of the space of possible expressions of documentation in terms of task, application, and user experience, I explore what it means to express help at different levels and conclude by proposing an interface users could adapt dynamically to find help expressed at their own "right" level.

Table of Contents

Acknowledgements.....	v
Abstract.....	vi
Table of Contents.....	vii
List of Tables	ix
List of Figures.....	x
Chapter 1: Introduction.....	1
1.1 Help-Based Tutorials.....	1
1.2 Troubleshooting Strategy Preferences	4
1.3 Tuning Help to a User's Level of Experience	5
Chapter 2: Background.....	6
2.1 Approaches to Tutorials.....	6
2.1.1 An Adapted Microsoft Tutorial	6
2.1.2 A Help-Based Tutorial.....	7
2.2 Troubleshooting Preferences	9
2.3 Multi-Layer and Multi-Dimensional Interfaces.....	11
Chapter 3: Methodology	16
3.1 Subjects.....	16
3.2 Experimental Design	17
3.3 Post-Session Survey.....	18
3.4 Data Coding.....	19
3.5 Assessing Troubleshooting Strategy Preferences	20
3.6 The Tau (Task, Application and User) Model.....	23
Chapter 4: Results.....	25
4.1 Use of Help Across Subjects	25
4.2 Troubleshooting Approaches.....	27
4.2.1 Transition Patterns	27
4.2.2 Possible Causes.....	28
4.2.3 Choice of Approaches.....	29

Chapter 5: Discussion	31
5.1 Participant's Characteristics	31
5.2 Participant's Use of the Tutorials	31
5.3 Analysis of Results for Hypothesis 2.....	32
5.4 Qualitative Findings.....	35
5.5 Initial Approaches.....	37
5.6 Transition Approaches.....	39
5.7 Strategy	40
Chapter 6: Conclusion	43
6.1 Limitations.....	44
6.2 Supporting Problem-Solving	46
6.3 Rewriting Help for Different Tau Levels	50
References.....	57
Appendix A: Tasks	59
Appendix B: Post-Session Survey	62
Curriculum Vita	63

List of Tables

Table 4.1: Results for hypothesis 1 (overall $n = 22$).....	25
Table 4.2: Results for hypothesis 2 (overall $n = 22$).....	26
Table 4.3: Initial episode successes.	30
Table 4.4: Transitions between approaches.....	30
Table 5.1: Mean task performance, by condition and task.....	32
Table 5.2: Mean seconds using help, by condition and task.	33
Table 5.3: Mean accesses of help, by condition and task.....	33

PREVIEW

List of Figures

Figure 2.1: Excerpt of an adapted traditional tutorial.....	7
Figure 2.2: Instructions on accessing help.....	8
Figure 2.3: Excerpt of help-based tutorial.....	9
Figure 2.4a: Interface with low feature complexity.....	12
Figure 2.4b: Moving the slider up increases feature complexity.....	13
Figure 2.5: A tutorial with layered explanations.....	13
Figure 2.6: Why-what-how dimensions.....	14
Figure 3.1: The Tau dimensions.....	24
Figure 5.1: Mean seconds using help, by condition and task.....	35
Figure 5.2: Mean accesses of help, by condition and task.....	35
Figure 6.1: Explanations for all levels of experience in the same sentence.....	51
Figure 6.2: A novice-level explanation on how to copy and paste multiple items with the clipboard.....	52
Figure 6.3: An intermediate-level explanation for the same topic.....	53
Figure 6.4: An expert-level explanation for the same topic.....	53
Figure 6.5: A proposal for a multi-layer help system.....	54
Figure A.1: “Arrow sign” reference design for task 1.....	59
Figure A.2: “Business card” reference design for task 2.....	59
Figure A.3: “Golden apple advertisement” reference design for task 3.....	60
Figure A.4: “Raise petition” reference design for task 4.....	61

Chapter 1: Introduction

Novice users of applications encounter various kinds of issues when learning how to do a task or sets of tasks, for which it would be helpful to use the help system included with these applications because help is a readily available guide to the features they need to use. Yet, application novices also encounter issues with the documentation and even when they use it frequently it does not necessarily improve their performance. Moreover, they seldom rely solely on reading help to troubleshoot problems with the applications they are using and resort to other approaches to recreate or simulate the desired outcome, often exploring the interface when it proves to be a helpful strategy.

In response to such preferences, many improvements can be made to provide support for users who have a mixed learning strategy so they can leverage a hands-on approach with the information that is readily available in a help system. The help system should also be redesigned to address the dimensions of issues in which usability problems exist, and should also be pitched at different levels to accommodate varying levels of understanding of users based on their experience with such dimensions.

