Design Principles, Heuristics, Representation and Learning Curves

Robert Bobkoskie
rbobkoskie3@gatech.edu

1 DESIGN PRINCIPLES AND HEURISTICS

The following fifteen principles provide heuristics for interface design [1].

	Don Norman	Jakob Nielsen	Constantine & Lockwood	Universal Design
Discoverability	✓	✓	✓	
Simplicity		✓	✓	✓
Affordances	✓			
Mapping	✓	✓		
Perceptibility	✓*	✓		✓
Consistency	✓	✓	✓	
Flexibility		✓		✓
Equity				✓
Ease				✓
Comfort				✓
Structure			✓	
Constraints	✓	✓		
Tolerance		✓	✓	✓
Feedback	✓	✓	✓	
Documentation		✓		

Figure 1: Fifteen Principles of Design

1.1 Design Principles and Invisible Interfaces

This section will discuss three design principles that support the creation of an invisible interface. The relationship between how specific phases of each design principle bridge the gulfs of execution or evaluation will be explored. Recall the definition of the gulfs of execution and evaluation:

Gulf of Execution: *The distance between a user's goals and the actions required to realize those goals.*

1. **Bridging the Gulf of Execution**:
    a. *Make functions discoverable.*
    b. *Let the user experiment.*
    c. *Be consistent with other tools.*
    d. *Know your user.*
    e. *Feed forward, anticipate a user's actions*

[1] Joyner

Gulf of Evaluation: *The distance between the effects of actions to realize goals and the users understanding of the results. The user needs to evaluate the new state of the system following an action.*

2. **Bridging the Gulf of Evaluation**:
    a. *Give constant feedback.*
    b. *Give immediate feedback.*
    c. *Match the feedback to the action: subtle actions $\Rightarrow$ subtle feedback, significant actions $\Rightarrow$ significant feedback.*
    d. *Vary feedback: Auditory, visual, hepatic.*
    e. *Leverage direct manipulation.*

1.1.1 *Affordances*

The design of the interface affords, or hints at how to use it. Use of affordances can create an invisible interface for the user. It should be noted, that strictly speaking, affordances cannot be added, however signifiers can. A signifier can enable a user to interpret an affordance that may be unfamiliar to them. For this discussion, the definition of affordance will be loose, i.e., it will be assumed that users have an inherent knowledge of the design, thus the implied affordance.

To apply affordances to virtual designs, e.g., interfaces, the architect can:

- *Incorporate virtual buttons that imply user interaction by "pushing".*
- *Flip through virtual pages in a digital document by swiping your finger across the screen, much like turning pages in a physical book. A signifier, such as an arrow, or an autonomous page flip could be utilized to abet the user in discovering this feature.*

CONSIDER: An affordance for one person does not imply this affordance applies to all people. That is, affordances are defined by who the user is. Therefore affordances cannot be explicitly added to an interface. However, signifiers can be added to assist the user understand the affordance.

Bridging the Gulf of Execution: Affordances bridge the gulf of execution by way of *[1a, 1c]* in the *Gulf of Execution* list. By definition, the designer must know the user in order to implement affordances od add signifiers. Through knowing the user, the designer can also feed forward information to the user, by designing interfaces that anticipate user actions.

Bridging the Gulf of Evaluation: Affordances bridge the gulf of evaluation by way of *[2c, 2e]* in the *Gulf of Evaluation* list. Affordances match user feedback to user actions, e.g., a user pushes a button on a virtual interface and the button appears to depress. This is accomplished by direct manipulation, where the user interacts directly with the task.

1.1.2 *Tolerance*

Tolerance provides forbearance for user experimentation. Undo and redo, recoverability from error are components of tolerance that can enhance user discoverability, thus allowing interfaces to become more invisible for users.

Bridging the Gulf of Execution: Tolerance bridges the gulf of execution by way of *[1a, 1b]* in the *Gulf of Execution* list. By definition user experimentation is tolerated: undo/redo, recoverability from error conditions are concepts related to tolerance. Through experimentation, user discoverability is enhanced.

textbfBridging the Gulf of Evaluation: Tolerance bridges the gulf of evaluation by way of *[2b]* in the *Gulf of Evaluation* list. TO recover from error, design interfaces to provide the user with immediate feedback to notify them of the error. In this manner, a user will not stray too far down a path toward an unrecoverable state.

