Course: Human Computer Interface (HCI)

Week 13 – User Experience and Support

Lecturer: **Martha Gichuki**

Lecture Learning Outcomes

At the end of this lecture, the learner will be able to:

i. Define the basic requirements for user support
ii. Describe the various styles of user support
iii. Describe issues to consider when designing user support systems

Introduction

The type of assistance required by users varies broadly depending on many factors such as user familiarity with the system, the job they are trying to do among other factors.

There are four main types of assistance that users require:

(i) *Quick reference* - mainly used as a reminder to the user of the details of tools they are basically familiar with and have used before. It may, for example, be used to find a particular command option, or to remind users of the command syntax.

(ii) *Task-specific help* - required when users have encountered a problem in performing a particular task or when they are uncertain how to apply the tool to specific problems. The help that is offered is directly related to what is being done.

(iii) *Full explanations* - more experienced or inquisitive users may require a full explanation of a tool or command to enable them to understand it more fully. This explanation will almost certainly include information that the user does not need at that time.

(iv) *Tutorials* - aimed at new users of a tool and provides step-by-step instruction (perhaps by working through examples) of how to use the tool[1].

REQUIREMENTS FOR USER SUPPORT

[1] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 423

How would an ideal help system look like? There are useful features used as benchmarks against which designers can test support tools, they design[2]. HCI users have different requirements for support at different times and user support should have the following features:

(i) *Availability*: users should access help any time they are interacting with systems. Users don't have to quit the current application to open the help application. Help systems should run concurrently with any other application and this is a not the case for non-windowed systems where the help system is independent of the application running. With windowed systems a help facility is available constantly, at the press of a button[3].

(ii) *Accuracy and completeness:* due to the frequent software applications updates, accuracy and completeness are difficult aspects of support since different versions may be active at the same time. If the assistance provided does not match the actual behavior of the system users will be disillusioned with the help facilities and get into difficulties. The help-function should cover the whole system and this completeness is very important if the help provided is to be used effectively. Designers cannot predict the parts of the system the user will need help with, and must therefore assume that all parts must be supported. Finding no help available on a topic of interest brings frustration to users.

(iii) *Consistency*: different parts and versions of the help system should be consistent in terms of content, terminology and style of presentation.

Users require different types of help for different purposes and helps system have different parts. The help provided by each of these must be consistent with all the others and within itself. Online help should also be consistent with paper documentation in terms of content, terminology and style of presentation. Applications with internal user support; should be consistent across the system. It is unhelpful if a command is described in different ways or if help is accessed differently across applications. Consistency supports the user since it reinforces learning of system usage.

[2] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 424

[3] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg.424-426

(iv) *Robustness*: the help function should be predictable and not easily influenced by errors since it is mostly used when the user is experiencing system problems. Help systems are often used by people who are in difficulty, mainly when the system is behaving unexpectedly or has failed altogether. Help systems should be robust, both by correct error handling and predictable behavior. Users should be able to get assistance whenever they need it.

(v) *Flexibility:* the ideal help system should adapt to the properties of the users and the environment. Many help systems are rigid since they generate the same help message regardless of the expertise of the users. Flexible help systems allow all users to interact with them in a way appropriate to their needs ranging from designing interactive help systems, through context-sensitive help, to full-blown adaptive help systems, which will infer the user's expertise and task. For example, help systems built using hypertext principles allow users to browse through the help, expanding topics as required. The top level provides a map of the subjects covered by the help and the user can get back to this level at any point. Although hypertext may not be appropriate for all help systems, the principle of flexible access is a useful one.

(vi) *Unobtrusiveness:* the help system should not prevent users from continuing with normal work, nor should it interfere with the user's application. Textual help system on a non-windowed interface may interrupt the user's work and a possible solution to this is use of a split-screen presentation[4].

APPROACHES TO USER SUPPORT

A number of different approaches to providing help exist and each of these meets a particular need. These vary from simple captions to full adaptive help and tutoring systems and user support comes in a number of styles[5]:

(i) ***Command assistance:*** users request for help on a particular command and are presented with a help screen of manual page describing it. This type of help is simple and efficient if the user knows what they are looking for; either a reminder or more detailed information. For complex computer system,

[4] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 424-426

[5] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 426-432

users will know and use some commands well while others though known are rarely used. Command assistance deals well with these but there are those commands the user does not know about but needs, and commands that the user thinks exist but which do not.

