The new face of procedural content: A real world approach

Connie Douglass Olberg
Microsoft Corporation
conniedo@microsoft.com

David Farkas
University of Washington
Professor
farkas@u.washington.edu

Joyce Karreman
University of Twente
Assistant Professor of Technical Communication
j.karreman@utwente.nl

David Kieras
EECS Department
University of Michigan
Professor
kieras@eecs.umich.edu

Axel Roesler
University of Washington
Assistant Professor for Interaction Design
roesler@u.washington.edu

Michaël Steehouder
University of Twente
Chair of Technical Communication
m.f.steehouder@utwente.nl

Ryan Baker
Microsoft Corporation
ryanba@microsoft.com

David Brunet
Microsoft Corporation
davbru@microsoft.com

Ninad Dalal
Microsoft Corporation
a-nidala@microsoft.com

William Keener
Microsoft Corporation
williamk@microsoft.com

Copyright is held by the author/owner(s).

CHI 2008, April 5 – April 10, 2008, Florence, Italy
ACM 1-xxxxxxxxxxxxxxxxxx.

Abstract

This paper describes a project that uses real-world computer-troubleshooting scenarios to test and expand upon the latest research and technical knowledge in the area of procedural content development. Our goal was to improve the effectiveness of on-line help procedures for solving computer problems experienced by novice to intermediate computer users. Our focus was not just on writing techniques. We also focused on presentation aspects such as the structure of the content, use of graphics and screenshots. The outcome of the project is to develop procedural writing and presentation guidelines and to apply them to developing future support content, which we believe will lead to increased success of the target user. Prototypes based on the guidelines were usability tested with one prototype released on our public site. These tests showed positive results.

Keywords

User assistance, procedures, troubleshooting, technical writing.

ACM Classification Keywords

H.5.2 User Interfaces— training, help, and documentation.

Introduction

Novice computer users are easily frustrated by computer problems. Their low domain knowledge and their lack of a developed troubleshooting mental model present tremendous obstacles to their success when they try to use procedural content to solve even the simplest computer problems. The support site with which we are working receives an average of 30 million visitors per month; 24 million of those visitors represent computer users with skill levels ranging from novice to intermediate. However, the writing and presentation of procedural content on this support site did not adequately address the site’s primary visitors. Therefore, our project set out to define and develop content writing and presentation guidelines, that when applied, would enable our target users to be more successful with self-solving their computer problems.

The project's approach was designed to leverage existing industry and academic expertise and research that we could apply to creating new forms of content, which we could then test and iterate upon. The goal was to produce guidelines for writing and presentation that development teams can use in future content publishing.

Although this project is still in progress, the project approach, specific concepts and examples, and preliminary results are described in this paper.

Project Approach

We approached the project systematically using the following discipline:

Defining the content problem

This research phase consisted of the following:

- Primary Research: We reviewed customer and support engineer ethnographic interviews and conducted focus groups. A comparative analysis of online support content was conducted in this phase to understand how the industry is addressing this problem space.
- Secondary Research: We conducted a literature review of current theories and best practices related to procedural content. See the citations list for publications that we found most useful.
- We identified experts in the field of technical communication who subsequently participated as consultants throughout the life of the project.
- We conducted usability sessions that established baseline data of current content that addresses a preselected set of computer problems typical of our target audience. We used these preselected problem scenarios throughout the project.

The result of this phase was the development of a project plan, focusing on research and testing the identified areas of improvement.

Diagnosing the problem

This phase involved analyzing and diagnosing the problem space to be addressed. The primary areas of concern with the content included the following issues: 1) Novice users had difficulty finding the right content that matched their understanding of their problem. 2) Content was written using a technical voice novice users did not understand. 3) Content was presented in a linear intimidating format that was hard to follow. 4) Steps lacked detailed step-by-step guidance, were difficult to execute, and assumed prior technical knowledge. 5) Content lacked verification steps and next steps.

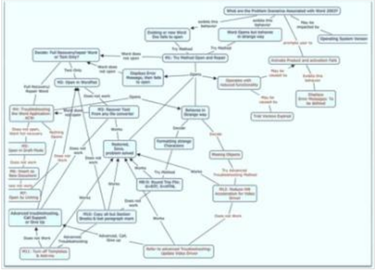


Figure 1: Example of task analysis concept map, showing detailed branching and workflow of problems and solutions.