1.1.3 *Feedback*

Good feedback provides a user immediate, informative notification that's consistent with the state of the system. A succinct, informative error may not make an interface invisible to a user, but it can be argued that the interface is less visible with good feedback. Consider older versions of the Windows Blue Screen of Death. This message is not familiar to most users, is not concise and doesn't not provide information on how to fix the condition. Newer versions of Windows correct this feedback, thus making the interface more invisible for users.

Bridging the Gulf of Execution: Feedback bridges the gulf of execution by way of *[1b]* in the *Gulf of Execution* list. By providing good feedback, users can become more at ease with experimentation in the interface.

textbfBridging the Gulf of Evaluation: Feedback bridges the gulf of evaluation by way of *[2a, 2b, 2c]* in the *Gulf of Evaluation* list. Provide good feedback hits on many of the parameters that bridge the gulf of evaluation. In particular, providing a user constant, immediate feedback that aligns with user actions will bridge the gulf of execution for the user.

1.2 Design Principles and the Participants View

This section will present two design principles for creating interfaces that emphasize the participant view of the user. I'll describe how each relates to understanding not just the users abilities and thought process, but also the context of the user outside their interaction with the interface.

1.2.1 *Equity*

Equity provides the same means for all users. Segregation or stigmatization of users is avoided. Help all users have the same user experience by accounting for target audience of the user, e.g., interface support for novice vs. experienced users. To accomplish this, the designer must consider the participant view for all users.

1.2.2 *Structure*

Structure strives to design interfaces that are organized in a clear and meaningful ways that are recognizable to users. Interfaces are organized to help the users mental model match the content of the task. To account for the users mental model, the designer should consider the participant view of the user.

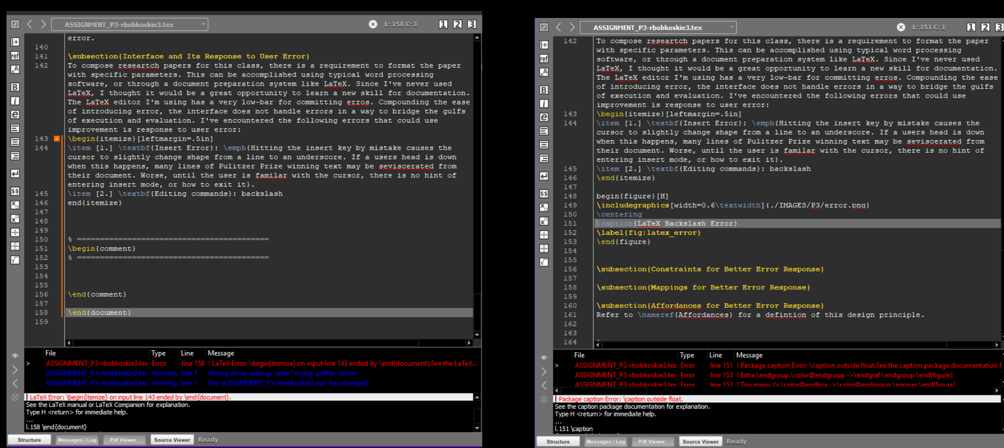
2 INTERFACES AND CONSIDERATION OF USER ERROR

This section will describe an interface that is intolerant of user error. Specific design principles, *[constraints, mappings]* will be proposed that can account for user error.

2.1 Interface and Its Response to User Error

To compose research papers for this class, there is a requirement to format the paper with specific parameters. This can be accomplished using typical word processing software, or through a document preparation system like LaTeX. Since I've never used LaTeX, I thought it would be a great opportunity to learn a new skill for documentation. The LaTeX editor I'm using has a very low-bar for committing errors. Compounding the ease of introducing error, the interface does not handle errors in a way to bridge the gulfs of execution and evaluation. I've encountered the following errors that could use improvement is response to user error:

