Week 2 Summary
Interface a la Mode
Critical Review of Raskin's *Meanings, Modes, Monotony, and Myths*

James F. Lofton
HCI594–Capstone
Instructed by: Dr. Adam Steele
DePaul University
Chicago, IL USA
1.23.2007

ABSTRACT

Jef Raskin is a well known figure in the CHI field as one of the original contributors of the Macintosh. In his book "The Humane Interface" Raskin lays out the core of his interface design philosophy. Chapter 3 discusses the concepts of Mode, Monotony and the myth of beginner/expert users of computing systems. Modes are a bad thing according to Raskin. He realizes that there are problems associated with modeless interfaces and his concept of Monotony is meant to mitigate those. His arguments are compelling and his examples engaging, but they don't always seem to apply to the concept he is discussing. Modes are a fact of life not just for interface design. As long as people find metaphors useful for interface interaction, then we will have to learn to better manage users perceptions of modes.

1. Modus Operandi

Raskin defines modes for an interface as when the interface responds to gestures and 1) the state of the interface is not the user's locus of attention and 2) the interface has more than one response to the gesture depending on its current state.[1] It is the first part of this definition which frames the rest of Raskins' discourse. Which of course comes to the conclusion that modes need to be eliminated from interfaces. This is, shall we say, Raskins' modus operandi, define the terms in such a way that he is always correct. This definition is by its very structure setting modes up to be a bad interface element.

A better definition of mode is simply a designated condition or status, as for performing a task or responding to a problem. [2] How an interface handles the display and changing between modes is what causes problems not modes themselves. Raskin is right in pointing out that mode errors typically result from the user not noticing or remembering what mode the interface is in currently.

2. Mode confusion

Raskin quotes Don Norman as explaining that inadequate feedback is the source of mode errors. [3] However, he posits that it is the user's locus of attention that is the problem. He then uses the example of CAD software that while otherwise sporting a well designed interface has a problem with its trace tool. He and other users forget to switch back to the arrow selection tool after using the trace tool. He insists that the problem is where you are focusing, not the fact that the icon representing the trace mode is almost identical to the normal arrow selection icon. This to me this is a perfect example of inadequate feedback. Technically, Raskin is agreeing with Norman as your locus of attention is not drawn to the fact that you are still in trace mode. If anything it is fooled by the cursor's size and similarity to the selection tool. The feedback is insufficient. At this point, he begins to make his case for experts not being immune from these errors and thus they are no different from beginners.

Later in the chapter he refers to Norman's description and use of term affordance and then proceeds to explain how bad the Bay Area Rapid transit (BART) machine is. I have no clue how this relates to affordance (users clearly know what are buttons, they just don't know which to push), nor does it relate to the problem of mode as far as I can tell. It does somewhat support his notion of user focus being a problem as there is an LCD screen that is not noticeable, but the real problem is a process/order of task problem and not an affordance or mode issue.

3. Monotony is tyrannical

It is no surprise that Raskin advocates a modeless interface or one where modes are managed in such a way that the user does not need to be concerned with them. Taking it further Raskin realized that there may still be confusion for users where two operations may end up with the same result. According to him, this is also something that is undesired. So he proposes a monotonous interface, where one and only one action can result in one and only one result. He is overly concerned that users will be confused by having performed a task one way and then discovering they could do it another way and then being unable to remember either way at some later point. He tries to clarify how this does not mean that the same content can not be arrived at in different ways, just that no command should have more than one gesture that triggers it. I am not sure what the distinction is. This is very academic as trying to come up with situations where by someone would be able to build complex content with out having to use the exact sequence as provided in the monotonous interface is absurd. In fact I find this to be rather tyrannical, as it means the interface designer has determined what is best for everyone, either you like it or you are wrong. This is all building towards his final supposition that there is no such distinction between experts and beginners.

4. We don’t need no stinking experts

According to Raskin there is no useful reason to design alternatives in an interface to meet the needs of an expert verses those of a beginner. Experts will have the same problems as beginners when it comes to modes. He goes on to describe failed attempts to automate change in application behavior for expert users. I totally agree with this sentiment, as I absolutely hate Microsoft’s menus that adapt to my “use” by hiding other menu items. However, this is getting way beyond simple things that can be done to accommodate expert users such as keyboard shortcuts. It is also easy to pick on Microsoft’s poorly implemented attempts at making interface improvements. This is like pointing out how badly American cars are made.

Raskin maintains that if you do not know every function of an application you are not really an expert user. As you will be a novice when it comes to using those features for the first time even if you have used other parts of a program for many years. This is a classic example of Raskin being right in one technical sense but being wrong in practice. I am an expert user of photoshop because I have a background in design theory and know enough of the program to be proficient. If you took away the keyboard shortcuts I would be pissed. Do I know every feature, filter or menu item? No. Yet, I do actually know how to prepare images for both online and print with photoshop. I know the features required to accomplish both, but I do not know all the features, nor do I ever need to until it becomes part of a job. Sure it is hard to draw the line of what makes an expert, but that does not mean you can’t differentiate them from novice users.

It is this very argument that designers can’t build interfaces to automatically distinguish between expert and beginning users as it is too complex an analysis that also belies his modeless/monotonous interface. How can a complex interface be modeless when the very task it may be trying to emulate from the real word probably involves modes? For instance, I draw with a pencil and make a mistake. I switch my pencil to eraser mode by turning it over or switching to the eraser. How else is this accomplished virtually without also using modes? A drawing program doesn’t require modes simply because it has tool bars. It requires modes because drawing in a virtual environment is as complex as it is in the real world.

5. Conclusions

Raskin was a champion of easy to use computing devices that are approachable and usable. His insistence that modes are to be avoided as much as possible in interface design is a misguided conclusion born out of overemphasis on simplicity. Simplicity is good, and often interface designers do make complex problems out of ones that should be simple. However, when a problem is complex sometimes a complex solution really is the best answer. He uses an example of a car radio in his attempt to show how modes are a problem. He shows how having 32 preset buttons is better than having seven. Yet the radio must still have mode controls... as how do you tune into a station not on preset? How does one switch to AM from FM? If there are 32 buttons on the radio face, then other buttons or knobs must be multi-modal to handle the task of choosing the station to set the preset.

The main problem with Raskin’s modeless and monotonous interface is that he is trying to solve problems which really don’t need to be solved with doing things only one way… his way. It is presumptuous that we can force efficiency on users. While we may empirically show that having no modes in a certain situation is more efficient that does not translate into user satisfaction as directly as Raskin implies. In the end, as interface designers we do need to pay close attention to the display of modes and make them easily distinguishable. We also need to be careful when adding expert functionality, in that it does not create unnecessary complexity.

References

1. Raskin, Jef. "*Ch3: Meanings, Modes, Monotony and Myths* " The Humane Interface, Boston MA, Addison-Wesley Professional, 2000
2. "mode." Dictionary.com Unabridged (v 1.1). Random House, Inc. 20 Jan. 2007. <Dictionary.com http://dictionary.reference.com/browse/mode>
3. Norman, Don. The Design of Everyday Things, New York NY, Doubleday, 1988