Insights from User Interface Literature

The following is a summary of ideas and concepts relevant to this project from recent literature on user interface design supplemented by data from a few older ergonomic-oriented references. Words that are underlined are taken from one or more of the references as significant. While the literature is oriented mostly to software design, many of the principles apply to hardware design as well. Paragraphs with the "➥" symbol are key conclusions for this project.

The primary books reviewed are: The Humane Interface *(Jef Raskin, 2000),* About Face *(Alan Cooper, 1995),*The Design of Everyday Things *(Don Norman), and* The Art of Human-Computer Interface Design *(edited by Brenda Laurel, 1990),* Bringing Design to Software *(Terry Winograd, 1996). In addition, more limited data were gleaned from:* Human Factors in Product Design *(Cushman and Rosenberg, 1991), and* Industrial Design in Engineering *(Flurscheim, 1983), and* Ergonomic Design for People at Work *(Eastman Kodak, 1983).*

Bolstering the Rationale for This Project

In many ways, the design principles espoused in the literature confirm the motivations and goals that underpin this project. It is often noted that consistency (standards) is a good thing. This is primarily noted within a single hardware or software product, but also applies across products. Standards and consistency help develop habits. They allow for some tasks to be done unconsciously, to not interrupt the flow of thought; ideally, the interface disappears. There exist population stereotypes, which is the type of behavior that people expect in objects; an example is the idea of 'up' on a switch meaning on, which is not true of all national populations. People will be more successful in setting up peripheral products if the connectors and cables are labeled with standard icons, colors, and labels (Ease of Use Roundtable).

While one could solve each problem in a different way, that would introduce much too much complexity. What is needed is a few solutions that well-solve many problems (and doing this is hard).

When inconsistency exists, problems result. Examples include the danger of reversing car brake and accelerator pedals, or the controls on remotely-piloted model airplanes, which force a task to become conscious that needs to remain unconscious to be timely and avoid crashes. When indicators are lacking, people will make errors by acting in accord with the wrong mode (though the same problem can occur if people are not focused on the indicator even if it is present).

The idea of devices going to sleep was applauded by several authors, as was the functionality that 'hibernate' provides.

Relation to Past Designs

There is some merit to sticking with past designs, for consistency, but as the task becomes more common, the reasons to consider change increase. Cooper is more emphatic, stating that only minimal heed need be paid to past products; design should be all about the future. Bad designs should not be maintained.

➥ For power controls, we should continue to use the most popular interface elements on current devices except in those cases in which it clearly causes confusion. The bar for changing design is higher for interface elements that are universal (as power controls are) than it is for software controls specific to a particular application.

Approach

For Raskin, an interface is "the way that you accomplish tasks with a product—what you do and how it responds", and it is 'humane' if it is "responsive to human needs and considerate of human frailties". For the customer, the interface *is* the product.

For Cooper, one should start with the users goals, before considering specific tasks that implement these (though others say that task analysis is determining what people want to do which edges back to goals). As "few users are consciously aware of their goals", ferreting this out can be challenging, but is necessary.

It is critical to take the user's perspective during key parts of the design process; ideally this means asking or watching real users, but can also include exercises to place oneself in the user's stead and assessing a situation or design. **Designers are not typical users**. There may also be considerable variations among users, so that one needs to find what is common. Most authors believe in iterative design processes which include user testing as part of the review/redesign. For example, one listing of design stages is Product Definition, Research, Brainstorm, Generate Design Solutions, Analyze, Prototype, Test, Redesign, Implement. It is important to bring people from all disciplines into the design process. User Centered Design is the current preferred term for this overall approach.

➥ Writing the documentation in advance of creating the product is a key way to get better designs. To implement this, we developed the "Instructions for Powering your PC". In addition, a document should be prepared that outlines the goals and tasks that users bring to power controls, and a set of scenarios developed for different types of products, users, and use contexts.

A "population stereotype" refers to the fact that "people expect things to behave in certain ways when they are operating controls or when they are in certain environments". As an example of how these can vary, in many parts of the world such as the U.S., people expect a toggle switch for a light to turn on when pushed up; in the UK, a toggle switch is usually pushed down for on by convention.

➥ We need the feedback from people in countries around the world to identify any relevant population stereotypes which differ from those in the U.S.

