Technical Writing

Technical Writing

- User Documentation (HCI)(this course!)
  - printed manuals
  - on-line help
  - tutorials
- Programmer documentation (software eng)
  - code comments
  - requirements specification
  - design document
  - testing report

Powerpoint documentation

- Help Menu (F1)
  - » Table of contents, search
  - » Office assistant
- Help within screens
  - » ? Button on dialogues
- Wizards (eg, New Presentation, AutoContent)
  - » Not really documentation, but there to help the user use and understand the system
- Web site

Design Decisions

- Organisation: task or functional
- Appropriate level of detail
- Writing style
- Visual appearance
- Include examples or sample sessions?
- How to test?

Example: Copy instructions

- Compare
  - » Windows help
  - » Win help for MSDOS
  - » Unix help

Windows help

To copy a file or folder

1. Open My Documents

   If the file or folder you want to copy is not located in My Documents or its subfolders, use Search to find it. To open Search, click **Start**, and then click **Search**.

2. Click the file or folder you want to copy.

3. Under **File and Folder Tasks**, click **Copy this file** or **Copy this folder**.

4. In **Copy Items**, select the drive or folder you want to copy to, and then click **Copy**.

Note

- To open My Documents, click **Start**, and then click **My Documents**.

- You can copy more than one file or folder at a time.

- To select consecutive files or folders, click the first item, press and hold down SHIFT, and then click the last item. To select nonconsecutive files or folders, press and hold down CTRL, and then click each item.

Related Topics

- Move a file or folder
- Move files by dragging
- Files and folders overview

Windows help

- Task organisation (how to copy)
  - » Just one way of doing this task!
- Detail – none beyond what needed
- Writing – simple (primary school level)
  - » User-oriented (“you”, or “click”, not “user”)
- Layout – fonts, hyperlinks, bullet lists
- Examples – none (?)

Good model for most user documentation

MSDOS Help

```
Copies one or more files to another location.

COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/A | /B ] source [/A | /B]
     [+ source [/A | /B] [+ ...]] [destination [/A | /B]]

  source        Specifies the file or files to be copied.
  /A            Indicates an ASCII text file.
  /B            Indicates a binary file.
  /D            Allow the destination file to be created decrypted
  destination   Specifies the directory and/or filename for the new file(s).
  /V            Verifies that new files are written correctly.
  /N            Uses short filename, if available, when copying a file with a non-8dot3 name.
  /Y            Suppresses prompting to confirm you want to overwrite an existing destination file.
  /-Y           Causes prompting to confirm you want to overwrite an existing destination file.
  /Z            Copies networked files in restartable mode.

The switch /Y may be preset in the COPYCMD environment variable. This may be overridden
with /-Y on the command line.  Default is to prompt on overwrites unless COPY command is
being executed from within a batch script.

To append files, specify a single file for destination, but multiple files for source (using wildcards
 or file1+file2+file3 format).
```

MSDOS help (help copy)

- Feature organisation (what flags do)
- Detail – limited
- Writing – moderate (early secondary)
  - » Some user-orientation (in append)
- Layout – limited, some attempt made
- Examples – none (?)

Use in reference manual for techies?

Unix Help (man cp, SunOS 5.9)

User Commands cp(1)

NAME

cp - copy files

SYNOPSIS

/usr/bin/cp [-fip@] *source_file target_file*

/usr/bin/cp [-fip@] *source_file*... *target* [……]

DESCRIPTION

In the first synopsis form, neither *source_file* nor *target_file* are directory files, nor can they have the same name. The cp utility will copy the contents of *source_file* to the destination path named by *target_file*. If *target_file* exists, cp will overwrite its contents, but the mode (and ACL if applicable), owner, and group associated with it are not changed. The last modification time of _*target_file* and the last access time of *source_file* are set to the time the copy was made. If *target_file* does not exist, cp creates a new file named *target_file* that has the same mode as *source_file* except that the sticky bit is not set unless the user is super-user. In this case, the owner and group of *target_file* are those of the user, unless the setgid bit is set on the directory containing the newly created file. If the directory's setgid bit is set, the newly created file will have the group of the containing directory rather than of the creating user. If *target_file* is a link to another file, cp will overwrite the link destination with the contents of *source_file*; the link(s) from *target_file* will remain.