Thus, this thesis presents three related, but partly independent topics that cover (1) motivating users to read help systems to learn more effectively about applications and their features and whether they do better at a particular task when using help to do a task, (2) understand the kinds of issues that cause users to give up on help, the choice of other troubleshooting approaches taken by users and possible ways to support mixed-strategy learning, and (3) rewriting help contents to reflect the broad categories of usability issues and provide a multi-level representation that users can adapt dynamically such that the level of technical detail matches each user's level of experience across such categories.

1.1 HELP-BASED TUTORIALS

Users prefer to explore the interface of an application to find the features they need to do a task instead of searching within the help system to learn about the application even though help systems are carefully crafted to provide thorough guidance to users, which should save them time and effort when learning about an application in the short term, and would act as a handy reference in the long term that would also save them the effort of recalling what they took a lot of time to learn by exploring. However,

users often complain that they do not find help systems to be useful and resort to other troubleshooting strategies, while still looking back at help in some cases. Because users do not have preference for a single troubleshooting strategy, alternatives for supporting multiple approaches can be suggested, including redesigning help to support users – particularly at different levels of individual experience with the task, application and computing environment.

Thousands of technical writers spend years producing manuals and help systems intended to help people use computer applications more effectively. Despite these enormous efforts, users of computer applications routinely muddle through with trial-and-error methods rather than take advantage of the help available to them. It may be possible to redesign the way in which users learn about computer applications so that they are more familiar with and more likely to use help systems.

Although recent research has helped to quantify low rates of use of help systems, this problem has been of concern for many years, notably as the impetus of the trend toward minimal manuals [Carroll 1990, 1998]. But even with wide acceptance of minimalism by the documentation community, use of documentation in general, including help systems, remains surprisingly low. While some studies have reported relatively higher uses of computer documentation and help systems (e.g., [Smart, Whitting and De Tienne 1998, 2001]), the evidence on balance is that most users, when faced with problems in a computer application, generally do not use documentation or help systems ([Novick and Ward, 2006a]; see also [Ceaparu, Lazar, Bessiere, Robinson and Shneiderman 2004], [Mendoza and Novick 2005]). Observations of people in work settings suggest that reluctance to use help is even more widespread than indicated by users' self-reports. While some studies relying on interviews (e.g., [Smart, Whitting and De Tienne 2001], [Novick and Ward 2006a]) reported that people used online help in roughly 20-35% of cases where they encountered a problem with a computer application, studies based on direct observation and participative evaluation indicated that the people used online help in fewer than 10% of such cases (see [Ceaparu, Lazar, Bessiere, Robinson and Shneiderman 2004], [Mendoza and Novick 2005], [Novick, Elizalde and Bean 2007]). Rather than use help, people tend to use less effective methods and trial-and-error techniques [Novick, Elizalde and Bean 2007].

Frustrations with computer applications arise in part because people are overwhelmed by the large number of functions these applications typically provide [Baecker, Booth, Jovicic and Moore

2000]. While people may read a tutorial when first encountering a new application, they still have difficulty finding ways of doing things in the application. In the authors' own experience, having any way of doing something tends to fossilize into the one way of doing that thing. For example, a person who crops images in Microsoft Word using the "Format Picture" dialog box may never learn, unless told by a colleague, that this can be done through direct manipulation via an icon on the "Picture" toolbar. Similarly, people will rely on repetitive cut-and-paste methods when, had they known about it, they could have used a mail-merge function [Novick, Elizalde and Bean 2007]. It appears that people who know of one way to do something may not suspect that there is another, easier way to do the same thing. And even when people know that there must be better way, they sometimes do not believe that they would be able to find and use it [Novick, Elizalde and Bean 2007]. This problem becomes more serious as our increasingly complex applications gain multiple ways of accomplishing a task. The users of the applications too rarely go into the help system to find better ways of doing things.

How, then, can developers of computer applications guide the users of these applications toward online help? Despite acceptance of minimalism among technical communicators, rates of use of help remain low among users of computers. And the complex, redundant functionality of computer applications engenders reliance on muddling through. If, however, users of computer applications, when first encountering an application, would learn to use the application's help system routinely, then these users might access help more frequently when experiencing frustration or encountering a new problem [Novick, Andrade, Bean and Elizalde 2008].

In this thesis, I explore the idea that users of a computer application might use online help more often if their introduction to the application provided significant experience in finding application functions through the application's online help system. I present two contrasting approaches to application tutorials — traditional and help-based — and describe an empirical study comparing the effectiveness of the tutorials. I report that a help-based tutorial can increase use of help but does not necessarily lead to more effective task performance because the subjects did not find help useful in many cases and, as a supplementary approach, solved application issues through trial-and-error.