Supporting Novice Application Users in Learning by Trial and Error and Using Help

Oscar Daniel Andrade
University of Texas at El Paso, oscarito.and@gmail.com

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd

Part of the Databases and Information Systems Commons

Recommended Citation
https://digitalcommons.utep.edu/open_etd/2636

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.
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
to my

MOTHER
GRANDPARENTS
and SISTER

with love and gratitude
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
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
<table>
<thead>
<tr>
<th>Chapter 5: Discussion</th>
<th>.................................................................</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Participant’s Characteristics</td>
<td>...........................................................................</td>
<td>31</td>
</tr>
<tr>
<td>5.2 Participant’s Use of the Tutorials</td>
<td>...........................................................................</td>
<td>31</td>
</tr>
<tr>
<td>5.3 Analysis of Results for Hypothesis 2</td>
<td>...........................................................................</td>
<td>32</td>
</tr>
<tr>
<td>5.4 Qualitative Findings</td>
<td>...........................................................................</td>
<td>35</td>
</tr>
<tr>
<td>5.5 Initial Approaches</td>
<td>...........................................................................</td>
<td>37</td>
</tr>
<tr>
<td>5.6 Transition Approaches</td>
<td>...........................................................................</td>
<td>39</td>
</tr>
<tr>
<td>5.7 Strategy</td>
<td>...........................................................................</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 6: Conclusion</th>
<th>...........................................................................</th>
<th>43</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Limitations</td>
<td>...........................................................................</td>
<td>44</td>
</tr>
<tr>
<td>6.2 Supporting Problem-Solving</td>
<td>...........................................................................</td>
<td>46</td>
</tr>
<tr>
<td>6.3 Rewriting Help for Different Tau Levels</td>
<td>...........................................................................</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>References</th>
<th>...........................................................................</th>
<th>57</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A: Tasks</td>
<td>...........................................................................</td>
<td>59</td>
</tr>
<tr>
<td>Appendix B: Post-Session Survey</td>
<td>...........................................................................</td>
<td>62</td>
</tr>
<tr>
<td>Curriculum Vita</td>
<td>...........................................................................</td>
<td>63</td>
</tr>
</tbody>
</table>
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
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
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.
1.2  Troubleshooting Strategy Preferences

People prefer to muddle through the application through trial-and-error methods rather than use the application’s help [Ceaparu, Lazar, Bessiere, Robinson and Shneiderman 2004], [Smart, Whitting and De Tienne 1998, 2001]. And even when they can be persuaded to use help, they still appear to prefer other strategies [Novick, Elizalde and Bean 2007].

Simply adding a search function to an electronic version of a manual does not create a useful help system. Common complaints about help systems include vocabulary mismatches between the user and help, and cumbersome and confusing navigation that makes it difficult to find features [Novick and Ward 2006a]. Help systems tend to focus on the procedural steps of using a feature but often miss assisting users at the application’s task level or have little support for problem solving. The lack of a task-based perspective or support for problem solving can lead users to have an incomplete or mistaken mental model of the task on which they are working and of the application they are using. These factors—task, application and user—were identified by Kearsley [1988] as dimensions for help systems, which I have formalized as the Tau model. Users routinely run into such issues and abandon the help system to try other approaches such as muddling through the application via trial and error [Newell 1966]. Other non-help problem solving strategies include working around the problem to get a similar result with a different procedure and recalling how they completed a similar task or found the feature they needed [Andrade, Bean and Novick 2009].

My second goal in this thesis is to examine why and how users of computer applications switch among three problem-solving approaches—consulting help, muddling through with trial and error, and recalling successful uses. In this paper, I also review research related to user preferences in problem-solving behaviors and common issues across the Tau dimensions, provide a framework for analyzing interactions of eight novice users of Microsoft Publisher and its help system, and analyze when and why the subjects switched among problem-solving approaches. I examine the issues that led subjects to switch between help and non-help approaches or to favor some strategies over others within the non-help group and present a summary of my observations and a discussion of the implications of my analysis for improving help systems, and propose an adaptation of the Tau model for presenting help across various levels of experience.
1.3 Tuning Help to a User’s Level of Experience

Help systems are often tuned to a level of technical detail that does not match the level of experience of users, who find the explanations in online help to be either too complex or too basic to address their questions adequately [Novick and Ward, 2006a]. But precisely because users vary in their needs in incompatible ways, it is unlikely that any particular version of a help system could meet the needs of most users. And it remains unclear how to produce documentation that adequately targets the levels of expertise that users say they want. This issue could potentially be solved through the design and development of an interface that enables users to tune the level of technical explanation to suit their level of experience, which requires help to be written in multiple levels with respect to a specific set of dimensions [Andrade, Novick 2008].

My last goal in this thesis is to explore kinds of levels of help and what it would mean to express documentation in different points in the space defined by multiple dimensions of help. To address this problem, I review work related to the design of multi-layer and multi-dimensional interfaces, build on previous models of help for different levels of experience by formalizing a multi-level model, and develop examples of rewriting help for multiple levels. I conclude with a proof-of-concept design for a user-controlled dynamic multi-dimensional help system and a discussion of the issues raised by my approach.
Chapter 2: Background

Novice application users learn through different resources available to them; help systems, manuals, tutorials, trial and error, calling customer support or asking colleagues for assistance, but users often prefer some learning approaches depending on the availability of resources [Rieman 1996]. One of the goals of this thesis is to determine whether users prefer a single particular approach or supplement one learning strategy with another, specifically comparing the use of help systems with other troubleshooting methods. Another goal is to determine whether people can be motivated to use help if they are trained to use it through a tutorial.

2.1 APPROACHES TO TUTORIALS

The traditional tutorial, as exemplified by the tutorial provided for Microsoft Publisher, provides step-by-step guidance for learners of a computer application. The material in this kind of tutorial, although it largely contains the same content found in the help system, stands apart from the help system. An alternative approach, described in this paper, creates the tutorial from building-blocks already present in the help system. This approach minimizes duplicated effort in developing help systems and, more important from the user’s perspective, familiarizes the user with the use of the help system as an inherent property of the tutorial [Novick, Andrade, Bean and Elizalde 2008].

To illuminate the differences between the traditional and the help based approaches to tutorials, this section describes a traditional tutorial adapted from the original tutorial for Microsoft Publisher and a help-based tutorial that we developed, which covers the same material.

2.1.1 An Adapted Microsoft Tutorial

An online Microsoft tutorial for Microsoft Publisher served as the basis for both the traditional and help-based tutorials used in this study. Microsoft’s original tutorial took nearly 40 minutes to complete and had audio and visual explanations, along with practice and testing modules, on how to use basic features in Publisher such as inserting text and images, editing text and images, and changing image properties. My adapted version of the traditional tutorial comprised a set of Web pages with step-by-step text explanations with supporting graphics and was completed by participants in a pilot
experiment in 15 minutes. The step-by-step explanations had the same wording and order as the original tutorial, but the audio clips and extra modules were removed. Figure 2.1 depicts an excerpt of the traditional tutorial.

| 1. Resizing changes a picture’s dimensions by making it larger or smaller. |
| 2. Cropping trims parts of a picture away to remove unwanted portions or emphasize the portion that remains. |

Once a picture is in a frame on a publication page, you can move and resize it as described in the following table.

<table>
<thead>
<tr>
<th>To Move a picture</th>
<th>Do this Drag it to a new location.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move a picture</td>
<td></td>
</tr>
<tr>
<td>Resize a picture</td>
<td>Press SHIFT, and then drag a corner handle.</td>
</tr>
<tr>
<td>and maintain the</td>
<td></td>
</tr>
<tr>
<td>original proportions</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.1: Excerpt of an adapted traditional tutorial.

The adapted tutorial and the help-based tutorial included similar instructions on how to access help to find additional assistance, as depicted in Figure 2.2. The adapted tutorial ended with this material, and the help-based tutorial began with it.

2.1.2 A Help-Based Tutorial

The help-based tutorial explained how to use the help system featured in Publisher, as depicted in Figure 2.2, and gave participants a list of topics for which they should search for assistance. The list of topics was identical to the set of topics covered in both the original Microsoft tutorial and our adapted traditional tutorial; indeed, the topic list came from the quick-reference card provided at the conclusion of Microsoft’s tutorial. Participants who used the help-based tutorial were shown a Web page where they read the instructions on how to use help and see topics on which they could search.
The adaptation resulted in a tutorial enormously shorter than the traditional tutorial. It had the brief introduction on accessing help and the list of topics. The tutorial instructed the participant to “Use the help system in Publisher to learn about these features.” About two thirds of the list of topics is shown in the screen-shot of the help-based tutorial depicted in Figure 2.3.

While the first Web page of the tutorial is minimal, the help system is not a “minimal manual” in the sense advocated by Carroll [1990]. The help-based tutorial includes all of the material in the full traditional tutorial. The difference between the tutorials lies in the involvement demanded of users. The
traditional tutorial functions like a book, where short chapters are presented serially. The help based tutorial has the same contents, but only has a list of what the features are and suggests that the users navigate the help system to reach them. In effect, the users train themselves in use of the help system while learning about the application’s substantive features.

The purpose of the help-based tutorials was train users how to find and use the help system so they would be more likely to use an application’s help system when they encounter a problem. So in this thesis I principally address the question of whether, in realistic settings for use of computer applications, a help-based tutorial will increase use of help systems. And beyond the issue of use of help, I also address the question of why help did not lead to corresponding improvement in the use of the computer application, and what other approaches did users take.