(ii) ***Command prompts:*** in command line interfaces command prompts provide help when the user encounters an error, usually in the form of correct usage prompts. These prompts are useful but assume knowledge of the command, so additional support is required. Another form of command prompting, is the use of menus and selectable icons, which, provide memory aid and clearly show the commands available at any given time.

(iii) ***Context-sensitive help:*** these range from those with specific knowledge of the particular user to those that provide a simple help key or function interpreted according to the context in which it is called to present help accordingly. Such systems do not require users to remember the command and they are often used in menu-based systems to provide help on menu options. E.g., in Microsoft Office - ***What's This?*** option, tool-tips and some kinds of web page rollover. When enabled, explanatory text is displayed when the cursor is over a screen widget. The invocation of help is interpreted in terms of the context in which it is made.

(iv) ***Online tutorials:*** allow users to work through the basics of an application within a test environment. An alternative to the traditional online tutorial is to allow users to learn the system by exploring and experimenting with a version with limited usability.

Users can progress at their own speed and can repeat parts of the tutorial if needed. Users experiment with examples or watch automated demonstrations of how to perform tasks. Online tutorials are not flexible and some fail to recognize correct answers to problems, since they are not formatted as expected. Users are presented with full interface versions with some functionalities disabled. Users can explore the system freely and whenever they attempt to use blocked functionalities, unavailability feedback is given to them.

(v) ***Online documentation:*** makes the existing paper documentation available on the computer for a larger number of users effectively and continually but on the assumption that the machine is running in the same medium as the user's work. Documentation provides a full description of the systems functionality and behavior in a systematic manner: a high amount of generic information. Minimal

manuals should provide enough information for less experienced users. Paper manuals get lost easily, are constrained to one physical location, and are always somewhere else when needed. Online documentation is one way of avoiding these problems. Manual pages contain high amounts of information, which can create problems for the user; since too much detail hides important information the user wants to find. For this reason, online documentation is often used by more expert users as a resource or reference, sometimes to enable them to advise less experienced users. Experts may not know the information off-head but they know where to find it and how to extract relevant details for a given problem. Hypertexts make online documentation more accessible to the inexperienced user since they store text in a network and connects related sections of text using selectable links. Upon clicking on a link, users can go to a related subject instantly.

The potential disadvantages of reading from displays rather than from paper include: -

- *Poor fonts*, especially on low-resolution displays. The dots composing the letters may be so large that each is visible, making users unable to recognize characters. Improper inter-letter spacing, inter-line spacing, and colors may all complicate recognition.
- *Low contrast* between the characters and the background and unclear character boundaries can complicate recognition.
- *Emitted light from displays* may cause reading difficulties compared to reflected light from paper; glare may be greater, flicker can be a problem, and the curved display surface may be troubling.
- *Small displays* require frequent page turning commands which is disruptive, and the page turns are unsettling, especially if they are slow and visually distracting.
- *Reading distance* for most displays is fixed in place and not easily adjustable like for paper. Display placement may be too high for comfortable reading.
- *Layout and formatting* e.g., improper margins, inappropriate line widths and justification, multicolumn layouts requiring constant scrolling up and down, distracting page breaks that waste space.
- *Reduced hand and body motion* with fixed-position displays, as compared to paper, may be more fatiguing.
- *Unfamiliar displays* and the anxiety of navigating the text can increase stress[6].

[6] Designing for Effective Human/Computer Interaction (4th ed.), Schneiderman, B., Plaisant, C.: Pearson Education, Inc.: ISBN 0-321-19786-0, (2005). Pg. 547-548

(vi) Wizards and assistants:

Wizard: a task-specific tool that leads the user through the task, step by step, using information supplied by the user in response to questions along the way. Wizards are common in application as they offer the user the possibility to perform a complex task safely, quickly and efficiently. Wizards can, however, be unnecessarily constraining and they are distinct from demonstrations since they allow users to complete tasks. An example is the Microsoft Word resumé wizard that creates a resumé matching the parameters submitted by the user.

Assistants: These are software tools that monitor user behavior and offer suggestions or hints when they recognize familiar sequences. Assistants should not be obtrusive[7].

Adaptive help systems - In any large or complex computer system, users are familiar with a subset of the available functionality, demonstrating expertise in some applications and having no experience with others, even to the point of being unaware of their existence. Different users have different needs and levels of understanding, hence adaptive help systems assist individual users making requests by actively suggesting alternative courses of action, which the user may not be aware of. Adaptive help is a special case of a general class of interactive systems, known as *intelligent systems*. These include domain-specific expert systems, intelligent tutoring systems and general adaptive interfaces.

