Human-Information Interactions with Complex Software

Michael J. Albers

East Carolina and University
Greenville NC 27858
albersm@ecu.edu

Abstract. This article extends the analysis of a usability test of C2PC, a US Marine Corps command and control software product. The study revealed the C2 operators were able to perform simple tasks, but had difficulty combining those simple tasks into realistic tasks. These differences highlight the need to both consider the complex interactions during HCI design and for using complex scenarios when testing complex systems. Poor human-information interaction (HII) is reflected in designs which fail to support effectively rolling up the individual tasks into the complex interactions that people must perform. Usability tests show basic tasks can be accomplished but these systems fail to support people solving open-ended, unstructured, complex problems which require extensive and recursive decision-making or problem solving This paper discusses how the issue appears in many software products, causing problems in effectively communicating information. It considers the broader design issues for complex information spaces.

Keywords: Human-information interaction, usability testing, complex information systems, complex system design.

1 Introduction

Individuals seek information because they realize that their knowledge about a situation is incomplete and that the information they need can be found within the system. Design teams are increasingly being called upon to address information needs which go beyond providing simple answers or step-by-step instructions and which involve communicating information for open-ended questions and problems. Questions and problems that can only be addressed by providing information specific to a situation and presenting it in a way that supports various users' goals, information needs, and cognitive processing strategies.

With respect to HCI and usability, it's the human-information interactions (HII) which lead to comprehending the information [8]. As a system's complexity or its information content complexity increases, the need to understand and predict people information interactions make it imperative that designs work from a basis of how people organize their thoughts. This, of course, contains implicitly the importance of understanding people's thoughts and perceptions.

Mirel's [15] studies have found users have different conceptions of how to accomplish a task, which she described by saying: "In actual work settings, users

A. Marcus (Ed.): Design, User Experience, and Usability, Pt II, HCII 2011, LNCS 6770, pp. 245–254, 2011.

define their own tasks and task needs according to situational demands, not program design" (p. 15). The design of those systems must encompass a totality that revolves around the goals and information needs of a person and supplies information that makes sense within the person's real-world situation.

Handling the shift from a linear information model to the ill-structured model requires a design shift. Redish [19] argues that documentation move up a level and address goals and task repertoires and she recognizes that that emphasis on "higher than discrete task" is crucial for anyone doing everyday work. Accomplishing this requires us to view the creation and presentation of content and its subsequent communication from a humanistic viewpoint, rather than a mechanistic one [25]. The user goals and information needs must be placed within a proper social and technical contexts and designed to assist people, rather than doing it for them [21, 22]. The driving force for this shift arises because people, rather than machines, are reading the information. [3, p. 4]

The issues of simple and complex information systems and designs which either conflate the two or assume only simple interactions form the basis of many poorly designed systems [5]. I've considered the issue of simple and complex in more detail in earlier work [3]; here I'm using the same definitions, which in a reduced form are:

Simple systems. A single or multiple fully-defined paths from problem to solution can be defined. It is possible to state that an action was either correct or incorrect.

Complex systems. Multiple and ill-defined paths to a solution exist. The solution depends on situation-specific context and can change between people or across time. It is impossible to clearly state if an action is correct.

People within a complex situation need to react within a highly dynamic environment and are faced with open-ended, complex problems. In a complex situation, the problem will almost away include factors or circumstances not foreseen as part of the original analysis. In these situations, the analysis and design must consider the highly dynamic situational context of information, the aspects of the information, and the information interrelationships required to support fundamental user wants and needs. These fundamental issues apply to any dynamic complex environment, such as military command and control, healthcare or financial systems. How can the design issues be resolved and what does this say for future system design? What does it mean for HCI to transition from designing for simple operations to complex information-based interactions. How do well do common HCI design strategies scale to complex dynamic environments, how can we help support the scaling, and how do we test the quality of the resulting design. This paper explores these ideas.