Design Principles

As for general design principles, many authors applaud simplicity. Unnecessary complications tend to make interfaces more difficult to use. Another principle is to take advantage of affordances—this is a powerful idea which is that some entities have natural actions one can take with them that are inherent and intuitive, as that a knob can be rotated, a toggle switch flipped, etc. When designing interfaces it is important to notice how often an action is taken, not just the fact that it is sometimes taken, so that rarely-used actions don't clutter up menus, or and don't rank before commonly used ones.

➥ The basic interface should be the simplest one which meets the needs of the majority of devices and users. There will be exceptions which require exceptions or complexity, but these will be relatively few.

It is critical to distinguish between several different concepts that can explain a product's operation (Cooper):

➥ The "Implementation Model" is the specific internal details of an application or product. For example, a hard disk drive is arranged with pointers, sectors, and cylinders which the user need not have any awareness of.

➥ The "Conceptual Model" or "Mental Model", which is what user imagines is going on. In the disk example, it includes files being "inside" of folders or directories which is not literally true.

➥ Possibly different from the user's model is the "Manifest Model" which is what the software or device actually shows to the user. This difference is possible because user's may ignore or misunderstand (possibly for good reason) the manifest model. Another source refers to the "product's functional capability" for its actual operation.

There is no need to burden the user with the internals, and in fact in good design practice the user interface will be designed before the internals exist.

➥ A goal of this project should be to make the manifest and mental models as consistent as possible, masking the various underlying implementation models. Hiding internal terminology is the most apparent aspect of this.

"An agent is accessible if a user can predict what it is likely to do in a given situation on the basis of its character" (Laurel, p363).

➥ While this refers to software agents, it does raise the question of whether devices which have or lack particular capabilities should indicate that in hardware or software (e.g. whether a device can go into a sleep mode).

One author argues that the next leap in interface technology after the Graphical User Interface is 'cyberspace'—a three-dimensional animated interface. This will be multi-sensory, and provide for and require additional types of interface elements.

➥ This suggests that power indicators for the cyberspace context will need to be developed. For example, as entities in cyberspace have exteriors more in common with human faces than they do with current control panels, then standard elements, behaviors (in sight, sound, etc.) that model power states will be needed.

Metaphor, etc.

The idea of metaphor is one of the more controversial in the user interface world. Cooper and some others believe they are bad; whatever initial benefit exists isn't worth the limitations and "dead weight" they impose. Others believe that using metaphors does not mean adhering to them slavishly; to some degree they are always present and it is just a matter of choosing them well and using them to the right degree. They can be "cognitive aids to users and … aids to creativity in designers" (Laurel). Metaphors should provide some structure, to hang ideas on, as well as lend themselves to representation, through sight or sound.

As metaphor is extended to entities which take action independently on behalf of users (e.g. software agents), anthropomorphism may be used. This is reasonable, as as with metaphor, it can be used selectively. Two aspects are present: "responsiveness" to wants, and "capacity to perform actions".

Raskin rejects the idea of an intuitive interface; rather, he believes they are all just familiar or habitual and thus feel intuitive.

Related to the idea of metaphor is the paradigm used to create the interface (Cooper). These include:

- Technology, which presents the 'raw' internal implementation to the user;
- Metaphor, which uses an external reference to guide the product design;
- Idioms, which are small, clever concepts; and

- Global, which imposes a single metaphor as much as possible (Cooper sees global metaphors as a form of insanity).

Another proposed alternative to metaphor is the "well thought out unifying idea".

➥ The ideas of 'on' and 'off' as well as the existing power symbols are entirely idiomatic. The metaphor of sleep and the moon symbol are used narrowly (extending the metaphor could suggest that a device that is on is 'alive' or that one that is off is 'dead', neither of which we are trying to do).

Modes

Another controversial idea is the idea of modes, in which the product will respond very differently depending on which mode it is in (examples include early text editors with 'insert' and 'command' modes, and paint programs with modes like draw, select, or zoom. Personalizing software (or hardware) can be seen as a form of a mode—usually not a visible or standard one—and so is not recommended. Some authors dislike modes intensely; others see them as OK when used appropriately.

One aspect of modes that is not popular is distinct 'beginner' and 'expert' modes which lead to different behavior. Most people will be intermediate users—neither beginners nor experts—and the design should be optimized for them. While programs should learn from user behavior, the interface should not be changing significantly based on this, but adapt more subtlely. Programs/devices should be able to remember things and learn when appropriate.

One way to avoid mode problems is to not reuse commands between modes. One author asserts that when a physical action is required, mode problems disappear.