2.2 TROUBLESHOOTING PREFERENCES

How frequently do users switch problem-solving approaches, and what causes users to prefer one approach over another? As Newell [1966] observed: “Problem solving is always a matter of search — of
starting from some initial position (state of knowledge) and exploring until a position is attained that includes the solution — the desired state of knowledge."

As users of computer systems solve problems, they generate a search space, using a variety of approaches. Rieman [1996] concluded from an observational study that users learn about software by trial-and-error, consulting a manual, or asking for help. These approaches were prevalent across users of all experience levels, although novices had no consistent approach and experts learned to recognize similarities with other applications or had a task-driven approach to problem-solving. Rieman’s study also showed that the strategy of choice was trial and error (also known as exploratory learning), which was used “just-in-time” and was based on the task at hand—as opposed to being task-free exploration. Users in the study, however, did not learn by trial and error alone; they also read the manual or talked with co-workers, depending on resource availability. The subjects used help occasionally, but used other resources infrequently.

Similar attitudes among users of help systems were described by Novick and Ward [2006b], who interviewed a diverse set of users of computer applications. The participants indicated that they preferred documentation that is easy to navigate, provides explanations at an appropriate level of technical detail, enables solving problems through examples and scenarios, and is complete and correct. But participants’ preferences sometimes conflicted, such as between a preference for coverage versus a preference for precision, which makes it difficult to satisfy all users’ preferences simultaneously. For example, providing both coverage and precision would increase the volume of help material, which would require a more complex topic structure to accommodate all the available contents and would be harder to navigate and use than a topic listing for a smaller volume of contents.

When solving problems, users appear to be highly pragmatic; they learn about the program and the task just to the extent needed to accomplish the task. This perspective on users’ strategy for accomplishing tasks in computer applications is supported by evidence from other computer-based problem-solving situations. In particular, the issue of problem-solving strategies was addressed in the context of programmers asked to modify a complex program. Koenemann and Robertson [1991] reported that programmers followed a pragmatic as-needed strategy rather than a systematic strategy.
The programmers tended to learn only about parts of the program that were directly relevant to their task.

As early as 1980, an experimental study of the effectiveness of help [Dunsmore 1980] concluded that consulting online help takes time away from problem-solving. It appears that users prefer to muddle through an application to complete a task, and even when they can be persuaded to use help they still resort to other strategies because help does not provide them with the assistance they need. Some of the issues causing high failure rates are vocabulary issues, difficulties in navigation, uncertain boundaries, lack of topic sorting by relevance, and lack of specific examples [Newell 1966],[Novick and Ward 2006a].

2.3 **MULTI-LAYER AND MULTI-DIMENSIONAL INTERFACES**

Users would also like to find the level of technical detail in help that suit their individual level of experience [Novick and Ward, 2006b], for which a multi-layer interface could be designed. Multi-layer and multi-dimensional interface designs have been proposed for user-interface applications, particularly to address changes in usability requirements as users gain experience. Review of the literature indicates that this approach has not yet been applied to help systems, which appear to be particularly vulnerable to mismatches between user and system in their respective levels of expertise [Novick and Ward 2006a], [Novick and Ward 2006b].

Multi-layer interfaces display content organized in different layers that vary across increasing levels of complexity relative to user experience. This kind of design enables users to learn in a structured sequence while limiting feature complexity [Shneiderman 2003]. This approach has been called a “training wheels” interface (see, e.g., [Carroll and Carrithers 1984]).

An example of such an interface is Shneiderman’s multi-layer interface for a text editor, which features a slider bar that moves through eight layers of feature complexity [Shneiderman 2003]. As seen in Figure 2.4a, the interface displayed a limited set of features at the first layer, offering only basic editing functions to novice users of the application. In Figure 2.4b, the interface changed as the slider bar is moved to the third layer, displaying more features and grouping previous features into submenus. Each layer in the interface presented features grouped into categories of increasing complexity such as
basic functions, editing, formatting, structures (e.g., tables and frames), and document styles [Shneiderman 2003].

Shneiderman also suggested that layers may be grouped by usage patterns, task complexity, or according to the topic structure of training textbooks. Another example of a multi-layer interface displays a step-by-step tutorial on how to build an electrical circuit [Vetere and Howard 2003]. The interface reveals additional information about the current flow in the circuit at each step in the tutorial. The interface is shown in Figure 2.5.

Figure 2.4a: Interface with low feature complexity.
Figure 2.4b: Moving the slider up increases feature complexity.

Figure 2.5: A tutorial with layered explanations.
The multi-layer tutorial does not display varying levels of complexity like the multi-layer text editor, but the layered information displayed in the tutorial shows that additional information is introduced to address common user questions identified by Kearsley [1988] such as “How can I do this?”, “What happened?”, and “How should I respond?”. Some of this information is already available in online help, but additional work is needed to provide answers at each level of user experience.

A different kind of interface, a multi-dimensional interface, was proposed by Novick [2000], who organized different aspects of a flight crew operating manual into three dimensions:

- Why: purpose of a procedure (the procedure’s goals).
- What: what to do (the procedure’s acts).
- How: how to carry out a procedure (the procedure’s actions).

These dimensions are displayed in the interface and can be toggled on or off by using check boxes to select which aspects to show or hide; they correspond to the stages of the Gulf of Execution in Norman’s model of the user-interface action cycle [Norman and Draper 1986]. The interface is displayed in Figure 2.6.

![Figure 2.6: Why-what-how dimensions.](attachment:image.png)
The why-what-how dimensions provided a model that may improve navigation in help by organizing its contents into selectable modules whose contents are categorized similar to the manual presented by Novick [2000]. Also, as we will show in Section 4, these dimensions vary across multiple levels. The multi-layer and multi-dimensional interfaces presented examples of models that have been applied to computer applications and which could be applied to the design of online help by separating help content into different dimensions and varying the content across multiple layers relative to complexity levels. Even if a set of dimensions could be defined, though, the difficulty lies in categorizing the help content into layers at the “right” level of help for users with varying degrees of experience.
Chapter 3: Methodology

This thesis partly focuses on determining whether a help-based tutorial would increase the use of help systems and would enable users of computer applications to be more effective in accomplishing tasks. Specifically, I hypothesized that

1. Participants using the help-based tutorial would use help more than participants using the traditional tutorial; and
2. Participants using the help-based tutorial would perform better on novel tasks than participants using the traditional tutorial.

In fact, until the analysis phase of this study, I expected that the results for both hypotheses would be negative. My experience in observing users of computer applications had suggested that changing the tutorial might not have a large enough impact on the participants to change their ingrained behaviors with respect to use of computer applications.

To test these hypotheses, I developed the two tutorials, described in Section 2.1.1, to teach people how to use Microsoft Publisher 2003. I conducted a between-subjects experiment comparing the effects of use of these tutorials. In this section, I describe the subjects who participated in the study, outline the experimental design, and briefly describe our post-session survey [Novick, Andrade, Bean and Elizalde 2008].

The second focus is on determining alternate troubleshooting strategies used by the subjects when help was not useful; in particular the context in which users chose one strategy over another. For example, this thesis describes at which point during a task users decided to stop reading help to try other methods for a particular sub-task and how I decided to classify these transitions along with the categories of strategies used by the subjects.

3.1 Subjects

For study subjects, I sought relatively experienced users of Microsoft Office applications who had little or no experience with Microsoft Publisher. I recruited 22 administrative assistants at the
University of Texas at El Paso (UTEP), of whom 21 were professional staff and one was a student employee. Subjects were not compensated for their participation.

The study participants worked in different departments within the university performing clerical work, mainly with Microsoft Office applications. I selected adults between the ages of 22 and 67. All participants were female with an average age of 45.5 and worked as administrative assistants at the University of Texas at El Paso. Participants had, on average, more than 17 years of experience in their profession and spent an average of 7.5 hours a day on the computer. Although I did not select participants on the basis of sex, the prevalence of women among the pool of potential subjects led to having all female participants. All subjects, other than the student employee, had a college education. Most of the subjects had at least 15 years experience as administrative assistants, while the rest had been working for eight years or less. Four subjects indicated that they helped others frequently on using other office applications such as Microsoft Word; twelve helped others with Word less frequently; and six of the subjects did not assist others. Fifteen subjects reported using a computer eight hours a day; one participant reported using it for ten hours a day; and six of the subjects used a computer for six hours or less each day.

Participants self-assessed their general proficiency with computers with a mean 3.18 on a scale from 1 to 5, indicating that they considered themselves to be of average proficiency. Participants self-assessed their specific proficiency with Microsoft Publisher with a mean of 1.64 on a scale from 1 to 5, indicating that they considered themselves to have little or no proficiency with the application.

Even though Nathaniel Bean (my research associate) and I designed the tasks to permit all participants to complete the tasks, only differing in the amount of time to completion, the average task completion score was 3.8 out of 5.

3.2 Experimental Design

I used a between-subjects balanced design, assigning subjects randomly to the traditional and help-based tutorial conditions. Each session lasted about an hour and was conducted at the participant’s own office, where she used her office computer for the tutorial and tasks. All of the computers ran the Windows XP operating system. Six of the participants had Publisher 2007 and 16 participants had
Publisher 2003. I gave each tutorial to every other subject, so that eleven subjects in our study completed the Microsoft tutorial, and the other eleven subjects completed our tutorial.

Nathaniel and I asked the subjects to complete a computer-based tutorial that lasted 15 minutes, after which they were given 40 minutes to complete four tasks, where ten minutes were allocated for each task.

Subjects were asked to replicate in Publisher every part of a one page document (the “reference design”), which was handed to them for each task. Subjects were allowed to refer back to their respective tutorials if they needed assistance.

