Assignment P1:
CS6750 Spring 2019

Dan Higgins
Daniel.higgins@gatech.edu

Introduction

In this assignment, we will be investigating a number of Human-Computer interface (HCI) topics. In the first section, we will be focusing on design principles related to three views of the user (Processor, Predictor, and Participant) which facilitate creating an interface that is both useful and usable.

In the second section, we will delve into the import design topic of feedback cycles. And look at how this ubiquitous concept helps drive many critical design decisions.

We analyze these topics and apply them to specific problems keeping in mind that "Two of the most important characteristics of good design are discoverability and understanding. Discoverability: (Helping us answer the questions) Is it possible to even figure out what actions are possible and where and how to perform them? Understanding: What does it all mean? How is the product supposed to be used? What do all the different controls and settings mean?" (Norman, 2013, p. 3)

.

Introduction to Principles (from Lesson 2.1)

Question 1

As a Georgia Tech OMSCS student, I regularly use Udacity for viewing lecture materials for my courses. The Udacity interface works well as an intermediary for achieving many learning tasks but does have some weaknesses which could be improved. In this section, we will be looking at how we might identify these weaknesses and make some suggestions for improvements.

Processor Model

To gain some insight into the Udacity interface let us look at the interface from a processor model perspective. "If we are designing with this role in mind then our main concern is that the interface fit within known human limits. These are things like what humans can sense, what they can store in memory, and what they can physically do in the world." (Joyner, 2017, Lesson 2.1) The processor model finds its origins in the psychology of behaviorism and looks only at those things that we can observe and physically measure.

When looking at the Udacity interface from the Processor view. We might be interested in how quickly we can achieve a specific task or how easy a button is to see based on its color, size or location in helping to accomplish a task. Let's look at a concrete example of Searching in Udacity. If your task is to "search" on a subject of interest, you can navigate to the search page (the interface provides a helpful search icon), type your search criteria, select the search button. The interface returns a search results list. The list contains information about the searched concepts and the locations within the video where the keywords are found, but the user is required to remember the place in the video where the material can be seen this is often a less than optimal solution when reviewing numerous results.

Predictor Model

In the predictor model, our concern is still with one user and one task but "We want them (the users) to be able to predict what will happen in the world as a result of some action they take. So we want them to be able to map input to output." (Joyner, 2017, Lesson 2.1)

Let us continue exploring our Udacity search example but from a predictor view. If you select an item from the search results list, you are redirected to the beginning of the lecture concept that contains the searched data. If you, however, select the back button from this new location you would expect to be returned to the search results list from where you just came. This, however, is not the behavior in the current system. Your prediction would be wrong.

Suggested Improvements

In looking at the different models, the process model might suggest off-loading the cognitive responsibility of remembering the video location to the interface. By starting the concept video close to where the search term was found in the video transcript. While the predictor model may suggest modifying the behavior of the back button to return to the search results list after reviewing an item on the list thereby better meeting the user's expectations.

Question 2

Question 2 asks us to explore the ***Participant view*** of the user looking at applications that are used in multiple contexts. "While the processor and predictor views emphasize only the interaction of some user and some interface, the participant view also looks at the interaction of both within the context of a larger system." (Joyner, 2017, Lesson 2.1)

Audible.com provides a service for discovering, purchasing and listening to Audio Books the service has multiple modes of delivery and can be used in numerous contexts. For example, there is a cellphone application which I use while jogging and when I'm driving in the car. Audible also provides an interface for Amazon Echo Devices which I use when I am listening at home often while doing odd jobs around the house. In addition to these modes of delivery, there are applications for computers, tablets and smart TV's.

Audible has spent considerable design efforts on having their application be accessible in many different contexts. However, this is not to say that there are not areas of opportunity for improvement. Let us look at a few challenges to using the app in specific contexts.

Constraints or challenges to using the app

Listening while Jogging: I enjoy using the Audible App while out for a run I am often running on rail trails in my area where cell reception is weak or not available so I do appreciate that the books can be downloaded to the device. But one of the challenges of listening while running is that the phone is usually in a pocket or not easily accessible without stopping. Thus, functions like changing the book you are listening to or advancing the chapter of a book are not easy to achieve in this context.