1.1 Communication, Not HCI Performance

Hollnagel [11] has described human work as fitting on a scale ranging from "doing" to "thinking." A simple task design based on performing any single menu option (with menu used loosely to include simple web interactions) fall within the "doing" region of the scale: a person has a stable, linear path to complete a task. With the rise of software and web pages providing interactive support for problem-solving or providing large amounts of information which a person is expected to mentally process and act upon, the interaction has shifted into the "thinking" region. Here people are engaged in knowledge intensive “thinking” activities focused around diagnosis, planning, and problem solving, rather than manually “doing” tasks. This shift has fundamental consequences for the design, content, and usability testing of the resulting system and the information it must communicate to the user.

Many traditional usability tests focus on function (essentially the button pushing) and not on the process of how the information is used. Recent work in usability calls this model into question, arguing that it is too divorced from reality to provide useful information [18]. In its place a new approach has been called for [20], one that recognizes that most users operate, or carry out their tasks, within complex systems that present multidimensional challenges—layers of changing depth that, unfortunately, traditional usability methods often cannot adequately measure. [5, p. 3]

The system’s overall goal involves communicating information to a user. The development and testing of any design and interaction should focus on the quality of that communication. Do the users comprehend the information and can they apply it to their situation? Thus, a goal of a usability test in a complex situation is to ensure that the information is being efficiently communicated and that the person is building information relationships. Developing an understanding of a complex situation requires understanding the relationships between information elements, not simply interacting with single information elements. [6]. This is a departure from traditional usability testing methods which dealt with single tasks and measurements of easy to quantify data, such as time to complete a task or total mouse clicks.

1.2 Design and Usability of Complex Information Systems

A problem facing design teams is that as products get larger and more complex, usability testing methods prove increasingly inadequate for testing usefulness. This has recently received discussion in the literature [1, 3, 16, 18, 19], and most of us have experienced it when performing tests of larger systems. Traditional, one- or two-hour usability tests find nothing of significance; yet when the product is used in a real-world situation people fail to complete tasks successfully [9].

Part of the testing problem can be a desire (demand?), either by management or the test team, to produce numerical results. Focusing on individual components makes it relatively simple to construct quantitative measures such as time to complete a component task or total number of clicks. However, quantitative evaluation of a complex design is suspect since the quantitative evaluation often privileges the easy-to-measure over other factors without a solid theoretical foundation for that privileging [24]. Instead, design teams creating products for complex systems must embrace flexibility and usability, and work to ensure that flexibility meets user needs [23]. On the other hand, as I’ll discuss later, too much flexibility can also impact usability if the users don’t need or want that flexibility to achieve their goals.

This articles extends the analysis of the results of a usability test of C2PC, a US Marine Corps command and control software product, which has been published elsewhere [7]. First, I’ll briefly review the C2PC study. Then, I’ll consider three design factors arising from the study and consider how they apply across many different design scenarios.

2 C2PC Test Summary

C2PC, currently deployed by the US Marine Corps, is a Windows-based military command and control (C2) system which provides an integrated views of the battle space in the Command Operations Center (COC). The C2 operator must manipulate a complex set of information in order to maintain situation awareness of the battle space (figure 1). Commanders at multiple command levels, from the battalion up to the main force level, depend on C2PC to maintain situational awareness and make operational decisions. To support these commanders, the C2 operator must manipulate a complex set of information (tracking military units, movement paths, and targets) to maintain a tactical picture of the battle space. By design, C2PC requires the C2 operators' interaction to shift between the map and a set of changing injectors (essentially modes) to accomplish their goals.

A usability study [7] revealed an underlying design problem seems to be that the C2PC designers provided an interface for performing a sequence of individual simple tasks, while real-world operation requires working in a cyclic manner with a complex set of information focused around open-ended questions. Instead, both the usability test and reports by training center instructors, reveal C2 operators tend to work either within the injector or within the map; they do not fluidly shift between them. Only after their preferred interaction area fails to yield a solution do the C2 operators shift to the other area. In the usability test, multiple C2 operators failed to ever change the zoom level (4 of 13) or change the injector (3 of 13) from its original settings. As would be expected, these C2 operators had high failure rates on the usability test tasks.