We designed two of the four tasks so that the reference design could be replicated using features covered in both tutorials. The other two tasks depended on features not covered in the tutorials. For each session, the first task had familiar features and was followed by a task with unfamiliar features, which was followed by another task with familiar features, and the last task again had unfamiliar features. The tasks and the features used to create them were

1. Arrow sign: change the appearance of a text box, add a picture, rotate a picture, add a background color or color frame to an image (features covered in the tutorial).
2. Business card: locate the correct template, change the organization logo, add a decorative frame to a text box (features not covered in the tutorial).
3. Golden apple advertisement: divide a textbox into columns, recolor a picture, change how text wraps around a picture (features covered in the tutorial).
4. Raise petition: change the color of a picture to grayscale, add a shadow effect to text (features not covered in the tutorial).

Appendix A shows the reference designs for the tasks, which we provided to the participant in the order they were listed above and gave the participant ten minutes to complete each task, for a total of forty minutes for the task session. We told the participants that they could use any non-human resource to complete these tasks.

3.3 POST-SESSION SURVEY

After the tutorial and task sessions, we asked the participants to complete a questionnaire with six questions about their general experience with computers and their attitudes towards learning new
software, where for each question each participant had a choice of options ranging from “I strongly agree” to “I strongly disagree” on a six-point Likert scale. The survey can be found in Appendix B.

Ten participants strongly agreed that they enjoyed working with computers; eleven agreed; and one person slightly agreed. Fourteen subjects expressed disagreement with needing other people to help them; five slightly agreed; two agreed; and one subject strongly agreed. Almost all participants reported being comfortable learning new programs, while all participants agreed that learning new programs is a good use of time. Finally, most subjects only slightly agreed that they were better computer users than average, and seventeen strongly agreed that learning about the programs they use at work is a good use of time [Novick, Andrade, Bean and Elizalde 2008].

3.4 DATA CODING

The audiovisual recordings of the participant’s use of Publisher were coded for amount of time (in seconds) spent using the help system and number of times accessing help.

Nathaniel and I assessed the participants’ task performance by printing the documents they created and judging performance relative to the reference design the participants were asked to replicate. We designed each task with a number of features in mind and determined how much the subject tasks looked like the template designs we produced and ranked them accordingly from the scale of 1 to 5 as follows

- 1 for doing minimal work (namely, only inserting the arrow icon for task 1)
- 2 for using one feature
- 3 for using two features
- 4 if they used all three features for the current task but did not exactly match the template
- 5 for a near-identical re-creation of the task.

We evaluated inter-rater reliability with Cohen’s Kappa [Carletta 1996]. Kappa scores for the four tasks ranged from 0.79 to 1.00, and Kappa for overall agreement on the four tasks was 0.89. This suggests that the raters’ judgments of task performance were highly reliable.
3.5 **ASSESSING TROUBLESHOOTING STRATEGY PREFERENCES**

To determine the transitions people make among computer application problem-solving approaches, we analyzed the high definition audiovisual recordings collected as part of the study. Eight out of the 22 subjects from the previous study were selected because they had the highest number of accesses to the help system, averaging eight accesses per task.

The interactions the subjects had with the application and the help system were classified in episodes to determine the set of strategies that the subjects used for each task step, where completing a step required them to use a number of features with which I created the reference designs. Though the boundaries between episodes were sometimes not clear (see [Novick 2009]), I attempted to define the episodes as beginning when:

- A subject began a new task.
- A subject began working on a part of the task that involved one of the Publisher skills that the reference documents were intended to elicit.
- A subject switched from one problem-solving approach to another. For example, the subject was reading help or the tutorial and then began working on a step in the task through trial-and-error.
- A subject stayed in the same condition, but began a new attempt. For example, the subject started a search with a relatively general search term, read a help topic, and then began a new search with a more specific search term, perhaps based on knowledge gained from reading the help topic.

Similarly, episodes ended when:

- The subject completed a task.
- The subject completed a step.
- The subject abandoned a step after repeated trials.
- The subject was interrupted because time expired for that task.

The problem-solving approaches used by the subjects to accomplish tasks in the application were categorized as help-based and non-help-based. Approach preference varied across subjects, especially as they gained more experience with the application, namely they began to reuse some of the material they
had learned and explored the interface less. Although this could be considered training effects, I argue that the tasks were designed such that all steps required different features – no feature could be used across tasks.

I found two help-based approaches:

- Starting a search in the help system or looking back at a topic window they left open.
- Looking back at the tutorial to review how to use a feature they perceived as necessary to do a particular step. In the case of the help-based tutorial, they scanned the list of topics for suggestions on what to search in the help system.

I found two non-help-based approaches:

- Exploring application menus and toolbars to find relevant or similar features for what subjects needed to do, learning about the features by trial-and-error. That is, the subjects were trying to find features that appeared to be pertinent to what they wanted to do. In some cases, subjects using this trial-and-error (“T&E”) approach accomplished tasks through work-arounds—solutions that approximated or provided the appearance of doing something rather than actually doing that thing such as, for example, arranging text in columns by inserting tabs rather than setting the number of columns.
- Recalling how to use a feature, or related feature, about which the subject had learned in a previous task. Recall differed from other approaches because the subjects made precise moves to the features they needed, taking little time to access the feature. Recall episodes averaged less than 30 seconds, compared with about 77 seconds for T&E episodes and about 71 seconds for help episodes. This suggests that subjects remembered how to find these features and did not rely on exploration to find them. Recall was used more in the last two tasks because by then most subjects had gained enough experience with Publisher to recall features that they had used. The tasks were designed based on the feature set covered in the training materials, which covered a small set of basic functions; thus many features were similar across different kinds of objects (e.g., borders for text boxes and images).
In the 124 episodes we studied, the subjects used the T&E approach more than any other as we will discuss later in this thesis. To take one episode, for example, a subject intended to add a shadow effect to text. The subject inserted text in a text box and then clicked on Word Art in the Objects toolbar. She tried different Word Art styles during six minutes to see find one that would approximate the format of the message in the reference task. She then clicked on Text Box, typed part of the message and highlighted it, looked for a text formatting option or a font type that would have a shadow but found nothing relevant, clicked on Format, paused for eight seconds on Text, looked at the available effects, clicked on Shadow, and clicked on OK to make the changes.

Within the T&E approach, users created work-arounds in some tasks more than in others, particularly in task 3, the “Golden Apple Advertisement” task shown in Figure A.3. In this task, the subjects often simulated the two columns on the lower text box by typing text and using the space bar or tabs to push the text over to the far right of the text box. Also, some subjects separated the top and bottom sentences of the upper textbox by typing them and using the Enter key to insert lines rather use Publisher’s text-wrapping options for images. In contrast, the “Business Card” task, shown in Figure A.2, offered fewer opportunities for work-arounds.

The subjects used the recall approach when they had used similar features earlier in their session. For example, some subjects recalled recoloring the apple in the advertisement and were able to find and use a similar function in completing task 4. In all, subjects used the recall approach in 34 episodes.

In 64 of the episodes, subjects used a help approach to problem solving. In one episode, for example, a subject intended to rotate an image. She searched for “rotate” in the help system, clicked on the first topic (“Rotate or flip objects”) from the list of topics resulting from the query, and then clicked on first sub-topic (“Rotate objects freely”). The subject followed the instructions and was able to rotate the arrow [Andrade, Bean and Novick 2009].

The issues that the subjects encountered when using Publisher and its help system were related to the task characteristics, the application and help themselves and to the individual understanding that each subject had of both. As such, these issues can be classified within the Tau framework, which consists of user, task and application characteristics as defined by Rieman [1996]. Such dimensions enabled me to make clear distinctions of the kinds of usability problems that caused subjects to switch.
strategies. As an example, subjects who had an inaccurate or incomplete understanding of the task looked for assistance within help and gave up due to a vocabulary mismatch between the user and the application, resulting in the subject giving up on help and resorting to work-arounds.

3.6 THE TAU (TASK, APPLICATION AND USER) MODEL

Kearsley identified three important dimensions for help systems: the user’s task experience, program experience, and computer experience:

- Task experience describes the user’s experience regarding the particular task or set of tasks for which an application or group of applications were designed. Namely, a person may be an expert in statistical analysis, but does not know how to use the statistical formulas in Microsoft Excel.
- Application experience denotes the user’s level of experience with respect to a particular program.
- User experience refers to the user’s general experience using computers and other applications.

I express Kearsley’s dimensions as a function called Tau (abbreviated as τ) for task, application, and user. Thus τ(t,a,u) represents the level of experience of a user with respect to each dimension. Following Kearsley, and Covi and Ackerman’s [1995] adaptation of the model, different values of the τ parameters situate a particular user in a three-dimensional space possible expression of help, as depicted in Figure 3.1.
In the simple initial model of the τ dimensions, I used the following level ratings for users: 1 for novice, 2 for intermediate, and 3 for expert.
Chapter 4: Results

My study showed that people can be motivated to use help if they are guided on how to do so, but their performance does not improve from using help because in most cases they do not find it useful and resort to other strategies for troubleshooting application issues to do their work.

4.1 Use of Help Across Subjects

My first hypothesis posited that participants using the help-based tutorial would use help more than participants using the traditional tutorial. As indicated in Table 4.1, this hypothesis was confirmed. In terms of mean seconds using help, participants using the help-based tutorial used help about eight times more than participants using the traditional tutorial.