If mode indications are not familiar, they may be distracting.

➥While some authors decry modes, they are inherent in power status—limiting the available capabilities of a product in sleep or off modes is required for the power reductions to occur. While utilizing modes for power status is unavoidable, but limiting their number (we propose just three for most devices) makes them less confusing.

Interaction/Transitions

Devices should go from off to fully functional in as little time as possible (to be appliance-like). Returning to the same state (hibernate) is also desirable. Confirmations that are routine become automatic and so lose their effectiveness. Confirming that a file save is OK to do when the previous version of the file is being replaced is an example of this. Similarly, explicit reporting that all is well is useless to do.

People take about 10 seconds to switch contexts, but when the perceive delays, any sort of sense stimulus can assuage annoyance, with sound a key example. When interruptions occur, people should be returned to the previous state (this is accomplished by system hibernation rather than a conventional off followed by reboot).

Routine actions should be streamlined (e.g. don't report dramatically that nothing is wrong). Feedback can be key, particularly if there is any delay involved. "Progress indicators" should report how much longer a lengthy task (e.g. downloading a file) will take, and provide a way to cancel it. A "splash screen" (something with marginal content displayed when a device or application is started) must be displayed immediately after initiation.

➥ Turning a device on or off, or putting it to sleep or waking it up may all be lengthy and so require some sort of progress indicator—visual and/or auditory.

With windowing systems there is a question as to whether a mouse click that changes the focus to a new window (and possibly a new application) should be used only for that context switching, or also used as a regular click in the window. While there is merit to each argument, discarding them is the safer route. The same issue applies to input device events that wake a system up from a sleep mode.

Graying out menu items that aren't available is a useful tool.

➥ For common power management functions, this should be done so that the overall structure remains more consistent.

An "accelerator" is "an additional, optional way to invoke a function from the keyboard… usually with a function key". These should only be used for common actions, should be shown on menus as reminders, and should follow standards (e.g. control-C for copy, control-V for paste, etc.). Better standards for these are needed. Regardless, for common functions, multiple ways to do them should be implemented.

➥ An accelerator for on/off seems unnecessary, but a standard one for 'sleep' may be appropriate for ease of use, and for devices which lack a dedicated sleep button. Some notebook computers use function keys for this, though the number of the function key varies from implementation to implementation. It might be worth having sleep or off functionality available by clicking on an icon on the screen, and/or on a pop-up menu easily available (e.g. by a right mouse click on Windows PCs).

Indicator Lights

When there is a known delay that the user will notice (such as from disk access, computation, etc.), it is essential to provide user feedback that the process is underway, and an indication of how far it is in the process or how much time remains. On current computers, this is accomplished by changing the cursor or providing a dialog box.

➥ The mechanism for providing such feedback is not appropriate for standardization, but the implementation of particular mechanisms may be. Text feedback can use the terminology identified in this project. For graphic feedback (e.g. a changed cursor), there four relevant cases: turning on; turning off; waking up; and going to sleep. This deserves further consideration.

For indicating status, colored indicator lights are the preferred mechanisms. There are standards that existed in the 1970s and 1980s for the meaning of different colors, but whether these are still in effect and if so, what standards they reside in, we don't yet know. A prominent example is a British Standard (4099) which addresses "color coding for lamp indicators". This has:

	British Standard 4099[a]	**"Widely held" associations[a]**	**"Population Stereotype"[b]**
Red	Danger – alarm; action needed	Alarm, critical, disabled, emergency, failure, stop	'stop' or 'danger'
Yellow	Caution — impending change	Marginal condition (caution), standby	'caution'
Green	"safety – proceed, equipment safe	Active, enable, normal, on, on-line, run	'go' or 'on'
Black		Off	

Note. One source said that most commonly, audio/video recorders use red lights for recording, green for play, and yellow for pause. [a]From Flurscheim (1983). [b]From Eastman Kodak (1983).

Yellow is listed as potentially problematic for color-blind people (curious since it is usually called red/green color blindness)[1]. There were counterparts in other European countries and in IEC Standard 73 which no longer exists but may have been incorporated into another IEC standard. While up to 10 colors can be readily distinguished, it is recommended to limit the number used to three. Color-blindness is said to affect 8% of males and 0.5% of females.

➥ Is color blindness a concern for the green/yellow/off color set? If red is added for errors does that raise concerns?