Listening while Driving: I also enjoy listening to books while driving in this context streaming content is typically not an issue and since the phone can be placed on a dashboard mount seeing the screen is also convenient (But potentially distracting). The screen does provide additional useful information including the chapter number you are currently listening to and the time remaining in the current chapter. However, this context still suffers from the lack of ability to execute more advanced functionality like selecting and then listening overviews and recommendations of new books.

Listening at home: Although I do still enjoy reading books when time and opportunity permits I also like the convivence of listening and freeing my hands and eye to attend to other tasks while at home. This is where the Alexa App is convenient. I can esimply say "Alex Play Educated" (or any book title that is in my Audible library), and the book will start to play I last location it was stopped this includes advancing to the place I may have left off in my Kindle reader if I was also reading the book in addition to listening. With Alexa, you do get a rich set of voice commands, but the computer or tablet application is still required to do many of the more interesting and socially engaging aspects of the Audible experience. This includes numerous gamification aspects of the app like listening levels (i.e., level up) or badge collection awards.

Let us turn to a few areas of opportunity that would help the user while in different contexts.

Recommendations to overcome constraints

A first and maybe obvious step to overcoming constraints on the phone App would be to enable Alexa directly into the Phone Application. Add this is highly likely as Amazon owns both of these systems. But additionally, advancing the voice interface to a “conversational” interface that is context-aware would greatly enhance the capabilities. Alexa in its current form is still a “command driven” interface. The conversational interface would potentially know through GPS my location and if I were driving or through an accelerometer if I was running. It could use this information plus information from previous interaction to better anticipate my needs as a user.

Feedback Cycles (from Lesson 2.2)

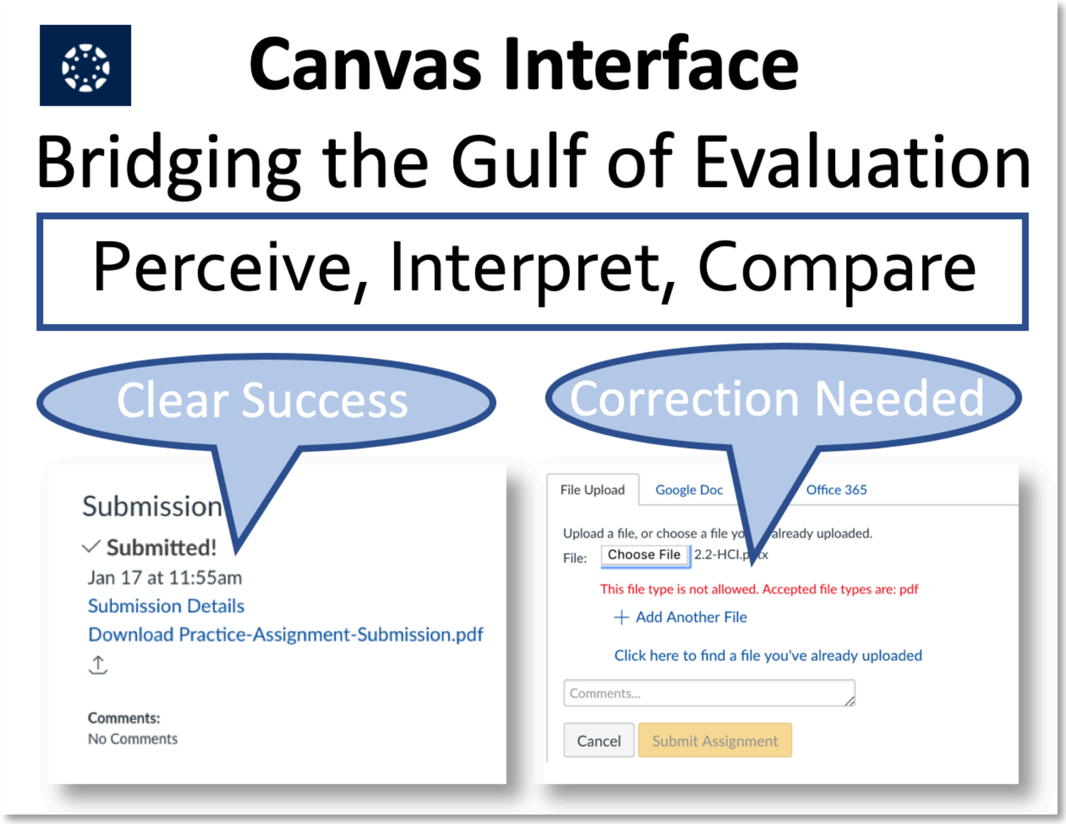
Question 3

Question 3 we describe the process of submitting an assignment to Canvas in terms of feedback cycles. When interacting with Canvas or any system, there are two general challenges. First is the users challenge to accomplish a task via the systems interface, this is known as the Gulf of Execution. And second is the communication of the system to the user in response to a tasks execution, this is known as the Gulf of Evaluation.