[…]

Unix Help (man cp, Linux Fedora 11)

```
CP(1)                    User Commands                    CP(1)

NAME

     cp - copy files and directories

SYNOPSIS

     cp [OPTION]... [-T] SOURCE DEST

     cp [OPTION]... SOURCE... DIRECTORY...

DESCRIPTION

     Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

  Mandatory  arguments  to  long  options are mandatory for short option too.

     -a, --archive

          same as -dR --preserve=all

     --backup[=CONTROL]

          make a backup of each existing destination file...

    -b    like --backup but does not accept an argument
```

Unix help

- Feature organisation (what flags do)
- Detail – quite technical
- Writing – harder (late secondary)
  - » Techy terms: ACL, mode, setgid bit, …
- Layout – minimal
- Examples – provided if you scroll down (not on previous slide)

Don’t do this for beginners!!!

Organisation & detail

» Task oriented often best

» Include examples

» Tailor to user

Task-Oriented Help

- Decide who you think the user will be.
  - Background, expertise, preferences
- Decide what tasks you expect the user to do with your system.
- Write instructions telling him/her how to perform these tasks with your system
  - How to prescribe a drug to a patient
  - *not* How to use the Prescription screen
- Test your instructions with users

Similar to GUI design!!

GUI design and GUI documentation:

- *Both*: user modelling
- *Both*: task modelling
- *GUI design*: design interface which makes task easy to perform
- *GUI doc*: explain how to perform tasks with the GUI
- *Both*: test with users

Email User Analysis (New Students)

- *Tasks*: send and receive email
  - » give step-by-step instructions for this
- *Information needs*: CS Dept email addresses
  - » include list of important CS email addresses
- *Motivation*: course information, chat with friends
  - » base examples on these
- *Expertise*: know Windows, may not know about email
  - » don’t explain mouse, do give examples, background on email
- *Preferences*: learn by experimentation
  - » Give basic instructions and cautions/warnings
- *Usage patterns*: will use every day
  - » doc read a few times, not constantly referred to

Include Examples

- Show how system is used to solve a real problem
  - » good example shows context of use
  - » Screen shots are useful
- Show how system is typically used
  - » what’s important in typical usage
- Useful as templates for users

Level of detail depends on user

- Don’t tell experts what they already know
- Don’t leave out what novices need to know
- Don’t confuse novices with too much info

Multiple Types of User Manuals

- Introduction/tutorial for new users
  - » Examples of common usage
  - » Show novices what program can do
- How-To reference
  - » Step-by-step instructions for common tasks
  - » Best default structure
- Functionality reference
  - » Describe each field, option
  - » For experts doing fancy things

For experts, functionality based organisation can be fine!

Writing style

Keep it simple and direct

Simple Sentences

- Short sentences
  - » The dialogue box will show all files in the directory so click on the one you want to open.
  - » The dialogue box will show all files in the directory. Click on the one you want to open.
- Avoid long noun groups
  - – *read-only flag of the target file* vs
  - – *target file read-only flag*

User-centered

- Active voice, user-centered
  - » *Press Enter* vs
  - » *The Enter button must be pressed*
- Don’t mention “the user”
  - » You will see a new cursor.   *vs*
  - » The user will see a new cursor

Simple Grammar

- Use articles (*the*, *a*)
  - » *Press the Undo button* vs *Press Undo*
- Avoid complex tenses
  - » *The data file is updated when the Save Dialog finishes*
  - » *The data file will have been updated when the Save Dialog finishes*

Tone

- Positive tone
  - » *Please insert the CD* vs
  - » *You forgot to insert the CD*
- Not too formal
  - » *Most users make regular backups of their documents* vs
  - » *It would be to your benefit to make a back-up copy of your documents*

Write short sentences

- The author performs the following tasks:
  - » Collect the necessary information.
  - » Analyze and evaluate the information.
  - » Write a structured draft.

- Authors will approach any writing project by collecting the necessary information first, and after carefully analyzing and evaluating it, they will create a structured draft.

Use same format for same content

Printer Installation.

1) Remove the printer from the shipping carton.

2) Remove the printer from the plastic bag.

How to install your scanner.

1) Lift the scanner from the shipping **box**.

2) Discard all **packaging material.**

Write full sentences

Write:

Do you wish to continue the installation of the software?

Do not write:

Continue installing software?

Avoid Spelling Mistakes

- Texts that contein speling misstakes complicate the translation procces.

Measuring language complexity

- Sometimes measured by "age level"
  - » "Age 12" means should be readable by a 12-year old who has passed the standard reading tests for people of this age
  - » Determined by word choice, sentence length, sentence complexity, etc

- Word gives (crude!) estimates
  (Select **check grammar with spelling** and **show readability statistics** under Tools/Options/Spelling and Grammar; Click on **Tools/Spelling and grammar**)
  - » "grade level" – add 5 for age

Target Age-level

- Users with advanced degrees – age-level 20 is OK, 15 is better
- Users with degrees – age-level 15 is OK, 12 is better
- Average people – 12 is OK, 10 is better
- People with literacy problems – the lower the better!