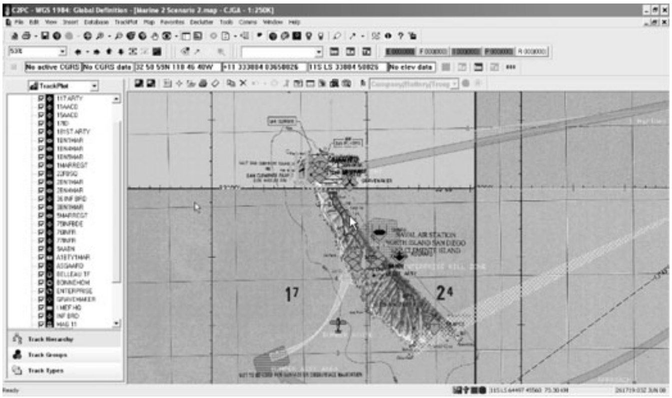

Fig. 1. C2PC main screen. A zoomable map of the battle space is on the right which shows all of the units. The list on the left is the injector, which varies depending on the software mode.

During the usability test the observations revealed an interesting disconnect between the underlying design assumptions and how the C2 operator interacted with the system. Many users never used the map zoom functions and had difficulty with simultaneously working with injectors and the map. A substantial usability problem which contributed to many task failures was the C2 operator's highly non-optimal use of the zoom function. Some subjects never changed the zoom and others made non-productive changes, such as zooming in too much or not enough, and then attempted to interact with C2PC. A second major usability problem was that C2 operators would search through the contents of the wrong injector before changing it or before going to the map to try and complete the task.

The problem with C2PC was neither the technical nor the computer-interaction level design, but rather that its HII aspects do not seem to have anticipated how people would actually interact with the program. Or more importantly, how they would interact with integrating all of the information available in the system and in their environment. A software design team too typically sees the software project as a collection of basic tasks, but realistic use requires combinations of those tasks (which the design team may or may not understand). The basic usability issues found seem to reveal that its design has not anticipated how people would actually interact with the program. Usability tests of individual tasks show they can be accomplished with minimal problems. However, individual interface tasks are not performed in isolation as people work with real-world tasks. Instead, they achieve their real-world goals by performing a grouping of interface-level tasks. C2PC fails to support effectively rolling up the individual tasks into the complex interactions which the C2 operator must perform. It fails to support working with solving open-ended, unstructured, complex problems which typically require extensive and recursive decision-making or problem solving [18, 19]. In the bigger picture, this support of individual tasks and lack of support for combing them into real-world tasks is a consistent problem with the design of many complex information systems[1].

Granted, the ideas of the previous paragraph are well accepted in HCI research and design; however, the implementation issues surrounding them have profound effects on people's actual performance and remain an open research question.

3 Broader Implications of the Results C2PC Usability Test

The remainder of this article will not dwell on the results of the C2PC usability test, but will consider how the fundamental HII disconnect found within that study occurs across many software products and causes an excess of failures in effective information communication. In an era long past, writing technical information meant writing for closed-ended questions, such as procedures or answering questions of "what is X." (Whether or not that was effective communication is a different question.) Thinking in terms of simple tasks or answering closed-ended questions exposes substantial issues with the usability of many designs. As Howard [13] summed it up: "Complex problems for users can masquerade as simple problems for authors, editors, designers, and even expert reviewers with experience in the tasks being performed" (p. 84).

Consider how, rather than answering "what is X," many texts and web pages are created with an intent of providing complex information for open-ended questions with a goal of communicating information that address questions such as "How are X and Y effecting Z?" Modern design teams need a much deeper understanding of what constitutes effective communication. Design teams are increasingly being called upon to address information needs which go beyond providing simple answers or step-by-step instructions and which involve communicating information for open-ended questions and problems. Questions and problems that can only be addressed by providing information specific to a situation and presenting it in a way that supports various users' goals, information needs, and cognitive processing strategies.