1. **Insert Error**: *Hitting the insert key by mistake causes the cursor to slightly change shape from a line to an underscore. If a users head is down when this happens, many lines of Pulitzer Prize winning text may be eviscerated from their document. Worse, until the user is familiar with the cursor, there is no hint of entering insert mode, or how to exit it.*
2. **Editing Commands**: *LaTeX provides a rich set of features for accomplish the task for composing complex documentation. However, there is a level of familiarity that is required to navigate the myriad of error messages. Compounding the high probability for novice user to introduce error, the messages are not correlated with the error condition. In addition the highlighted portion of the document containing the error is incorrect and misleading to the users, see figure 2.*



(a) LaTeX Backslash Error: \enddocument

(b) LaTeX Backslash Error: \caption outside float

Figure 2: LaTeX Backslash Error

2.2 Constraints for Better Error Response

To better account for user error, constraints can be implemented to prevent the user from inadvertently entering insert mode. At the very least, immediate notification can be presented to the user that they have entered insert mode see figure 3 on page 6, which will be discussed further in Affordances for Better Error Response. Constraints may also be used to prevent, or suggest to the user that a syntactical error may have occurred, e.g., missing a special character such as a backslash "\".

2.3 Mappings for Better Error Response

To better address relation between the interface and the error. The system should speak the users language and follow real world convention to define the error. In other words, define the error more closely to the condition that cased it: e.g., for a backslash error, map that error to the correct location in the document with appropriate highlighting.
Mapping can also be used to provide better feedback to the user for the insert condition. The interface could provide immediate audible feedback to suggest the user entered insert mode, as well as continuous feedback the user is devouring characters while editing text in insert mode. A fitting sound might be that of a Pac Man chewing up dots [2].

2.4 Affordances for Better Error Response

Affordances can be introduced for improved error feedback to the user. Refer to Affordances for a definition of this design principle. Figure figure 3 shows affordances introduced to the LaTeX interface that could help the user identify if they have entered insert mode (Pac Man image) as well as the location of potential errors (Stop Sign image). As previously discussed, affordances are dependent on the users inherent knowledge of the affordance, e.g., a user knows of the game Pac Man and Stop Sign traffic symbol.

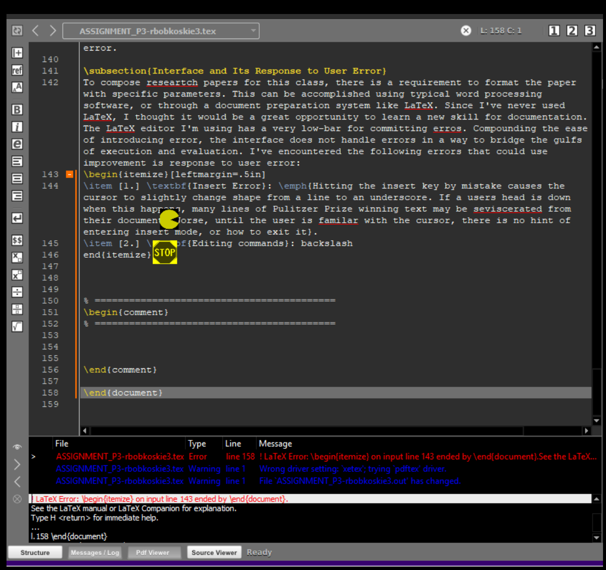


Figure 3: Affordances to Address User Error

[2]PacMan

3 INTERFACES AND USER SLIPS, MISTAKES, ERRORS

This section will discuss a game where the interface is prone to user slips and error. The section will conclude with a discussion about a feature that makes the interface challenging, but neither a slip nor a mistake.

3.1 User Slips

In football, player slips occur often. One example is an interception thrown by the quarterback [3]. An interception is not the intended result of the quarterbacks action, but is a slip do to some misinterpretation of the current state of play by the quarterback. This same event can occur in a football video game, where the user playing the role of quarterback throws an interception. This often happened when I was playing an older Mattel gaming system, where the user interface was a disk and four buttons. I often knew where I wanted to pass the ball, but would hit the incorrect button, causing the throw to go to the wrong location and be intercepted. A better user interface would be less generic and provide fewer, or even a single button to perform tasks.