Controlled Language

- Documentation can be critical to safety
- Controlled language specifications exist in some areas, notably aerospace and defense.
  - Unambiguous sentences
  - Restricted vocabulary
  - Restricted grammar
  - http://www.asd-ste100.org/
  - http://www.boeing.com/phantom/sechecker/se.html

Aerospace

- Documentation for an aeroplane can weigh as much as the plane!
- Often needs to be translated into multiple languages
- Use of controlled language helps
  - » AECMA Simplified English adopted by entire industry

Controlled Language

- Many companies use proprietary specifications/tools for controlling language in documentations
  - Microsoft, IBM, Ericsson, Sun Microsystems, Xerox, Kodak
  - General Motors, Saab, Scania, Rolls-Royce
  - Boeing, Dassault, European Aeronautic Defence and Space Company (EADS)
  - Many More...

Controlled Language

- What do proprietary checking tools do?
  - » Point out ambiguity
    - –**Fasten the assembly with the lever.**
    - –**Keep it running**
  - » Point out non-standard grammar or vocabulary
- Only a writing aid, author has to take responsibility

Visual appearance

Font, layout, figures

Visual Appearance: General Principles

- Visual features help the text
  - » Easier to read, reinforce main points
  - » Do not (in most cases) replace text
- Font/style indicate special words
- Layout shows structure of document.
- Figures used for examples, overviews, reinforce key points, ...

NetBeans Tutorial

- Text
  - **Font**: `Courier` for names (Java classes)
  - **Bold face**: key info (eg, duration)
  - Colour: Headers
  - Size: large font in main headers
- Layout
  - Headers for sections
  - Numbered lists for action sequences
  - Bullet lists for definitions
- Graphics
  - screenshots for examples

NetBeans Example

- Follow a style
  - » Java classes always in `Courier` font
  - » Sequences always in numbered list
  - » Main headers in red
  - » etc
- Consistent use of visual features helps reader
  - » Can easily tell what is a `Java` class name

Text

- Font
  - » main font should be well known (Times?)
  - » other fonts for special purposes
    - – non-proportional font (`Courier?`) for program code, filenames
    - – Sans-serif font (Arial?) for headers, titles?
- Style
  - » normally in plain style
  - » *Italics* for emphasis?
  - » **Bold** for definition, critical information?
- Size
  - » warnings, titles larger?
  - » footnotes smaller?

Be consistent (could use stylesheet)

Layout

- Headers, titles, lines: document structure
  - » Help user find correct section of document
  - » Should stand out (eg, large font)
- Bullet, numbered lists: domain structure
  - » Help user see groupings, orderings
  - » Instruction sequences
  - » Grouped objects (options, definitions)
- Tables: also show groups of objects

Figures

- Many different kinds!
  - » Screenshots: useful in examples
  - » Flow diagrams: show how system works
  - » Flowcharts: show instruction sequences
  - » Icons: indicate warnings, cautions
  - » Maps: locations of things
  - » Cartoons: reinforce points in a humorous way
  - » etc, etc

Figures

- Usually supplement (not replace) text
  - » Figures present key information in a different way
  - » Multiple presentations help understanding
  - » Figures also can tell user what information is important
- Don’t communicate important information just in figures
  - » some people ignore; accessibility issues

Figures: Screenshots

- Suggestions
  - » Include example data
  - » Ensure large enough to be readable
  - » Don’t be afraid to crop

Printed Page vs On-Screen

- Text
  - » size: can be smaller on paper (10pt vs 12pt)
    - – paper is easier to read
  - » font: should use standard ones on screen
- Layout
  - » Larger margins (>= 1 inch) on printed page
    - – binding, space for handwritten notes
  - » Warnings/cautions on same page?
    - – Real problem with on-screen
- Graphics
  - » resolution usually better on printed page

Visual Appearance: Key Points

- Be consistent
- Extra channel for information
  - » word type, document structure, etc
  - » reinforce, restate key messages
- Useful in moderation
  - » Overuse decreases effect, also decreases document readability

Summary

- Documentation is to help users
  - Based on analysis of user needs
  - Helps users do things

- Task-oriented How-To structure
  - Describe how to perform common tasks
  - Good default structure

- Use simple and direct language
- Use visual appearance (not just a long text)
- Include examples

Final observation

GUI design and GUI doc are similar tasks

- Goal in both is to help the user perform tasks
- Both require understanding of users
- Different technical skills (programming, writing)
- Should technical writers and programmers work together developing systems?
  - » Instead of first build, then document?