<table>
<thead>
<tr>
<th></th>
<th>Mean Total Seconds Using Help</th>
<th>Mean Number of Times Help Was Accessed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help-Based</td>
<td>385.9</td>
<td>5.1</td>
</tr>
<tr>
<td>(n1 = 11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traditional</td>
<td>46.2</td>
<td>1.2</td>
</tr>
<tr>
<td>(n2 = 11)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This difference was significant (p < 0.05, one-tailed t-test, unequal variance). This pattern held true for each of the four tasks. Participants in the help-based tutorial also accessed help a greater number of times (p < 0.01, one-tailed t-test, unequal variance) than those in the traditional tutorial. These results are particularly striking in light of the small numbers of participants in the two conditions.

My second hypothesis posited that participants using the help-based tutorial would perform better on the assigned tasks than participants using the traditional tutorial. As indicated in Table 4.2, this hypothesis is not confirmed. In general, participants who used the help-based tutorial did not perform better on the tasks than participants who used the traditional tutorial.
Table 4.2: Results for hypothesis 2 (overall n = 22).

<table>
<thead>
<tr>
<th></th>
<th>Mean Task Performance Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help-Based</td>
<td></td>
</tr>
<tr>
<td>( n_1 = 11 )</td>
<td>3.81</td>
</tr>
<tr>
<td>Traditional</td>
<td></td>
</tr>
<tr>
<td>( n_2 = 11 )</td>
<td>3.82</td>
</tr>
</tbody>
</table>

Participants using the help-based tutorial got higher scores (not significant, two-tailed t-test, equal variance) in tasks 3 and 4 and slightly worse (not significant, two-tailed t-test, equal variance) in tasks 1 and 2. Given that tasks 2 and 4 posed problems not covered in the tutorials, the effects of increased help use should have been greater in these tasks; however the data do not suggest an effect [Novick, Andrade, Bean and Elizalde 2008].

Every subject in the experiment did the tasks in numerical order (1, 2, 3 and 4), which accounts for fluctuations in performance, particularly because once subjects learned features from task 3 they could use them to mimic the features they needed in task 4. Task 2 was also designed to be notoriously difficult, and perhaps giving it to subjects as the last task would have enabled subjects to try different solutions to approximate the task once they had gained some experience with Publisher. I argue that the difficulty level might have been more balanced had the task order been permuted across subjects.

Participants who used help mentioned that they did not find the contents useful for the steps they were trying to complete, and even when they did search for help they would give up searching for assistance or trying the steps outlined in the help topics to do the task by other means such as recalling a step they had done before that was similar, working around the problem or exploring the interface to find the features they considered relevant to the task.
4.2 TROUBLESHOOTING APPROACHES

To understand why subjects preferred some strategies over others, I used the annotations of the subjects’ problem-solving episodes and developed a representation of their patterns of transition among problem-solving approaches, analyzed possible causes of these transitions, and examined the subjects’ choices of problem-solving approach for initial episodes and for transition episodes.

4.2.1 Transition Patterns

Newell [Newell 1966] tracked problem-solvers’ searches in a representation called a problem behavior graph, which described the search space in terms of transitions between states of knowledge. In my study, I developed a more abstract representation that describes the subjects’ transitions between problem-solving approaches rather than the detailed problem-solving steps taken using the approaches. For each episode, I created a state-transition diagram that represented the subject’s use of a problem-solving strategy, including a start state, a problem-solving approach, and a transition state. Figure 4.1 shows the state-transition representation for two episodes in sequence in which a subject tries to accomplish a task step through the T&E approach, fails, switches to the help approach, and succeeds.

![State-transition diagram](image)

Figure 4.1: A subject fails using trial-and-error, switches to help, and succeeds.
Summing up all the individual state transitions from 124 episodes, I generated a finite state diagram of the transitions among the problem-solving approaches used by the subjects in the study, as shown in Figure 4.2. Users reached the end state when they succeeded in using any of the problem-solving approaches and switched to another approach when the current one had failed. Recall was by definition successful; if subjects tried to use a remembered feature but failed to accomplish the task step, the episode was coded as using the T&E approach [Andrade, Bean, Novick 2009].

4.2.2 Possible Causes

For each episode, we analyzed the possible causes of transitioning from one strategy to another. Our analysis was based on six elements:

- What the subject was looking for when typing in help or exploring the interface. For exploring, I was able to infer what the subject was looking for because I had the task
specification at hand and watched up to one additional minute forward on the video to see on which step she was working.

- Whether the subject’s activity was directly related to accomplishing the task, or if it was for some other purpose, such as trying to understand instructions that were not clear on a previous help topic.
- The result of the subject’s consulting help or searching for the feature. This involved following the text and mouse-overs that followed on the video.
- The amount of time of the episode.
- The reasons for the subject’s success or failure in the episode. I noted the point at which the subject failed and looked at the actions that preceded the failure.
- Whether the subject was able to apply any help she received and the reasons for this.

I used these elements and the state transition diagram to classify transition patterns and to identify the factors that appear to lead to the transitions, as discussed in the next section.

4.2.3 Choice of Approaches

As the subjects interacted with Publisher and its help system, they began each task or major task step with an initial approach. Table 4.3 presents the distribution of the approaches used in these initial episodes. Subjects could transition between episodes with the same approach; for example, a subject could end a T&E episode and transition to a new T&E episode if she started a new step with in the task. Similarly, a subject could end a help episode and start a new help episode if she searched for a different term [Andrade, Bean, Novick 2009].

Some initial episodes were directly successful—the subject completed the task step without having to transition to another episode. We note that the recall approach was successful by definition, so its high success rate in this table is expected.

For episodes in which the use of an approach was not successful, subjects would have to transition to another approach or abandon the task step, as shown in Table 4.4. The table presents the sequence of approaches involved in the transition, the number of episodes resulting from the transition that succeeded or failed for the task step, and the rate of success for resulting from the transition (i.e., the percentage of episodes into which the subject transitioned resulted in task success).
Table 4.3: Initial episode successes.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Instances</th>
<th>Successes</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>14</td>
<td>14</td>
<td>100%</td>
</tr>
<tr>
<td>T &amp; E</td>
<td>30</td>
<td>15</td>
<td>50%</td>
</tr>
<tr>
<td>Help</td>
<td>3</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>47</strong></td>
<td><strong>32</strong></td>
<td><strong>68%</strong></td>
</tr>
</tbody>
</table>

Table 4.4: Transitions between approaches.

<table>
<thead>
<tr>
<th>Transition</th>
<th>Instances</th>
<th>Successes</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help → T &amp; E</td>
<td>25</td>
<td>7</td>
<td>28%</td>
</tr>
<tr>
<td>T &amp; E → Help</td>
<td>25</td>
<td>10</td>
<td>40%</td>
</tr>
<tr>
<td>T &amp; E → T &amp; E</td>
<td>10</td>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>Help → Help</td>
<td>11</td>
<td>3</td>
<td>27%</td>
</tr>
<tr>
<td>Help → Recall</td>
<td>3</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td>T &amp; E → Recall</td>
<td>3</td>
<td>3</td>
<td>100%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>77</strong></td>
<td><strong>27</strong></td>
<td><strong>35%</strong></td>
</tr>
</tbody>
</table>

The subjects’ most frequent transitions were from the T&E approach to the help approach (25) and from the help approach to T&E approach (24). In both cases, seven of these transitions led subjects to succeed by switching approach. The transitions to the recall approach were successful by definition. Otherwise, the transition with the highest resulting success rate was the transition from the help approach to the T&E approach; 42% of these transitions resulted in success for the task or step [Andrade, Bean, Novick 2009].
Chapter 5: Discussion

I now discuss related findings and observations, including characteristics of the participants, their use of the tutorials, and possible reasons for the failure to confirm Hypothesis 2. I also discuss the subjects’ patterns of problem-solving, their choice of approach, and possible strategies for transitioning between approaches.

5.1 Participant’s Characteristics

A significant correlation (-0.44, p < 0.05) indicated that task performance suffered as participant age increased. Self-reported Publisher proficiency correlated positively (0.52, p < .05) with task performance scores. Additionally, individuals who rated themselves as more able computer users performed better on the tasks (Pearson correlation -0.53, p < .05). These results suggest that participants were, to some degree, aware of their own abilities. Reporting less confidence with learning new computer programs correlated positively (0.45, p < .05) with increased use of the help system.

5.2 Participant’s Use of the Tutorials

In general, participants who trained with the traditional tutorial read through the bulk of the tutorial and practiced some of the features that were covered in it, only reading the last page, which suggested how they could use help if they needed it, without spending any time in help during the tutorial session. Most subjects finished the tutorial early and did not use the whole 15 minutes available for training. Most subjects practiced using features on a blank page in Publisher as they trained.

Participants who trained with the help-based tutorial read through the tutorial and searched for a few topics and practiced using some features, but tended to practice on fewer features than participants who trained with the traditional tutorial. Participants who used the help-based tutorial tended to explore fewer features and terminated their tutorial session earlier than participants using the traditional tutorial. For example, a participant using the help-based tutorial did not follow directions during the tutorial and was able to accomplish most of the tasks through trial and error. Few subjects spent time practicing features in Publisher because they did not search for many features. Instead, they spent more time reading about the few topics for which they found help. I argue that this is because the traditional
tutorial had more feature coverage because more features were embedded into the examples that were provided in the tutorial and subjects who practiced the examples had exposure to more features as a consequence. Another argument is that even though the help-based tutorial listed all the features, it lacked concrete guided examples that the subjects could follow through to learn each feature. This is compatible with the findings of Rieman [1996] that users learned to use applications in the context of a particular task or tasks as opposed to learning the features without a particular task in mind.

5.3 ANALYSIS OF RESULTS FOR HYPOTHESIS 2

The negative result with respect to improvement in task performance may reflect differences in the difficulty of the tasks. As indicated in Table 5.1, the mean task performance for task 2 was lower than that of the other tasks. This effect was particularly apparent in the help-based tutorial condition.