The problems found in the C2PC study tended to be complex problems which the design teams had viewed a collection of simple problems. However, complex problems are not the sum of their parts, they are much more than the sum. This section examines three design issues which expand on this idea.

3.1 Different Audiences Exhibit Radically Different Interaction Strategies

The C2PC study revealed interesting disconnects between the underlying design assumptions and how C2 operators at lower experience levels interacted with the system. They were able to perform simple tasks ("doing"), but were unable to combine those simple tasks into realistic tasks ("thinking"). These differences highlight the need to both consider the complex interactions during HCI design and for using complex scenarios when testing complex systems [19].

With complex information, the information needs and effective presentation formats can radically vary between audience groups. The system is expected to effectively communicate high-quality information to each group, although the information needs and background knowledge vary with each group.

In C2PC, the users could work with the system without using the zoom function. The C2 operators knew how to change the zoom, but they made a conscious decision not to do it or to minimize the changes. As a result, they were very inefficient and error-prone, but they could accomplish the task (at least they believed they had accomplished the task). With experience, they started to use zoom. From a designer's viewpoint, it's hard to visualize any screen which is not already at the proper zoom level or on the proper injector, or which the user cannot easily adjust with at most a couple of mouse clicks. And this is a major issue: a designer visualizes how to perform the task properly and has trouble figuring out how to cue the user that something needs to change (especially without interfering with expert level performance). The beginner or low-intermediate user, on the other hand, does not perform the expected integrated interaction and has usability problems which they often associate with bad design.

Most systems have to deal with a range of experience levels. As people's experience with a system increases, their interaction strategies change. The too common mantra "Write for the novice and the expert will understand it too" is not true; providing information designed for the lowest knowledge audience group impairs comprehension in the other groups [14]. The changes in interaction strategies are not simply becoming better or faster, but changes in how the overall system interaction is viewed and how individual tasks are mentally assembled.

Satisficing plays a major role with how people interact with a system. "Users will make sense of a document in whatever ways best suit their needs and purposes at the time; they will usually settle for the first solution they find that satisfactorily answers their particular problem at that point in time" [12, p. 69]. This also means their performance changes can occur in jumps as they eventually are forced to replace a method with worked with a new method which handles a broader set of circumstances. As example of satisficing, consider a situation which many of us would claim is simple, but which was found to be otherwise: looking up citation formatting in a writing handbook. Students stopped at the first piece of information which looked good and never evaluated if the format they had picked fit their needs [13]. From a design aspect, the text seemed very straightforward. From a complex usability aspect, the information was not presented in a manner which supported the student in a problem-solving situation.

These same issues are seen in using electronic medical record systems. There is a huge amount of available information and each person who interacts with it needs a different subset of that information. As a result, how they interact with it differs by audience. "A human/computer interface has to be ergonomic (coherent, concise, reactive, structured and flexible) and customizable to and by the user. It must automatically adjust its look and feel to suit the requirement of individuals or groups of users" [10, p. 220].

3.2 Complex Situations Revolve around Decision Making

The operator-based HCI psychology research has defined a good set of guidelines on how to effectively communication simple information. However, those same guidelines can work at cross-odds with communicating complex information. The question is how accepted design practices scale for a complex application such as C2PC. Highly effective interactions with these complex applications are more than simply a sum of a sequence of individual tasks. Instead, the interaction requires a constant back and forth between multiple interface elements with the current best answer constantly changing in the dynamic information environment. Examining a system for genome data analysis, Dicks [9] said:

The product was better designed than most such systems, and it included many of the necessities for successfully completing such work that Mirel [18] points to, including means for nonlinear, multipath studies; for collaboration among disparate professional groups….Nonetheless, while each of the smaller parts seemed to work quite well, it proved extremely difficult to determine how useful the system was in an overall sense (p. 209).