➥ The green/yellow/off color set seems to have no serious competition, as does the addition of red for errors/warnings. In addition, it seems prudent to reserve flashing for transitory states (transitions between basic power states) and/or for non-power-status information (e.g. receiving information, etc.). If flashing is used for transitions, then it is probably most important to show the state being entered which suggests flashing green for turning on or waking up, flashing yellow for going-to-sleep or turning off. Since red for an error doesn't show the power state, then an alternating red with green or yellow could do both.

One source stated that flashing is said to be the best mechanism for implementing "warning lights"; another that "displays that blink … imply urgency and excitement". Good flash rates for warning lights are said to be 3 to 10 times per second, with a minimum on-time of 0.05 second.

➥ Flashing with the goal of getting attention as for errors should probably be faster than flashing for indicating state transitions.

Icons

A good icon is "visually distinct … [and does] a good job of representing the appropriate concept" (Raskin). Icons may not translate well across cultures. A "Graphic symbol" is usually "an abstract or arbitrary symbol without obvious meaning" as opposed to a "pictographic symbol" or "pictogram" which depict familiar objects. Research has shown that icons developed with the participation of typical users have better effectiveness than those developed by product designers alone. Solid shapes are preferred to those with outlines.

Icons can be derivative of a metaphor (to a greater or lesser extent), or idiomatic.

➥ The moon symbol should be made up of just two basic arcs, and solid rather than an outline. Angling the moon might help distinguish it from the left parenthesis "(". Some testing of recognition of the power/standby and moon symbols should probably be done.

One source (Flurscheim) states that "The preferred position for symbolic labels is *above* the associated mechanical/electrical elements", but this probably refers to control panels with many elements and so may not be relevant to power control elements that are more isolated.

A Cautionary Tale

A particularly relevant user interface discussion is one by Don Norman (in "Bringing Design to Software") about his effort to improve and standardize the treatment (placement and function) of the power switch on Apple Macintosh products. Practical concerns of design and especially organizational difficulties thwarted the effort.

While many aspects of the Macintosh software and hardware interfaces are consistent,

[1] Cushman/Rosenberg states that for users with "normal color vision" use "red, yellow, green, white, blue". For when "some users have abnormal color vision, use:" red, yellow-green, green or white, cyan, blue". How to interpret this isn't clear.

> "the lack of standardization of the power switch seems bizarre. Some machines have it in the front, others in the back. Some have toggle switches, others have pushbuttons. Some machines do not appear to have any power switch at all. Users continually have trouble finding the switches."

A particular problem was models which placed the switch away from the power indicator, in a place that a disk eject button might be expected to be found (Macs don't need such a button so don't have one). This caused many people to press the power button by accident, often losing their recent data.

Norman determined that no one in the organization was really responsible for the issue. Design generally was distributed among four divisions. The consumer division placed a priority on price; the notebook division on power conservation; and the server division on protecting the switch from accidental use. Other corporate goals came into play, such as localization for international markets, safety issues and regulations, and accessibility for the disabled.

A single solution was elusive, even though many recognized the value of one. The switch issue was further complicated as user's were supposed to normally turn the machine on with a keyboard switch (which wasn't also for turning off). The keyboard switch is itself instructive:

> "In our current models, the keyboard power-on keys are labeled with a left-facing triangle. Why? Because the symbol does not mean anything! The symbol used earlier (a vertical line inside a circle) was not permitted because the European standards authorities insisted that it was reserved for hard power switches. The triangle has no meaning, so it does not violate any standards. Few people—European or American—are confident about the meaning of the vertical bar and circle (on and off respectively), let alone a bar inside of a circle (a toggled on-off), or a vertical bar inside a broken circle (toggled soft power), but the European standards committee is strict."

He continues, "The final proposal had a soft power key on the keyboard, labeled 'on/off' (translated … [as] appropriate". Also, "A policy of indicator lights was established, so that a user could tell whether the machine was on, off, or in energy-saving mode." Also, holding the power key for five seconds would cause an emergency power down.

One of the many barriers in the organization to addressing this was that user interface design was perceived as a solely software domain, with hardware being limited to industrial design issues.

➥ One solution not available when the controversy was brewing is for the power switch to engage hibernation on the assumption that this is commonly used. A shutdown to the non-hibernate off mode could still be accomplished from a menu selection.

➥ The ISO and IEC are international organizations, but Norman's view of them as at least oriented to Europe is widespread in the U.S.