Developing the solution construct
This phase identified the important concepts and best practices that should be applied to the content. This stage also involved the design for the usability test. The test was designed as a qualitative formative usability study using a think-aloud protocol. The goal was not to isolate and test specific variables in our solution; instead, the goal was to combine and develop a hybrid collection of previously researched techniques that could be applied to real-world procedural content. The phase yielded preliminary guidelines for problem and solution task analysis, writing, and presenting content.

Developing the prototypes
We developed prototypes with three new visual designs. Each prototype incorporated the content elements and user task flow that had been developed from task analysis and writing. Three versions were designed: (1) a linear model, which displays all the content at once; (2) a wizard-like interactive model, where content is filtered based on choices made by the user in the workflow; and (3) a paned model, which compresses content into a window on the left side of the screen, so that users can interact with the program on the right side of the screen. The result of this phase was prototypes ready for usability studies.

Testing and prototype iteration
The prototypes developed in the previous phase were tested in usability settings using the previously defined Design of Experiment. Six usability tests are currently in progress with each test producing new ideas for updating the guidelines and prototypes. The planned outcome of this phase is final guidelines and prototypes.

Guidelines

The insights from the above project approach crystallized in a set of guidelines, intended for the writers and designers of new procedural instructions on our web site. The following guidelines were established:

Task Analysis
Procedural content must be accurate and complete. If it is not, users may lose confidence, may become blocked and unable to continue, or may experience other problems. Ensuring accurate and complete content requires us to accurately define all possible customer scenarios for the problem and the troubleshooting flows that result.

This task is carried out by a team of product experts representing both the customer experience and the product. The team employs concept mapping *(Figure 1)* to create a visual representation of the end user's workflow in successfully solving the problem. The concept maps determine method priority and other dynamics that may require branching the user seamlessly to a new set of procedures. In addition, Goals, Operators, Methods, and Selection rule (GOMS) principles are applied to define the methods and procedures in detail. This detailed workflow is then handed off to the writers who are then able to create accurate KB articles

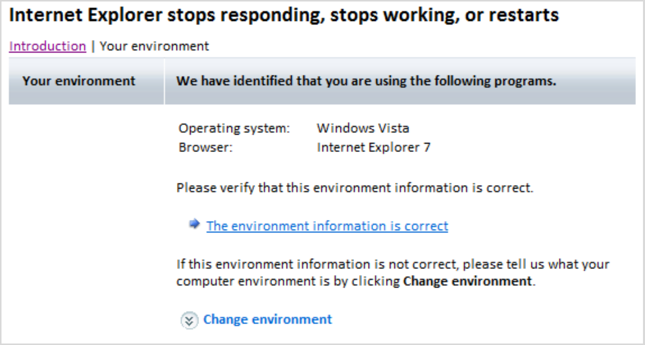
Writing
The writing guidelines separated the procedural content into individual components that were then categorized as general declarative text, procedural declarative text, and procedures. General declarative text includes declarative text outside of the actual methods/steps

such as content title or introduction of procedural content. Procedural declarative text includes declarative text components within the actual methods such as introductions, notes, warnings, or error deflection text. Finally, procedural text forms the actual “steps” of the methods. Our writing guidelines were implemented as a help (.CHM) file and follow a tutorial format, segmenting the guidelines into conceptual model introduction, concepts, and then writing guidelines for each content component.

Presentation and Media style guides and prototypes
These style guides detail the technical parameters for development and design details for creating working knowledge base articles. The final prototypes represent the interactive *(Figure 2)* and side pane presentations *(Figure 6)* that produced the highest success in usability. The prototypes are intended to be used as content and presentation examples by writers and developers.

Writing and Presentation Concept Implementation

The detailed task analysis surfaced content workflow components that were not straightforward. For example, prerequisites are required to determine whether the user is in the right content. Prerequisites (*Figure 2*) could include items such as product version validation, computer administrator privileges, or install disks—each of these is a critical criterion that must be incorporated into the flow of the content.



Internet Explorer stops responding, stops working, or restarts

Introduction | Your environment

Your environment | **We have identified that you are using the following programs.**

Operating system: Windows Vista
Browser: Internet Explorer 7

Please verify that this environment information is correct.