Table 5.1: Mean task performance, by condition and task.

<table>
<thead>
<tr>
<th></th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help-based</td>
<td>3.59</td>
<td>2.91</td>
<td>4.36</td>
<td>4.36</td>
</tr>
<tr>
<td>Traditional</td>
<td>3.95</td>
<td>3.45</td>
<td>3.77</td>
<td>4.09</td>
</tr>
</tbody>
</table>

As task 2 required use of functions not covered in the tutorial, the inherent difficulty of this task may have swamped the experimental effect because one of the steps required that the subjects used a specific feature and could not be replicated with work-arounds and no features from other tasks could produce the same outcome. In particular, the subjects had to change the logo design on the card and had to choose a different design from a gallery, and even when they tried to use other features they had to separate the logo picture from the text box beneath it.

I did not balance task assignments across conditions, as the tasks 2 and 4 necessarily had to involve functions not covered in the tutorial. Normalizing the help-based scores by the traditional scores did not clarify the results because the subjects in the help-based condition for task 2 performed relatively worse than the subjects in the traditional condition. If it were possible to calibrate the tasks in advance of
the experiment, it is possible that I could have minimized differences in difficulty between the “covered” and “not-covered” tasks [Novick, Andrade, Bean and Elizalde 2008].

Ironically, the positive outcome for Hypothesis 1 may have inadvertently caused the unexpectedly poor task performance of those in the help-based group. Participants in the help-based group spent more time using help and thus took longer to transition to the trial-and-error approach of the participants in the traditional group, who may have felt freer to give up using help. That is, if participants did not find help useful, then time spent in help was time not usefully spent on completing the task.

Consistent with this interpretation, the data for task performance also suggest that Hypothesis 2 was disconfirmed in part because participants used help less as they worked their way through the four tasks. Tables 5.2 and 5.3 illustrate my main finding that participants using the help-based tutorial used help longer and more often than participants using the traditional tutorial. Indeed, for the traditional tutorial, these data are consistent with our observation from the literature that users of computer applications in general rarely use help.

Table 5.2: Mean seconds using help, by condition and task.

<table>
<thead>
<tr>
<th></th>
<th>Task</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Help-based</td>
<td>135.0</td>
<td>104.7</td>
<td>62.1</td>
<td>57.6</td>
</tr>
<tr>
<td>Traditional</td>
<td>25.2</td>
<td>12.8</td>
<td>8.2</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 5.3: Mean accesses of help, by condition and task.

<table>
<thead>
<tr>
<th></th>
<th>Task</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Help-based</td>
<td>2.0</td>
<td>1.4</td>
<td>0.9</td>
<td>0.8</td>
</tr>
<tr>
<td>Traditional</td>
<td>0.6</td>
<td>0.4</td>
<td>0.2</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Tables 5.2 and 5.3 also show that use of help decreased monotonically as a function of task. This result was unexpected, as tasks 2 and 4 required the use of functions not covered in the tutorial, which should have led to increased use of help, especially for task 2, which was the hardest. The trend of decreasing use of help further meant that the participants in neither condition used help more when faced with functions not covered in the tutorials.

If the use of help continued to decrease monotonically, it would eventually have to asymptote. Indeed, for participants using the traditional tutorial, use of help reached zero seconds and zero accesses by task 4. This may have been because task 4 was the easiest of four tasks, given the results in Table 5.1, but this may also have been because the participants gained proficiency as they made their way through tasks 1 through 3.

When I reviewed the tasks following the experiment, I discerned that task 4 was likely easier for participants because, having re-colored the apple in task 3, they could easily create the effect of graying out the money symbol because they could, and often did, simply re-color the money symbol gray. Applying a shadow to the text proved also to be a relatively less difficult subtask because this function exists in most Microsoft office products. And, as task 4 was relatively less complex the participants’ performances received higher scores for doing less.

Interestingly, though, as shown in the graphical representation of the data in Figures 5.1 and 5.2, and despite the relative easiness of task 4, the rate of decrease of use of help for the participants using the help-based tutorial appears to flatten out markedly for task 4. A rough extrapolation of the trend for help-based tutorial participants would have otherwise declined to near the zero-level reached by the participants using the traditional tutorial. This suggests that possibly the novel functions required in this task prompted greater use of help than would have otherwise been prompted by the overall trend toward decreasing use of help.
5.4 Qualitative Findings

Most of the participants in the traditional tutorial condition looked at the traditional tutorial for reference while completing the tasks. Only a few of the participants in the help-based tutorial condition
referred back to the tutorial while completing the tasks; rather, they tended to go directly to the help system.

Although Nathaniel and I did not design the experiment to stress the participants, some reported that they felt stressed and confused by being asked to complete the tasks. One participant dropped out because she felt that she would not be able to complete the tasks, and another stated as she finished a task, “I failed at this one too, sorry.” This comment surprised me because, consistent with best practices for usability tests, I had emphasized that the participants were not to be concerned if they were unable to complete the tasks and that it was probably the system’s fault and not theirs. Another participant complained for about five minutes about not being to do the task, not knowing where to start, and so forth. Yet after she started actually working she did a great job relative to the other participants.

When participants did use the help system, they often found that it did not actually help them. Participants typically lacked knowledge about the correct terminology to use when searching; sometimes they even looked at the tutorials just to find the right words to use for the help system. One participant even tried using the help system to search for clip art, and another tried using the Research system (under the Tools menu) as the help system. The existence of both a local online help and a Web-based online help system led to more confusion. Participants appeared not to understand how the two differed and would often get stuck in the Web-based help system. Moreover, the materials offered by the two help systems are not complementary and often overlapped. One participant, in trying to locate clip art, was led to use the Web-based help system, then led to the online clip art section where she located the correct clip art. If allowed to continue, she would have spent all of her time trying to download that piece of clip art even though it was already available locally on her computer.

When using help, participants spent most of their time searching with different keywords, skimming the results, and sometimes looking into the specific topics. Even within the specific topics participants typically only skimmed the information and often could not correctly use it even when their search had led them to the correct topic. This may occur because participants are unsure if they reached the right topic and know that it would be a great waste of time to read everything presented. It was not uncommon for participants to open up an incorrect help topic, read it for a minute (and even try to use the information to perform a task), and then eventually realize that they were in the wrong topic. Thus
the participants’ pattern of skimming, even in the correct help topics, becomes understandable, but compounds their problems in that, when in the correct topic, they skimmed the material and thus were unable to take sufficient advantage of the help available for their task. As a result, they would sometimes abandon both looking for help and working on the task [Novick, Andrade, Bean and Elizalde 2008].

Many of the participants approximated the reference design and then gave up, especially with respect to details. Many of the participants’ workarounds proved quite effective. Examples of effective workarounds included, among many others:

- Using two text boxes instead of creating columns
- Re-coloring a picture instead of graying it out
- Placing a separate colored box behind a drawing instead of filling the drawing’s background
- Drawing a box to put around text instead of coloring the lines of the text box itself.

These workarounds also suggest that participants’ understood the reference designs, and thus the tasks of creating them, in ways that differed significantly from the ways in which the reference designs were created. I will now discuss how these differences in understanding the task and the help system affected the choice of troubleshooting strategies amongst users.

5.5 INITIAL APPROACHES

When subjects started a new task, they began with one of the initial episodes summarized in Table 4.3. The recall and help approaches had much higher success rates than the T&E approach. The reasons for this difference are due to the definition of recall, subjects’ perception of the appropriateness of help, and the actual difficulty of completing the tasks using exploratory methods as explained below.

Recall. If subjects understood the task step and remembered an appropriate way to accomplish the task, then they would of course use the recall approach. In practice, subjects probably experienced degrees of recall. That is, they might have recalled something closely similar to or hinting at to the new task. In this study, we coded all of these as recall.

In my study, the subjects used the recall approach 14 times out of 47 initial episodes. My analysis of the individual episodes suggests that the necessary knowledge of task and application came from the subjects’ initial training, from the subjects’ prior use of a similar function earlier in their
session, and, increasingly as they continued the session, from the subjects’ prior use of exactly the needed function.

If subjects were not aware that they had the necessary task and application knowledge to take advantage of recall, then they would have to start with either the T&E or help approach.

**T&E.** Episodes using the T&E approach amounted to about 64% of all the initial episodes. Of these initial T&E episodes, 50% were successful. However, this success rate reflects a higher success rate for work-around solutions than deeper solutions. In work-around episodes, subjects succeeded (again as a matter of definition) in 100% (13 out of 13 episodes) of the initial episodes. In T&E episodes where the subjects attempted to do better than a surface solution, the success rate was only about 12% (2 of 17 episodes). This suggests that subjects applied approximate methods if they could, and that, naturally enough, solutions through these approximate methods achieved nearly the same effect but were easier or faster to achieve than solutions through methods that would have been more appropriate because, for example, they would be more robust for future situations. Moreover, in the cases of the recall and help approaches, and in the case of the work-around category of T&E approaches, the subjects apparently had enough knowledge to proceed reasonably directly to a solution. It is in all of the other cases—those that fall into the category of using exploratory learning to achieve the more robust solutions that experts would apply—that the subjects faced problems that posed substantial difficulty. Thus it is not surprising that the subjects’ success rate was only 12% for these cases.