In these situations, the information is essentially dumped before the user, who was left to figure out what to do with it. Unfortunately, people are very poor at figuring out what to do with it since they inherently strive to minimize cognitive load and effort. With a design based around efficient performance of simple tasks, the HII imposes a high cognitive load since the person is dealing with both complexities of their goals, and the information in the situation, and the mental transformation required for interacting with the display.

Ensuring that information is usable rather than a dump, means that usability testing in complex situations needs to focus on knowledge-based measures which measure the internal thought processes. Typically this involves asking the person to describe the current situation at various points and asking what information led to those conclusions. Performance-based measures (task time, click counts, etc.) are easy to collect but provide little useful information [5].

3.3 HII Design Issues and Interaction Complexity

As Mirel says [17, p. 233], “Complex tasks and problem solving are different in kind not just degree from well-structured tasks.” Complex information is multidimensional. There are no simple answers and there are a dynamic set of relationships which change with time and in response to situational changes [3].

Ensuring proper information salience should be a significant focus of design teams when dealing with complex information. It is easy to create designs were all of the information receives the same level of emphasis, but this results in designs which are difficult to interact with. The user now has to read an excess of information (since the irrelevant or lower importance information is not flagged) and mentally reduce the information. Experts in an area can accomplish this, it’s one of the distinguishing characteristics of an expert, but both beginner and intermediate uses cannot. As a result, the person’s contextual awareness suffers [4] and they are unable to build the necessary information relationships to fully grasp the situation [6].

Interestingly, the person may understand what is happening in the system. In the C2PC study, the Marines were able to give a basic explanation of how their task operated within C2PC. However, when given specific tasks to accomplish, they were unable to effectively interact with the interface to perform that task.

4 Conclusion

Traditionally, HCI addressed highly structured situations, with a basic goal of efficiently completing a task. But many interaction situations are no longer highly structured; instead, they have shifted to complex situations revolving around information seeking, problem-solving, and decision making, which call for developing a high level of contextual awareness. This type of work needs a deep understanding of HII, which is essential to designing interfaces that communicate complex information.

Poor HII is reflected in designs which fail to support effectively rolling up the individual tasks into the complex interactions that people must perform. Usability tests show basic tasks can be accomplished with minimal problems. But any complex system must provide support for people working with solving open-ended, unstructured, complex problems which typically require extensive and recursive decision-making or problem solving [2]. Any complex information system involves complex situations; design teams must consider what factors influence how people perceive the information in their contextual environment and then build on those perceptions to enable the selection of relevant information to support judgments, decisions, and actions.

This paper looked at a study of C2PC and extended the ideas to other systems. A fundamental interaction problem with the design of complex systems is that people often fail to both effectively and efficiently work with the system. Across these complex software platforms, performance of basic operations tend to be very quick and simple to learn. But the learning curve for working within a complex environment is much more than simply knowing what buttons to push. An open research question revolves around designing easy to learn complex systems that remain easy to use and that support a multitude of interaction styles. The interaction moves from "doing" the button pushing to "thinking" about which buttons to push, in which order, why to push them, and how to support evaluation of the result of the push. Software does not exist to provide people with button pushing experiences; it exists to help people understand a situation and make decisions which influence that situation.

Acknowledgements. The C2PC research was performed while the author was on an Office of Naval Research Summer Facility Fellowship at Naval Undersea Warfare Command, Newport, RI.

