Cognitive Disability and Technology: Universal Design Considerations

Clayton Lewis
Coleman Institute for Cognitive Disabilities
RERC-ACT
clayton.lewis@colorado.edu
Prepared for AUCD Training Symposium, Nov. 10, 2008

Introduction

This document extracts and generalizes information on Web accessibility being prepared as a collaborative effort with the web accessibility initiative at the W3C. Comments and suggestions are welcome!

1. What problems may people with cognitive, language, and learning disabilities experience in using technology?

Here are some examples:

Some users have trouble understanding text, especially unusual words and complicated sentences.

Some users have trouble doing operations that require many steps, or that have many choices.

Some users have trouble skimming text to find what they are looking for.

For a longer list of challenges, see Appendix A.

2. What can designers do to make technology easier to use for people with cognitive, language, and learning disabilities?

While there are many problems users may face, there are a few general design approaches that will help many users. Most of these approaches will make your technology better for all users, and none of them will make it worse.

In addition to using these approaches, designers should follow a design process that includes getting input about user needs, and getting feedback on how well their design actually works. See Section 3, below.

2.1 Use multiple modalities (text, pictures, sound) to present information.

People differ in their ability to process information in different forms. So providing more than one form reaches more people. Also, a person who can read, but not well, will do better with a combination of text and spoken words than with only one of the two. Pictures may also help by suggesting what the context for information is, or by showing details that are hard to express in words.

Don't rely on any one modality to work on its own. While some users can't see the screen, others can't hear audio alerts.

For some users, using symbols, in addition to pictures, will be helpful.

2.2 Use clear, simple organization of information. Use headings and layout to make it clear what the organization is.

A challenge for all users is finding the information they want on computer screens or displays that contain a lot of information that they don't want. A look at a newspaper shows the basic tools for helping with this:

Use headings to make it easy to identify different topics.

Use layout to make it easy to separate different topics.

Present the most important information first. Newspaper writers are careful to organize a story so that the first paragraph can be read by itself to get the basic information. Readers who are interested, and have the time, can read more.

Use a simple linear structure within a topic. While sidebars and boxes are sometimes used, newspaper stories are usually structured so that they are read from beginning to end.

These principles from newspapers are good for computer screens, too.

Some Web sites are being redesigned to offer simple, uncluttered presentations on mobile devices with small screens. According to Markoff(2008) some users find these presentations easier to use that more cluttered presentations on large screens.

> Markoff, John. (2008) Slipstream: On a small screen, just the salient stuff. New York Times, Sunday, July 13, 2008, p. BU 3. http://www.nytimes.com/2008/07/13/technology/13stream.html?n=Top/News/Business/Companies/Facebook,%20Inc.

Use subheadings to identify parts of larger units of content. Newspapers make limited use of subheadings, but technical journals and books use many of them. They can help readers find specific information more quickly.

2.3 Make text easy to read.

Use short lines of text: Shorter lines of text will minimize users getting lost when transitioning from the end of one line to the start of the next line.

Avoid fully justified text that creates white space of varying size between words. The varying white space makes it harder to follow the line of words.

Use the default justification for the language being used. This helps readers go from the end of one line to the start of the next line.

Use solid backgrounds: Patterned backgrounds can make text "swim", making the text difficult to read. It can also make it difficult for users to recognize words and track along the required sequence.

2.4 Make it possible for users to shape your presentation to fit them.

Good designs make it possible for users to change the size of fonts, and change the colors that are used so that text is easier to read for them. Here are some other ways users can control your presentation.

Let users ask for definitions or explanations of unfamiliar terms and abbreviations.

If you use expressions that aren't meant literally, provide an explanation. Some users can work out what the words in a sentence mean literally, but may not get what is really meant. For example, the sentence, "The new phone is the best thing since sliced bread," has nothing to do with bread. If it is important to say it, indicate to the user that an explanation is available.

Support users' assistive technology. Some users will use a screen reader, or text to speech features on their computer. Your application should be configured so that it supports that.

Let users suppress details Let them ask to see only the headings when they first look at your display, and then ask for the details on only the parts that are important to them.

2.5 Make interactions as simple and clear as possible.

Most technology applications don't just present information, they let users do things by interacting with them. This means that the principles of usability that have been recognized for desktop software also apply to Web sites. Usability is important for all users, but especially for users who may have extra trouble reading complicated instructions, carrying out a long series of choices and actions, and recovering when things go wrong.

A good interaction is like this:

It is short.

It is always clear what the user must do next.

It uses common, familiar user controls.

It requires only a few choices, or no choices.

It requires only choices that users can easily make.

It provides clear feedback on what is happening, good or bad.

Creating a good interaction in a complicated situation can take a lot of work. Sometimes it takes programming work, not just user interface design work. For example, one way to eliminate a choice may be for the software to find the answer to a question without asking the user. It may be easier for the programmer to ask the user. But asking the user has big problems:
The interaction is longer.
The user may not be able to answer the question.
If the user gives the wrong answer there can be a big mess.

Be careful about doing new things with your controls. People get used to common controls, and they learn how to use them. If you use new things you may require problem solving that will be hard for many users. It may be especially hard for users who are not good at rapidly scanning or skimming a page to find and recognize an unfamiliar control. New controls may not work well with assistive technology.

Feedback should be obvious, using both visual and auditory cues. It should be presented in a way that works well with screen readers ##need specifics on this... the issue is that just putting something up on a page may not alert a screen reader user that it is there.