**Help.** When used as an initial approach for a task step, help was always successful, but subjects did this in only 3 of their 47 total initial episodes—and these were the 8 of 22 subjects who used help the most. In all three instances, these were cases of subjects who searched the help system and used the information in a help topic to complete the task step, rather than cases where the subjects returned to the training materials. One interpretation of this result is that the subjects were aware of their knowledge levels and chose to consult help only when they knew enough to formulate an appropriate query. Alternatively, subjects may have under-used help because they overestimated the utility of the T&E approach. Indeed, subjects may overestimate the utility of T&E in part because, from their perspective, a work-around is an acceptable solution. That is, the number of perceived solutions through T&E is much
higher than the number of actual, deeper solutions, leading to the perception that the success rate for T&E is 50% rather than 12%.

5.6 TRANSITION APPROACHES

Where the subjects’ initial approach was unsuccessful, subjects transitioned to another episode (or abandoned the task). Subjects could either initiate a new episode of the same approach (e.g., search on a new term, pick a different task step) or initiate a new episode with a different approach. In this section, I first review the results of transitions to an episode with the same approach and then review the results of transitions between approaches.

In 22 of the 78 total transitions, the subjects began a new episode with the same approach as the current episode.

**Help → Help.** In transitions from help to help, it appears that the subjects were not finding what they needed in the current help episode, so they refined or changed their query. However, the subjects’ missing or incomplete knowledge of the task or the application that led their query to fail in the first place appears not to have been ameliorated by the help topics they had been able to find so far: only one of the help → help transitions led to a successful help episode, for a successful outcome rate of 10%.

**T&E → T&E.** In transitions from T&E to T&E, the subjects typically abandoned a step inside a task and began work a different step in the same task. Subjects could do this because some steps did not depend on other steps in the task. In the “Golden Apple Advertisement” task, for example, the steps of inserting the apple image and creating the two-column text box did not depend on each other. As a strategy, this transition was successful in about 27% of cases. This low success rate was likely due to the circumstance that these were the difficult task steps that had not been amenable to recall or workaround methods. If there had been a work-around solution for this step, the subject would have already applied that approach.

In all of the other 56 transitions, the subjects began a new episode with a problem-solving approach that differed from the subject’s current approach.

**Help → Recall.** Of the 37 episodes where the subjects transitioned from help, in two instances the subjects transitioned to recall. These relatively rare cases occurred where subjects did not follow the
help instructions but instead for some reason recalled the way to accomplish the step. As in the case of
the initial episodes, if the subjects knew how to solve the problem then they did so.

**T&E → Recall.** Of the 38 episodes where the subjects transitioned from T&E, in two instances
the subjects transitioned to recall. These are cases where the subjects’ exploratory learning in the
application interface apparently reminded them that they already knew how to accomplish the task step.
Because they reflect learning, albeit exploratory, leading to recall of how to do something, these
transitions seem similar to those where learning from the help system also led to recall. Indeed, the
success rates (2 of 37 and 2 of 38) for the two kinds of transitions reinforce the idea that they reflect a
common factor of effectiveness of learning as a trigger for recall.

**Help → T&E.** There were 25 transitions from help to T&E. Of these transitions, 40% (10 out of
25) led to successful outcomes in the T&E episode. Because, as we have seen, transitions from help to
help (i.e., refining or changing a search) produced almost no direct successes, transitioning to T&E was
a much more promising strategy. Perhaps this disparity between outcomes is because the subject’s lack
of knowledge or incomplete model of the task and application would lead to a second search as fruitless
as the first, and the subject could learn more effectively from exploration of the interface.

**T&E → Help.** There were also 25 transitions from T&E to help. About 28% (7 out of 25)
transitions from T&E to help led to successful episode outcomes.

### 5.7 Strategy

Given these results, what strategy for choice of problem-solving approach should a rational user
of computer applications employ? In sum, the data indicate that:

(a) The initial episode success rate for help is 100% (but for N = 3);

(b) The first-episode success rate for T&E is 50% (but all except two successes were
shallow-solution work-arounds);

(c) The overall transition-to-help success rate is 23% (8 of 35); and

(d) The overall transition-to-T&E success rate is 34% (12 of 35).

From the standpoint of a user of a computer application, an interesting issue is the choice of what
do after a failed T&E episode. Should the user stick with T&E or switch to help? The chances of success
in the next episode are nearly equal (27% vs. 28%). Some subjects retried a T&E approach in the next
episode, but in most cases they ended up switching approaches or abandoning a step after retrying either strategy; in many cases switching led to eventual success.

The analysis suggests that the rational user should pursue a strategy of changing approaches back and forth between help and T&E rather than a strategy of sticking with either one. And, in fact, the most common transitions by far were from help to T&E and from T&E to help [Andrade, Bean and Novick 2009].

It is possible that a strategy of switching approaches enables users to overcome two of the key causes of failure within episodes: vocabulary mismatch and incomplete or mistaken mental models of the task or the computer application.

Vocabulary mismatches occur because help systems are likely to use the same terms as the application, which requires users to familiarize themselves with such terms to learn about the application. Users often do not know which words to use when searching for help on a feature [Novick 2006a]. In my study, for example, a subject searched for “Background color” to learn how to add a colored background behind an image but did not find appropriate help because the feature name was “Fill color.” Another subject was unfamiliar with the terms “menu” and “toolbar;” when help instructed her to find a feature in the Objects toolbar she kept looking for the Objects option inside the menus at the top of the application window. Sticking with help or T&E as an approach may tend to reinforce rather than solve vocabulary mismatches. For example, vocabulary mismatches can lead users into “dead ends” in the interface that look appropriate but are, in fact, not. Instead, switching between problem-solving approaches may enable users to gain a different perspective from which they can resolve vocabulary mismatches.

Likewise, switching approach may also help subjects overcome problems of incomplete or mistaken mental models of the task or the application. In our study, subjects sometimes appeared to have incomplete or mistaken understandings that led to using workarounds to approximate some task steps, particularly on task 3, the “Golden Apple Advertisement.” Similarly, some subjects, while working on task 2, the “Business Card,” used work-arounds to change the logo. In other cases, some subjects were not sure that a function existed in the dialog box or toolbar that they were using; this led them to cancel out of an appropriate dialog box or to hover away from an appropriate toolbar. Through interplay of
exploratory learning from T&E episodes and directed learning from help episodes, users might have a
greater chance of filling in these gaps in their mental models, such as distinguishing images from text
boxes, or grasping that some images are grouped.
Chapter 6: Conclusion

Delivering tutorials for computer applications through help-based rather than traditional methods appears to be an effective means of leading users of these applications to rely on greater use of help systems. A between-subjects experiment, using a balanced design with 22 participants, confirmed that a help-based tutorial leads to significantly greater use of help but failed to confirm that this greater use of help led to better task performance. Use of help by participants in both the traditional tutorial and the help-based tutorial conditions tended to decrease their use of help over the four tasks in the experiment. I observed that participants in both conditions also often tended to rely on trial-and-error methods and work-arounds rather than looking for and using more specifically appropriate methods through the help system.

When using the help system, study participants often lacked sufficient knowledge of key words to be able to use the help system effectively. When participants did reach an appropriate topic, they were often unsure that they had found what they needed. As a result, participants tended to skim the help topics and thus sometimes did not learn what they needed even if they reached the right topic.

The complexity of modern computer applications, such as the Microsoft Publisher application used in this study, appears to lead to under-use of help in at least two ways. First, users appear to be confused by access to both local and Web-based help, which may cause them to be less eager to try using either path. Second, users may think that the muddle-through method they found by trial and error is, in fact, the right way to solve the problem, which may cause them not to look for help to find a better way. Indeed, given the difficulties we observed in using help, their decision not to use help may be a rational one.

The subjects’ experiences in my study suggest that, from the subjects’ perspective, more successful outcomes occur through the T&E approach than by consulting help, although consulting help does lead to some outright successes and can lead to eventual success in a later T&E episode. Our analysis also indicates that subjects are able to achieve successful task outcomes by switching between help and T&E approaches. These findings have implications for the design of interfaces and help systems, and especially for the interaction between the two.
Through the rest of this chapter, I will review the limitations of the study as they relate to the results I have discussed in the previous sections. I will also present ideas for how to support user preferences for troubleshooting applications (namely, switching between help and trial and error) and discuss ways to address user concerns about help contents being at the “wrong level” by providing explicit support for dynamic content adjustment.

6.1 LIMITATIONS

While the study’s results for Hypothesis 1 were positive, the study suffered from a number of limitations that might be addressed in future work. These limitations included lack of calibration of the tasks, effects of differences in experimental instruction, and incomplete video records of the tutorial phase of the experiment sessions.

As discussed in Section 5.3, the negative result for Hypothesis 2 may be in part caused by not having calibrated the four tasks in advance of the experiment. Task 2, as intended, appeared to be more difficult than task 1. But in the bright, retrospective light of having conducted the experiment, as indicated in Table 3, task 4 appears to have been as easy or easier than the other three tasks, when it should have been more difficult because it was intended to require knowledge not covered in the tutorial. This lack of calibration did not affect Hypothesis 1, which did not depend on differences among the tasks. But for Hypothesis 2, the lack of calibration possibly affected the results. We note, though, that the remarkable similarity of average task performance across the help-based and traditional tutorial condition (3.81 vs. 3.82 out of 5) suggests that even with calibrated tasks it is unlikely that Hypothesis 2 would have been confirmed.

A second limitation involves the instructions given to the participants, particularly for those using the help-based tutorial. For the participants in the early part of the study, the researchers’ instructions for the help-based tutorial were non-specific, such as read the tutorial and complete the task. These participants, on average, used help about three times more than the traditional tutorial participants. This effect increased for later participants as I was more explicit in the instructions about learning to use the application by using the help system. While my data confirmed Hypothesis 1 with acceptable significance, I expect that both the effect size and level of significance would have been
increased had all participants using the help-based tutorial received the more directive version of the instructions.