References

1. Albers, M., Still, B. (eds.): Usability of Complex Information Systems: Evaluation of User Interaction. CRC Press, Boca Raton (2010)
2. Albers, M.: Complex problem solving and content analysis. In: Albers, M., Mazur, B. (eds.) Content and Complexity: Information Design in Software Development and Documentation, pp. 263–284. Erlbaum, Mahwah (2003)
3. Albers, M.: Communication of complex information: User goals and information needs for dynamic web information. Erlbaum, Mahwah (2004)
4. Albers, M.: Design for Effective support of user intentions in information-rich interactions. Journal of Technical Writing and Communication 39(2), 177–194 (2009)
5. Albers, M.: Usability of complex information systems. In: Albers, M., Still, B. (eds.) Usability of Complex Information Systems: Evaluation of User Interaction. CRC Press, Boca Raton (2010)
6. Albers, M.: Usability and information relationships: Considering content relationships when testing complex information. In: Albers, M., Still, B. (eds.) Usability of Complex Information Systems: Evaluation of User Interaction. CRC Press, Boca Raton (2010)
7. Albers, M.: Design and usability: Beginner interactions with complex software. Journal of Technical Writing and Communication (in press)
8. Albers, M.: Human-Information Interaction and Technical Communication: Concepts and Frameworks. IGI Global (in press)
9. Dicks, S.: Designing usable and useful solutions for complex systems: A case study for genomics research. In: Albers, M., Still, B. (eds.) Usability of Complex Information Systems: Evaluation of User Interaction. CRC Press, Boca Raton (2010)
10. Ghedira, C., Maret, P., Fayn, J., Rubel, P.: Adaptive user interface customization through browsing knowledge capitalization. I. J. Medical Informatics 68(1-3), 219–228 (2002)
11. Hollnagel, E.: CREAM—Cognitive reliability and error analysis method (2006), `http://www.ida.liu.se/~eriho/CREAM_M.htm`
12. Howard, T., Greer, M.: Innovation and Collaboration in Product Development: Creating a New Role for Usability Studies in Educational Publishing. In: Albers, M., Still, B. (eds.) Usability of Complex Information Systems: Evaluation of User Interaction. CRC Press, Boca Raton (2010)
13. Howard, T.: Unexpected Complexity in a Traditional Usability Study. Journal of Usability Studies 2(3), 189–205 (2008)
14. McNamara, D.S.: Reading both high and low coherence texts: Effects of text sequence and prior knowledge. Canadian Journal of Experimental Psychology 55, 51–62 (2001)
15. Mirel, B.: Analyzing audiences for software manuals: A survey of instructional needs for real world tasks. Technical Communication Quarterly 1(1), 15–35 (1992)

16. Mirel, B.: Advancing a vision of usability. In: Mirel, B., Spilka, R. (eds.) Reshaping Technical Communication, pp. 165–187. Lawrence Erlbaum, Mahwah (2002)
17. Mirel, B.: Dynamic usability: Designing usefulness into systems for complex tasks. In: Albers, M., Mazur, B. (eds.) Content and Complexity: Information Design in Software Development and Documentation, pp. 233–261. Erlbaum, Mahwah (2003)
18. Mirel, B.: Interaction Design for Complex Problem Solving: Developing Useful and Usable Software. Morgan Kaufmann, San Francisco (2003)
19. Redish, J.: nderstanding readers. In: Barnum, C., Carliner, S. (eds.) Techniques For Technical Communicators, pp. 15–41. Macmillan, New York (1994)
20. Redish, J.: Expanding usability testing to evaluate complex systems. Journal of Usability Studies 2(3), 102–111 (2007)
21. Ruskin, J.: The Humane Interface: New Directions for Designing Interactive Systems. Addison-Wesley, New York (2000)
22. Schriver, K.: Dynamics in Document Design: Creating Texts for Readers. Wiley, New York (1997)
23. Shearer, H.: An Activity-theoretical approach to the usability testing of information products meant to support complex use. In: Albers, M., Still, B. (eds.) Usability of Complex Information Systems: Evaluation of User Interaction. CRC Press, Boca Raton (2010)
24. Sutcliffe, A., Karat, J., Bodker, S., Gaver, B.: Can we measure quality in design and do we need to? In: Proceedings of DIS 2006: Designing Interactive Systems: Processes, Practices, Methods, & Techniques, pp. 119–121 (2006)
25. Whitburn, M.: The ideal orator and literary critic as technical communicators: An emerging revolution in english departments. In: Lundsford, A., Ede, L. (eds.) Essays in Classical Rhetoric and Modern Discourse, pp. 230–248. Southern Illinois University Press, Carbondale (1984)