Adaptive help systems monitor user activity and construct help models based on user experience, preferences, mistakes, domain knowledge - in which the user is working, general advisory or tutorial strategies. This enables the system to provide assistance based on the user's current task and suited to their experience.[8]

DESIGNING USER SUPPORT SYSTEMS

Designers of help support need to take into account the following issues: -

- ✓ User support design should not be seen as an 'add-on' to system design but the help system should be designed together with the rest of the system to ensure relevance and consistency of

---

[7] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 403

[8] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 426-432

the help system with the rest of the system. The same modeling and analytic techniques used to design the system should also guide the design of user support material.

- ✓ Designers should consider the content of the help and the context in which it will be used before the technology that will be required. Available technology is an important issue to consider but concentrating on the task and the user will help clarify the type of help required within the constraints of technical resources. Viewing the process in reverse may prevent the designer from seeing beyond the technology they are familiar with. Since user requirements vary, help support designers need to decide on how the help will be presented to the user and how this will be affected by implementation issues[9]

A. **Presentation issues**

For effective presentation the right language style, normal design guidelines such as readability and indexing should be applied in design.

(i) *How is help requested?*

Designers need to decide how users will **access** help either through commands, buttons, functions which can be switched on or off, or separate applications. Commands require users to specify a topic, therefore assuming some knowledge. Help buttons are easily accessible and do not interfere with existing applications, but may not always be context sensitive. Help buttons can support context sensitivity if they are keyboard or mouse based. Help functions are flexible since they can be activated when needed and disabled when not. Separate applications allow flexibility and multiple help styles but may interfere with the user's current application.

(ii) *How is help displayed?*

Designers must decide how the user will **view** the help e.g., in a new window, pop-up boxes or at command line level. In windows system help may be presented in a new window while other systems may use part of the screen or the whole screen. Help hints and prompts can be given in pop-up boxes or at command line level. The presentation style that is appropriate depends largely on the level of help being offered and the required space. Opening a manual page line by line is unhelpful, as is taking over the whole screen to give the user a hint. Some active help systems provide visual cues when they have a suggestion to make giving users the option of taking the suggestion without forcing them to abandon or

[9] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 439-442

interrupt their work. This decision should take account the rest of the design, with the aim of providing consistency[10].

B. **Implementation issues**

Designers of the help material should be involved in the design process and they need to choose the most suitable method for any given system since user support can be provided in many ways. Some implementation decisions may be forced by physical constraints, others by the choices made regarding the user's requirements for help. Each of these decisions involve implementation questions such as: -

a) Will help be an operating system command, a meta-command or an application?
b) What physical constraints does the machine impose e.g., screen space, memory capacity and speed? Speed is crucial; slow response time makes the system unusable regardless of the design strengths. Simple fast responding help functions supersedes complex intelligent systems that take ages to provide solutions.
c) How will the help data be structured: in a single file, a file hierarchy, a database? - this will depend on the type of help required, but any structure should be flexible and extensible. Systems are not static and new topics will need to be added to the help system. The data structure used determines the type of search or navigation strategy provided.
d) Will users be able to browse through the system or only request help on one topic at a time? - Users may want to make hard copies of part of the help system to study later (manuals and documentation).
e) Will this facility be provided as part of the support system? Designers of help material as well as its users need consideration since designers may write the initial help texts, which will be extended by other authors at different times. Clear conventions and constraints on the presentation and implementation of help facilitate addition of new material.[11]

[10] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 439
[11] Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004). Pg. 439-442

Content Covered in Week 13: User Experience and Support

(i) We have defined the basic requirements for user support

(ii) We have described the various styles of user support

(iii) We have described issues to consider when designing user support

References

1. Human-computer interaction (3rd ed.), Dix, A., Finlay, J., Abowd, G., & Russell, B. New Jersey: Prentice Hall. ISBN-10: 0130461091, ISBN-13: 978-0130461094, (2004).
2. Designing for Effective Human/Computer Interaction (4th ed.), Schneiderman, B., Plaisant, C.: Pearson Education, Inc.: ISBN 0-321-19786-0, (2005).
3. The design of everyday things, Norman, D. A. New York: Basic Books. ISBN-10: 0465067107, ISBN-13: 978-046506710, (2002).
4. Designing the user interface: Strategies for effective human-computer interaction (5th ed.), Shneiderman, B., Plaisant, C., Cohen, M., & Jacobs, S. New Jersey: Prentice Hall. ISBN-10: 0321537351, ISBN-13: 978-0321537355, (2009).