When I designed the study, I had focused on collecting the data that would confirm (or, as I expected, not confirm) the two hypotheses. These data would be produced through analysis of the task sessions. Thus I recorded the task sessions but did not record the participants’ use of the tutorials. As I observed the sessions and began to compare observations, I began to appreciate that the participants’ behaviors in the tutorial phase were interesting in themselves, even if they did not bear directly on the formal hypotheses. As a result, I was able to record the tutorial phase for our final six subjects, three in each condition. The qualitative results in Section 5.4 examining participants’ interactions with the tutorial are thus based on recordings of a subset of the participants, augmented by the direct observations of the researchers from the sessions. These results would have been more reliable had we recorded all of the tutorial-phase sessions.

For the reasons discussed, none of these limitations is likely to have led to erroneous confirmation of Hypothesis 1 or erroneous failure to confirm Hypothesis 2. Indeed, the limitations may have led to understating the effect size and significance for Hypothesis 1. The principal effect of the limitations, particularly with respect to the recordings of the tutorial-phase sessions, was to reduce the comprehensiveness of my qualitative analysis. For future work, I intend to also estimate the standard deviation for the results obtained in tables 4.1 and 4.2.

Another limitation is that the four tasks were designed to test a variety of skills, particularly as we sought to have some tasks covered by the tutorials and other tasks not covered. Thus the functions in the tasks differ in ways that led to unanticipated behaviors on the part of the subjects, such as the greater number of work-arounds in task 3. Additionally, although the steps in each task were intended to be relatively simple, many subjects were unable to complete the tasks in the time available. In future work, we would pre-test the tasks to determine that the tasks were all capable of being completed within the allowed time and we might schedule longer sessions, if the subjects were amenable. We would also check that the tasks required reasonably equivalent skills, although this is a difficult constraint precisely because each task is supposed to require new skills.
Task order is another limitation of the study as already mentioned; because all tasks were given in the same order to all participants, I argue that permuting their order may have an effect on the subject’s performance – especially because the last two tasks used similar features – and also on the survey responses from the subjects.

One more limitation involves the methodology of the sessions. The original study focused on task completion times and task outcomes, so we tried to make the sessions as realistic as possible. This meant, though, that our interpretations of the subjects’ mental states was based, other than occasional comments they made to the experimenter, on the subjects’ actions in the interface.

In future work, I would use a think-aloud methodology that would provide a more explicit indication of what the subjects were thinking.

6.2 SUPPORTING PROBLEM-SOLVING

The most important lesson to draw from the study’s results is that there is an opportunity to make computer applications that can support more effective problem-solving by providing features that enable users to switch more effectively between exploratory learning in the application’s interface and consulting the application’s help system. Help systems might support switching between problem-solving methods in these ways:

- **Automatic History.** Help systems could provide a history of the user’s interaction, visible by default, that provides quick access back to searches, help topics, and particular sections of help topics previously visited.

- **User-Managed History.** Help systems could also support switching by providing “save search,” “save topic,” “save help section” and similar features that enable users to actively manage a list of helpful topics or leave signposts as to where they left off when learning about application functions.

- **User Annotation of Help.** Help systems could include facilities for annotation so that users could append their own understanding to the topics they search, such as the meanings of unfamiliar terms.

- **Shared Annotations.** If help systems support user annotation, then these annotations could be shared among users, providing much more direct access to the sort of peer-user
information for which users now search the Internet. Access to shared annotations could be controlled by users and organizations.

- **Links to Application.** Help systems could provide hyperlinks from help text back to the application interface. This could help novice users who are confused between, for example, a menu item and a toolbar with similar names.

In addition to supporting switching between problem-solving approaches through better help systems, computer applications could be designed with the understanding that most learning, especially for novices, will occur through exploratory learning rather than formal consultation of help. These adaptations could include:

- **Tool Tips.** Application interfaces could provide more extensive roll-over help. This might include tool tips, which are more meaningful than function names that are shown after an extended hover. Microsoft’s “SuperTooltip” [Elsehemy 2007] is a good step in this direction. Providing this kind of just-in-time, right-in-context help should generate much of the benefit of switching problem-solving approaches without the disruptive overhead of invoking and navigating the help system. This approach might guide users to avoid false affordances [Gaver 1991], such as reaching Publisher’s “Design Gallery” when they intended to find a gallery of clip art.

- **Context-Aware Interface.** If the user has just consulted help, the application interface could reinforce the affordances—the cues and signposts—that would guide the user to the appropriate part of the interface. For example, the application could show or highlight a toolbar, just referenced in help, to assist the user in finding the toolbar rather than mistakenly reaching another part of the interface that has similar terminology.

- **Context-Aware Help.** Conversely, if the help system were aware of the user’s interaction in the application interface, the help system could do a better job of suggesting help topics. For example, if a user was working with a text box and searched in the help system for “add border,” the help system would begin with help relevant to text boxes rather than, say, tables.
Based on our analysis of the difficulties our subjects encountered and the problem-solving methods they employed, I have these further suggestions for writers of help systems:

- Put the procedure early in help topics. Subjects in our study would consult help, scan for a few keywords and then try to use a feature without reading the entire topic. Help systems could address this by putting the procedure first and the explanation, for those who want it, second. The most common scenarios for the use of a feature can be listed at the beginning, and other less likely scenarios could follow. The direction of the user’s attention to the procedural parts of the help could be made more direct by using headers that explicitly identify the components of the why-what-how model [Novick 2000] of help, placing the how section first.

- Add guidance for common tasks to help systems. This would be a counterpart to the familiar FAQ: frequently performed tasks. While it would likely be impossible to predict the entire set of tasks that users would want to perform, developers of computer applications could collect data from which the most common tasks, especially for novices, could be identified. Help systems could highlight these tasks, and application interfaces could provide more evident affordances for their accomplishment.

- Help users build their mental models of the application. The application and the help system could teach users to distinguish among the application’s basic kinds of components and functions that relate to these components. Techniques could include rollovers in the help system, even in lists of help topics generated by search, that showed pictures or other simple explanations of the components to which the help relates. For example, a help topic about borders for tables could have a rollover that shows examples of borders and a roll-over that shows an image of table. My analysis further suggests that trying to build the users’ mental models of the application through the help system topics themselves is less likely to be effective. First, as the research shows, users tend to consult help for pragmatic rather than didactic reasons. Second, if, as I suggested earlier, help topics should have the procedure early, then it would be difficult to put the model before
the procedure. One possible solution would be to preface the procedures with minimal explanations of the appropriate contexts of their use.

- Study the expectations and vocabulary of novice users. People come to the application with a vocabulary and semantics born of their experiences in a world other than that of a particular computer application. Designers and writers could alleviate problems, particularly for novice users, by studying what people expect from an application and how they refer to things and concepts and associated with the application. At a minimum, this could be useful in providing indexes of synonyms that could be offered by the help system as pointers, with appropriate explanation, to topics and functions that have the application’s own vocabulary and meaning.

Developers of computer applications and writers of the documentation for these systems can suffer understandable frustration when they see users in usability tests muddling through instead of following the obvious affordances of the application’s design or scanning help text and returning to the application instead of reading and following the clear procedure provided by the help system. The subjects’ behaviors in this study, like the subjects’ behaviors in previous studies that show low levels of consulting documentation and help systems, can be seen for what they are: pragmatic actions of people using exploratory learning and actively switching among problem-solving approaches. Developers and writers can help the users of their applications and documentation by understanding, accepting and supporting the users’ actual, rather than ideal, approaches to accomplishing tasks in computer applications.

Another proposal is to rewrite help to accommodate various levels of user experience across the Tau dimensions so that these contents may be implemented in a multi-layer interface that users can use to adapt such contents dynamically to find help with a level of technical detail that matches the individual level of understanding of each user. Such implementation would require that the contents are grouped into these dimensions and the level of detail for each category is different for novice, intermediate and expert users.
6.3 **Rewriting Help for Different Tau Levels**

A second possible solution to help system design is to enable users to tune the level of technical detail that is offered in help topics to their particular level of experience. Help can be written across the different Tau dimensions according to Kearsley’s model [1988] and across multiple levels of experience for each dimension, as defined earlier, so that users can choose the level of detail that is displayed in a multi-layer interface.

Implementing a help system in a multi-layer interface involves adapting the contents of the help system across the Tau (τ) dimensions of experience. Given the τ model, though, it was not clear that this could be accomplished in practice. Our challenge involved choosing which contents corresponded to each of the “why-what-how” and τ dimensions, and how these contents could vary across multiple levels of experience. If this were done appropriately, the organization and presentation of help across multiple dimensions and levels of experience could enable users to find the information and use they needed to perform a task with a computer application. This mapping proved to be non-obvious. Moreover, as noted earlier, it remains unclear if the why-what-how dimensions are actually distinct from the τ dimensions.

To explore this issue, I rewrote parts of the contents of an existing commercial help system to adapt them across multiple levels. My work involved determining which parts of the help topic corresponded to the why-what-how dimensions, and whether I could produce coherent variations of a help topic for novice, intermediate, and expert users. This involved providing τ levels for each topic and its contents.