Feedback, including error messages, should not require the user to remember things. For example, the message "invalid phone number" is not as good as "333 333 333 is not a valid phone number".

For technical people, having a "mental model" of a system is often helpful. But for many other people, whether they have cognitive, language, or learning disabilities or not, it is hard to develop and use mental models of computer applications. Try to develop interactions that work even when users do not understand in depth what they are doing and why, where possible.

Many users, whether or not they have cognitive, language, or learning disabilities, do not understand operations like storage and retrieval, or common computer metaphors like "folders". If possible, design interactions that work without requiring this understanding.

Make it as easy and obvious as possible how to recover from mistakes. Try not to permit actions that could be catastrophic, and require clear confirmation if these actions can't be eliminated.

2.6 Provide for user support.

Many people with cognitive disabilities are supported by family members or caregivers when they have difficulty with technology. Your system should make it easy for such support persons to configure the system for most effective use. If possible, it should provide for sharing of controls and displays, and remote management, so that assistance can be provided without requiring the support person to be physically present.

2.7 Other issues.

The design approaches above cover what designers should think about in common situations. Here are some points that come up less often.

Don't require users to respond quickly. If you find yourself doing this, say in an interactive game, provide a way for users to change the timing if it takes them extra time to respond.

Security and privacy are very important with much computer technology, but many people do not understand these concepts very well. Your system should provide defaults that protect users who are not sophisticated.

3. Roadmap: How to Create Accessible Technology

3.1 Key steps in creating accessible technology, from Just Ask: Integrating Accessibility Throughout Design, by Shawn Henry.

Before you begin your project:

1. Research legal and other requirements for accessibility of your

products.
2. Research accessibility standards and guidelines for your type of product.
3. Develop internal policies and guidelines for accessibility.
4. Budget and schedule to include people with disabilities as collaborators in your project.
5. Develop accessibility knowledge and skills through training and hiring, as appropriate.

You should involve users with cognitive, language, and learning disabilities, as well as other disabilities, in your planning and design process. User participation will help you understand user needs, a requirement for good design.

Analyze the objectives for the system:

1. Vision, goals, objectives
2. User analysis
3. Task analysis
4. Information architecture analysis
5. Workflow analysis

Design the system:

1. Conceptual/mental model, metaphors, design concepts
2. Navigation design
3. Storyboards, wireframes
4. Detailed design
5. Paper or other low-fidelity prototypes
6. Medium-fidelity prototypes, for example, online mockups
7. Functional, high-fidelity prototypes

Evaluate the design:

1. Design walkthroughs, cognitive walkthroughs
2. Heuristic evaluations
3. Guidelines reviews
4. Usability testing: low fidelity through high fidelity; informal through formal

Here are suggestions about some of these steps, specifically for users with cognitive, language, and learning disabilities.

3.2 User analysis.

Involve users, including users with disabilities, in your design process. Consider using proxies (for example caregivers, job coaches) to gather additional information about users and their needs.

Remember that many people with cognitive disabilities use assistive technology, such as screen readers, or screen magnifiers. Support for these tools is basic to all accessible design.

Do not assume that people with cognitive, language, and learning disabilities will not want to use your system. People with these disabilities are part of all professional and interest groups.

3.3 Task Analysis.

What are the most important tasks users will have? How can you make these tasks as simple and clear as possible, for all users?

3.4 Information architecture analysis.

What is the most important information for users? How can you make this information as simple and clear as possible, for all users?

3.5 Navigation design.

How can you make it easy for users to go back to a familiar state of the system?

How can you make it easy for users to tell what a new display is about?

3.6 Design walkthroughs.

Does your design require complicated reasoning to choose the next step? Can users get the definitions of unfamiliar words and abbreviations if they need them? Are sentences needlessly complex?

Appendix A: Challenges for Technology Users

Here are some areas that some users find difficult. The table does not include the possible causes of these troubles. In some cases the troubles may be related to disabilities attributed to a person. In other cases the problems may be situational. For example, a person may not understand a complicated idea because they are tired, or stressed, or because they do not understand the language being used. Designers should focus on solutions to the functional problems, and not dwell on the causes for each problem.

1. Understanding Information

* recognizing written words
* understanding unfamiliar words
* understanding spoken language
* understanding complicated or unfamiliar ideas
* recognizing objects in pictures
* rapidly skimming text
* understanding numbers
* recognizing and interpreting symbols
* understanding the figurative meaning of words and/or phrases
* understanding contextual implications
* categorization of content (for example, important versus unimportant)
* following required sequence of words
* chunking of content
* understanding sounds used as cues or feedback

2. Understanding computer operations

* interpreting complicated displays
* learning and using common cues to Web page layout
* ignoring controls or messages that are not needed
* recognizing changes in task context
* detecting and recovering from error
* using memory to guide search for operations or information
* interpreting complicated feedback
* noticing subtle feedback
* use unfamiliar controls
* understanding security and privacy

* understanding what is, and is not, in the scope of their control
* understanding how to store and retrieve information
* understanding computer metaphors (e.g., "folders" where files are kept)
* more generally, understanding a model used to present or organize a task

3. Taking Action

* deciding what to do when there are many choices
* keeping track of steps in complicated tasks
* taking action when given limited time
* remembering what action to take for a particular result
* difficulties with standard mouse/keyboard operation
* using actions that are about other actions (i.e. how to stop something, how to undo something, how to get help)