The environment information is correct

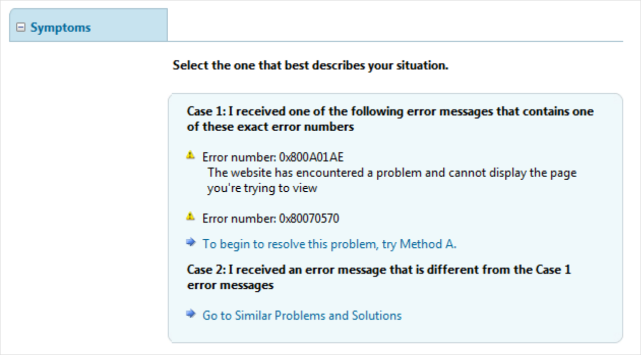
If this environment information is not correct, please tell us what your computer environment is by clicking **Change environment**.

Change environment

Figure 2. “Prerequisite” example to verify the product version of the user’s computer. The Web content automatically determines product versioning. All possibilities are addressed through task analysis and represented in the content.

Troubleshooting computer problems is not straightforward—our job is to minimize the technical burden of low-domain computer users and help them make the required decisions to enable their success using a friendly, non-intimidating format. In the symptoms section, we present various options. (*Figure 3)* The options must be communicated in a straightforward manner that disambiguates the selection process. The user is directly guided to the correct method based on their symptom selection.

The methods and procedures apply system and product declarative text to establish context for the user and to instill confidence that they should not only try the method but also that the method is presented in a manner that low-domain users can execute successfully.



Select the one that best describes your situation.

Case 1: I received one of the following error messages that contains one of these exact error numbers

⚠ Error number: 0x800A01AE
The website has encountered a problem and cannot display the page you're trying to view

⚠ Error number: 0x80070570

➜ To begin to resolve this problem, try Method A.

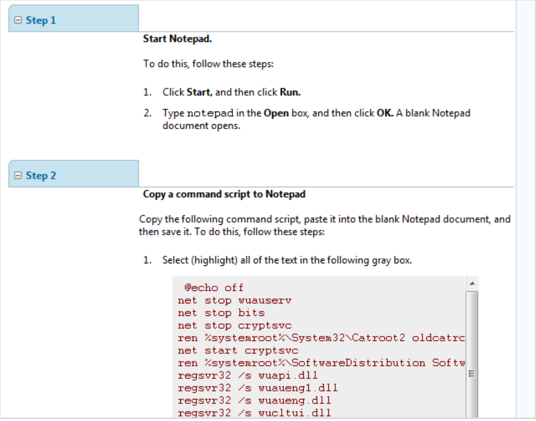
Case 2: I received an error message that is different from the Case 1 error messages

➜ Go to Similar Problems and Solutions

Figure 3. Symptom verification example.

Figure 4 presents a technically challenging command prompt procedure. Yet, 100% of the users successfully completed the method in our usability testing. The following techniques were applied:

- Chunking procedures into goals and sub-goals.
- Procedures are concrete: verb (action) + noun (object) =resultant system feedback.



Start Notepad.

To do this, follow these steps:

1. Click **Start**, and then click **Run.**
2. Type `notepad` in the **Open** box, and then click **OK.** A blank Notepad document opens.

Copy a command script to Notepad

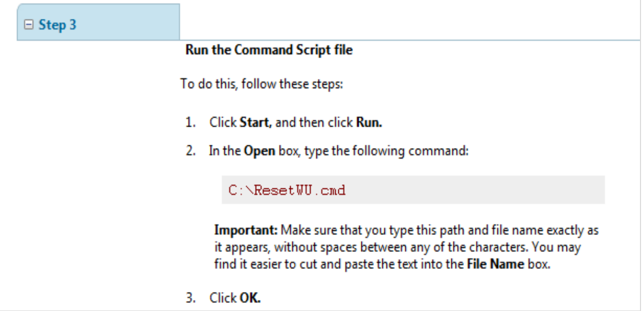
Copy the following command script, paste it into the blank Notepad document, and then save it. To do this, follow these steps:

1. Select (highlight) all of the text in the following gray box.

```
 @echo off
net stop wuauserv
net stop bits
net stop cryptsvc
ren %systemroot%\System32\Catroot2 oldcatrc
net start cryptsvc
ren %systemroot%\SoftwareDistribution Softw
regsvr32 /s wuapi.dll
regsvr32 /s wuaueng1.dll
regsvr32 /s wuaueng.dll
regsvr32 /s wucltui.dll
```

Figure 4. Chunking steps with procedures in concise format.