I developed multiple versions of parts of the help system for Microsoft Excel as a sample for a future study, where these versions corresponded to combinations of different parameters for τ. In choosing Excel, I sought an application for which we could reasonably expect to find populations of users who varied significantly across the τ dimensions. In contrast, in this related study, I needed subjects who had little or no experience with a particular computer application, and for that experiment I used Microsoft Publisher. As we plan to test the τ model as embodied in a user-settable help system, I am interested in having variable levels of experience. Other studies suggest that levels of experience with Microsoft Excel vary among users [Novick 2006a], [Novick 2007], so this application should have future study participants who can represent all levels of experience.
To develop examples of help expressed with different combinations of \( \tau \) parameters, I rewrote Excel’s help topic on copying and pasting multiple items across Microsoft Office applications using the clipboard, and produced three different versions of this topic: \( \tau(1,1,1) \), \( \tau(2,2,2) \), and \( \tau(3,3,3) \). While I have yet to assign absolute values to the \( \tau \) dimensions for particular topics, I was able to rewrite three help topics that can be rated with respect to each other. That is, some features seem less advanced than others. For example, copy and paste appears to be less advanced than copying and pasting multiple items across programs using the Microsoft Office clipboard. The help topics by themselves, in the absence of a specific task, proved unsuitable for rating in the task dimension.

As I analyzed the three Excel 2003 help topics, I found that the why-what-how dimensions were not clearly identifiable. Thus I had to write some additional contents for these topics, especially the “what” component. Interestingly, similar content was added in Excel 2007 help.

The main issue with both versions of Excel help is that all help topics contained mixed levels of explanation because technical terms, icons, and other explanations for each feature were included in the same paragraph or even within the same sentence. A partial example is shown in Figure 6.1.

![In the following:](image)

- To move cells, click Cut on the Standard toolbar (toolbar: A bar with buttons and options that you use to carry out commands. To display a toolbar, press ALT and then SHIFT+F10, or press CTRL+X.
- To copy cells, click Copy on the Standard toolbar, or press CTRL+C.

The upper-left cell of the paste area (paste area: The target destination for data that’s been cut or copied by using the clipboard).

Figure 6.1: Explanations for all levels of experience in the same sentence.

I rewrote the copy and paste among programs topic for each level. The \( \tau(1,1,1) \) explanation of the topic in Figure 6.2, aimed at novice computer users who are new to Excel, uses simple vocabulary and visual guidance on how to carry out each step, along with an explanation for what the feature is, why use it, and how to use it.
This feature enables you to copy different information such as text and pictures into the Clipboard so you can paste them onto a document in a different Microsoft Office program. As an example, you may want to copy text from a Word document to a brochure you created in Publisher, or a pie chart from Excel to PowerPoint for a presentation.

The Clipboard contains the different items you have copied and shows you which program you copied them from.

You may use this feature if you wish to swap text, images, and other information among Microsoft Office programs without having to use single Copy-paste commands for each text or picture item.

How to copy and paste information among programs:

1. Open the Office Clipboard from the Edit Menu.

Figure 6.2: A novice-level explanation on how to copy and paste multiple items with the clipboard.

A $\tau(2,2,2)$ explanation of the topic, aimed at intermediate users of computers who have intermediate experience with Excel, still reminds users what the clipboard is, but omits pictures and assumes the user knows how to find the menus that are outlined in the topic. This version of the topic is shown in Figure 6.3. Finally, the $\tau(t,3,3)$ expert-level explanation does not use any graphical guidance, uses more technical terms, and even suggests keyboard shortcuts so that users can copy and paste more fluently. This version of the topic is shown in Figure 6.4. These different levels of experience vary only with respect to application and user experience together. We have yet to determine how to vary help by task or in other $\tau$ combinations such as $\tau(1,3,2)$ or $\tau(3,1,2)$. If these further combinations prove to add little benefit or to be impractical, this would suggest that the $\tau$ model could be simplified by reducing the number of dimensions [Andrade and Novick 2008].
The τ model described a set of dimensions in which I was able to vary help relative to the why-what-how dimensions in which I categorized its contents. My goal was to vary the level of explanation in help to provide a suitable representation for users with different skill levels.
The τ model could be embodied in a help system for a particular application. Figure 6.5 shows a design concept for such a help system, which combines both the multi-layer and the multi-dimensional designs. Each τ level represents one layer for each help topic, and users can customize the level of explanation in each help topic by moving the slider bar to the right to increase the level of explanation or to the left to decrease the level of explanation.

![Figure 6.5: A proposal for a multi-layer help system.](image)

The main contents in the display are categorized in the why-what-how dimensions, which can be toggled on or off with the exception of the how dimension, which is always displayed by default. Such contents will vary along different levels as the slider bars are moved sideways by users. Whether the why-what-how dimensions can be expressed in the τ dimensions is another open research topic. The proposed design for a help system incorporates a multi-layer, multi-dimensional interface through which the user can dynamically specify how the documentation can be presented at different levels. My design concept provides check boxes through which the user can select which parts of a help system to view.
and three slider bars through which the user can change the level of the displayed content [Andrade and Novick 2008].

Many researchers have addressed issues of adaptive help (see, e.g., [Fischer 2001], [Carroll 1984]; a full discussion is beyond the scope of this paper), but they focused on automatically adapting the content of help systems through artificial intelligence techniques. My work differs from this line of research in three key ways, in that it:

1. Focuses on enabling the user to set appropriate levels of explanation her- or himself, rather than trying to build a dynamic mental model of the user through artificial intelligence.
2. Involves choices of presentation as well as of content.
3. Explores in practical terms what it means to produce written documentation at different levels of multiple dimensions of user expertise.

This interface for help systems represents a potential solution to users’ common complaint that help is at the “wrong level.” Dynamic, user-settable levels could enable users to customize the level of explanation through slider bars in a multi-layer, multi-dimensional interface. My exploration of producing help at different levels suggests that doing so is possible but that general rules remain unknown.

Given that the conceptual design in Figure 6.5 presents what I see as a reasonably intuitive interface (without considering the why-what-how dimensions, which are separate choices), and given that I was able to develop versions of help topics for Excel that illustrated points in the space defined by \( \tau \), it may be possible to develop an interface for help that enables users to find their right level of help by adjusting the sliders for the \( \tau \) dimensions.

Accordingly, I see my future work as addressing the following issues:

- Understanding how users’ \( \tau \) levels translate into more useful help. We plan to ask users to rate help by identifying which parts of help are at right and wrong levels of explanation for them and explain why.
- Defining specific examples for the task dimension; I generalized the task to “using spreadsheets” in Figure 6.5 because the range of domain tasks that can be done with a
particular application is rather large. Domain tasks that can be done with an application include creating a company brochure, personal business card or promotional poster for users of Publisher and calculating a business division’s budget and profits for a fiscal year in case of Excel users.

- Rating and tagging additional help topics according to each of the τ dimensions and levels. This includes determining which contents of help are related to the task dimension.
- Deciding how many levels of experience should be used to organize help. For the moment, we have three levels, but perhaps we can produce variations of help contents between the novice, intermediate, and expert levels.
- Understanding whether the why-what-how dimensions can be expressed in the τ dimensions.
- Determining whether users are, in practice, able to use controls like those in the interface in Figure 9 to tune help to reflect the levels of help they need.
- Looking into additional models that may be compatible with τ, or which can accommodate other dimensions to provide a more usable representation of online help.
References


Appendix A: Tasks

Figure A.1: “Arrow sign” reference design for task 1.

Figure A.2: “Business card” reference design for task 2.
Golden apples taste better than red and green apples.

Perhaps you should try them all and find out for yourself.

Figure A.3: “Golden apple advertisement” reference design for task 3.
Figure A.4: “Raise petition” reference design for task 4.
# Appendix B: Post-Session Survey

For these questions, please circle how much you agree/disagree with each statement.

1. **I enjoy working on computers.**
   - I strongly agree.
   - I agree.
   - I slightly agree.
   - I slightly disagree.
   - I disagree.
   - I strongly disagree.

2. **I often need other people to help me when I use a computer.**
   - I strongly agree.
   - I agree.
   - I slightly agree.
   - I slightly disagree.
   - I disagree.
   - I strongly disagree.

3. **I feel comfortable learning new computer programs.**
   - I strongly agree.
   - I agree.
   - I slightly agree.
   - I slightly disagree.
   - I disagree.
   - I strongly disagree.

4. **I am a better computer user than the average person.**
   - I strongly agree.
   - I agree.
   - I slightly agree.
   - I slightly disagree.
   - I disagree.
   - I strongly disagree.

5. **It is a good use of my time to learn more about the computer programs I use.**
   - I strongly agree.
   - I agree.
   - I slightly agree.
   - I slightly disagree.
   - I disagree.
   - I strongly disagree.

6. **It is a good use of my time to learn more about new computer programs.**
   - I strongly agree.
   - I agree.
   - I slightly agree.
   - I slightly disagree.
   - I disagree.
   - I strongly disagree.
Curriculum Vita

Oscar Daniel Andrade Gonzalez was born in El Paso, Texas on September 2, 1982 and raised by Martha Gonzalez, an architect and urban planner, in Cd. Juárez, Mexico where he attended school from kindergarten to high school, from which he graduated in 2000 to pursue a bachelor’s degree in computer science at the University of Texas at El Paso (UTEP).

He earned his bachelor’s degree in computer science from UTEP in 2006 and learned about human-computer interaction (usability in particular) as the field he wanted to pursue as a graduate student and professionally. He then decided to earn a master’s degree to learn to do research within the field and prepare for a career in usability.

Oscar’s research focused on the usability of application help systems; he has published two papers as a main author and co-authored two papers with Dr. Novick, his thesis advisor, and fellow student Nathaniel Bean, all of which have been presented at conferences for the Association for Computing Machinery Special Interest Group on the Design of Communication (ACM SIGDOC). Oscar currently works for International Business Machines (IBM) as a software developer in Lexington, Massachusetts, where he helps with the design and implementation of user interface components for products that are based on the Eclipse platform.

Permanent address: María Curie, No. 228
Cd. Juárez, Chihuahua, México, 32500

This thesis was typed by Oscar D. Andrade.