3.2 User Mistakes

In football, player mistakes occur often. One example is holding [4]. A player commits a hold when they know a defender must be stopped from making a play, however they cannot do so within the rules of the game. To accomplish this task, the player will commit to the mistake (holding penalty) to accomplish the task of stopping the defender. Bringing this analogy back to the video game, I was in a similar situation do to the movement of a player not corresponding the direction implied on the screen. That is, the player always faced the same way, independent of the direction they were moving. This made playing the game susceptible to mistakes, as I could not infer the players movement based on the graphical display. To enhance my mental model of this event, a better representation would match the graphic image, e.g., position of the player to the players direction of movement.

[3] Interception

[4] Holding

3.3 Football: What Makes the Game Challenging

There are many ways to illustrate the challenges of playing football. One example is the decision to go for the first down on fourth down. A team has four attempts to move a football ten yards. If successful, the team resets to first down and gets another four attempts. The nuance is that the team has the option on the fourth down to try to make the ten yards or surrender the ball to the other team. This decision is sometimes based on probability, gut-instinct or a combination of both. Many factors influence this decision and the outcome is often second guessed.

4 REPRESENTATIONS

Good representations create effective mental models for the user. This section will present interfaces that use both good and poor representations.

4.1 Interface: Good Representation

An example of an interface with a good representation is show in figure 4a.

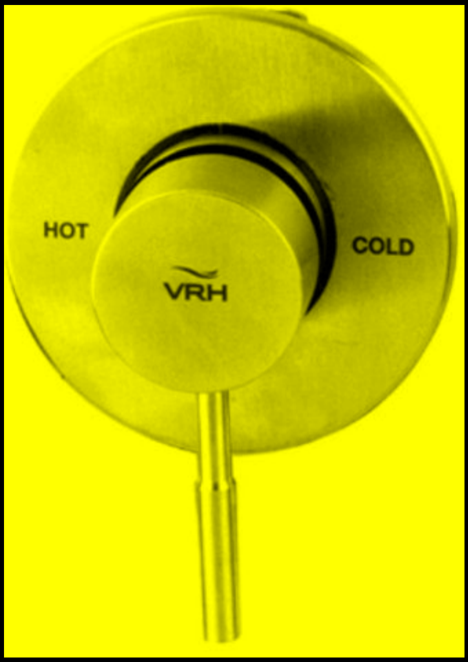
- *The interface makes relationships explicit. A user can infer the behavior of positioning the single central lever to the **HOT**, **COLD** indicators.*
- *The interface excludes extraneous details. There is only a single lever with two indicators presented to the user. The interface could be even more austere by eliminating the **HOT**, **COLD** marking. However, it could be argued that not enough information is presented, increasing the gulf of evaluation for the user.*

4.2 Interface: Poor Representation

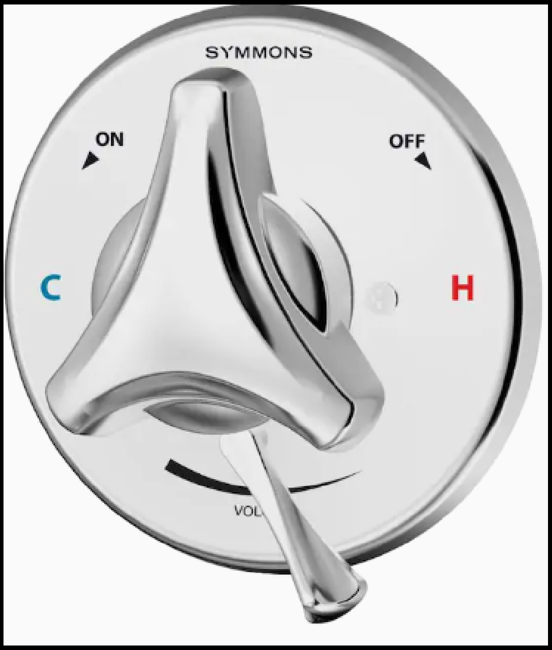
Figure 4b shows a poor representation for a shower interface.

- *The interface does not make relationships explicit. The 3-pronged shape for the central controller is confusing and difficult for the user to infer its use.*
- *The interface does not bring objects and relationships together. The relationships between the **ON**, **OFF** and **C**, **H** indicators is confusing. It's not clear to the user how to direct the central lever to deliver hot/cold water.*

5 APPENDIX



(a) A Good Representation for a Shower Interface



(b) A Poor Representation for a Shower Interface

Figure 4: Shower Interfaces