- Error deflection: Incorporated as a written procedure or added as a note directly below the error prone procedure (*Figure 5*).
- Confidence building statements: Incorporating encouraging text in method and step introductions motivates the user to try a method.



Run the Command Script file

To do this, follow these steps:

1. Click **Start**, and then click **Run.**
2. In the **Open** box, type the following command:

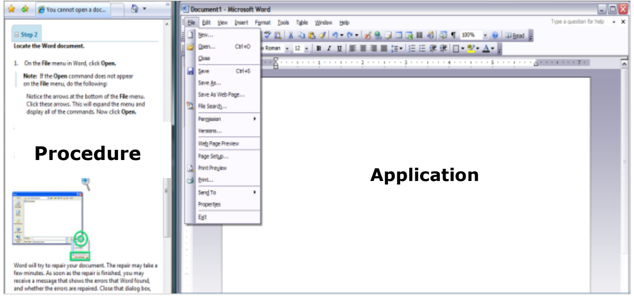
```
C:\ResetWU.cmd
```

Important: Make sure that you type this path and file name exactly as it appears, without spaces between any of the characters. You may find it easier to cut and paste the text into the **File Name** box.

3. Click **OK.**

Figure 5. Error deflection example incorporated as a note below the procedure.

Switching Effects: We applied a presentation we called side-pane help (*Figure 6*) to address switching effects when users diverted attention from the instructions to the application to execute the procedures. Low-domain users had difficulty retaining procedures in short term memory. This caused errors in procedure execution.



Locate the Word document.

1. On the **File** menu in Word, click **Open.**

Note: If the **Open** command does not appear on the **File** menu, do the following:

Notice the arrows at the bottom of the **File** menu. Click these arrows. This will expand the menu and display all of the commands. Now click **Open.**

Word will try to repair your document. The repair may take a few minutes. As soon as the repair is finished, you may receive a message that shows the errors that Word found, and whether the errors are repaired. Close that dialog box,

Figure 6. Instructions aligned to left, application to right to address switching effects.

Preliminary Results

Usability testing produced significant success improvement from the baseline results ranging from 10% to 20%. A pilot content released on the support site to the public that applied preliminary principles and guidelines produced a staggering solve rate increase from baseline solve rate of 9% to increased solve rate of 53%.

Acknowledgements

We thank the Support Engineers, their management and the content business managers for their content development help and contributions.

Citations

[1] Farkas, David K. 1999. The logical and rhetorical construction of procedural discourse. Technical Communication 46: 42-55.

[2] Karreman, Joyce, Nicole Ummelen, and Michaël Steehouder. 2005. Procedural and declarative information in user instructions: What we do and don't know about these information types. In Professional Communications Conference, 2005, Proceedings, 328-333.

[3] Steehouder, Michaël, and van der Meij, Hans. 2005. Designing and evaluating procedural instructions with the four components model. In Professional Communication Conference, 2005, Proceedings, 797-801. New York, NY: Institute of Electrical and Electronics Engineers.

[4] Loorbach, Nicole, Karreman, Joyce, and Michaël Steehouder. 2007. Adding motivational elements to an instruction manual for seniors: effects in usability and motivation. Technical Communication, 54: 343-358.

[5] Ummelen, Nicole. 1997. Procedural and declarative information in software manuals: Effects on information use, task performance and knowledge. Atlanta, GA: Rodopi.

[6] Kieras, D.E. 1996 A Guide to GOMS Model Usability Evaluation using NGOMSL. The Handbook of Human-Computer Interaction, cosc.brocku.ca

[7] Gellevij, Mark, and Hans van der Meij. 1999. The effects of screen captures in manuals: A textual and two visual manuals compared. Professional Communication, IEEE Transactions on 42: 77-91.

[8] Lazonder, A.W., Van Der Meij, H. 1995. Error-information in tutorial documentation: supporting users' errors to facilitate initial skill learning. International Journal of Human-Computer Studies. Volume 42, Issue 2: 185-206

[9] Novak, J.D., Cañas, A.J. 2006. The Theory Underlying Concept Maps and How to Construct Them. Technical Report IHMC Cmap Tools

[10] Karreman, J. 2004. Use and effect of declarative information in user instructions. Amsterdam – New York, NY: Rodopi.