Let us first look more closely at the Gulf of Execution for submitting to Canvas. This Gulf deals with the difference between what the user thinks they need to do and what they actually need to do to accomplish their goal.

Canvas provides an excellent intuitive interface to support this process. As a novice user, various signifiers help in identifying the steps to accomplish the task. Canvas uses easily identifiable icons, and commonly used metaphors (such as Dashboard) to facilitate identifying intentions. Additionally, many shortcuts are also offered to facilitate quicker navigation. Canvas also provides ***s***trong indicators to help the user in identifying actions in the system. The actions are what steps are necessary to accomplish a specific goal. Canvas achieves this with clearly named web links, and predictable hierarchical navigation making executing the steps in the process easy to identify and follow.

When we look at the gulf of evaluation, Canvas also does a good job providing clear, immediate and understandable feedback. For example, once an assignment is submitted the system immediately responds with an affirmation. With "Submission Submitted, the data is was submitted and a link to download the submission" And if the submission fails and immediate response is again provided but with a failure message (presented in red) indicating why the submission failed.



Canvas Interface

Bridging the Gulf of Evaluation

Perceive, Interpret, Compare

Clear Success

Correction Needed

Submission

✓ Submitted!

Jan 17 at 11:55am

Submission Details

Download Practice-Assignment-Submission.pdf

Comments:

No Comments

Upload a file, or choose a file yo already uploaded.

File: Choose File 2.2-HCI.p x

This file type is not allowed. Accepted file types are: pdf

Figure 1. Example of Bridging the Gulf of Evaluation in Canvas

Question 4

In this question, we explore two kitchen appliance designs one with a large gulf of execution and gulf of evaluation and one that bridges these gulfs well.

First, let's look at an appliance with a large gulf of execution. At first glance, this appliance would appear to be to simple to have a large gulf of execution as it only has two dials. I am referring to a simple toaster oven that I have owned for my year. One dial has an icon which looks like a temperature gauge and has clear incremental numbering from 0 to 450 degrees. The second dial has two icons. The first is a clock and the second is a slice of toast. On this dial, the first half contains some smaller toast icons with different shades of darkness starting from white and progressing to all black. Immediately following the last toast icon is a number scheme starting initially at 10 scaling and up to 30 minutes.

Unfortunately, this appliance suffers from a larger gulf of execution when attempting to ***execute in the interface*** and a gulf of evaluation when ***interpreting*** the interfaces feedback.

Your initial conceptual model when looking at this simple interface would be that the bottom dial is for toasting and the top dial is for cooking or reheating. And indeed, if you place toast in the toaster and turn the toaster dial to a setting for medium toast. A light turns on, and a timer will start ticking (with an audible ticking sound). In the gulf of evaluation you are receiving feedback that would indicate your bread is toasting, unfortunately, it is not. With the initial conceptual model broken, you may try to turn the temp dial to say 300 or 400 or some other potentially desirable temperature, but this will not work either. To properly execute toasting you must turn the temperature dial all the way up and past the max 450 temperature into toast mode (which is not clearly indicated). However, you need to do this step first doing so after starting the toasting process it still will not work. Additionally, if you want to use the oven mode, you (1) set the temperature, then (2) turn the timer dial past the 10-minute maker (3) set the cook time which can be in the toasting area or up to the max 30 minutes.

Now let us look at a K-cup coffee maker. This device also has a simple looking design with a small touch screen and only one physical button. There is a silver handle which clearly signifies lifting to insert your K-cup. Once you lift the handle, the button and screen are obscured (forcing you to only deal with the K-cup insertion you cannot hit any other buttons) Once you close the lid. A cup icon is immediately displayed in the screen along with arrows indicating smaller or larger cup sizes. And the button begins to flash indicating (push me to start brewing). Once you push the button, it stops blinking and starts to slowly dim and then become bright in a repeated pattern indicating that the coffee is brewing. Additionally, the display provides a coffee brewing animation. This remains on until the coffee is brewed and dispensed. Then the light stops, and the amination changes to an "Enjoy your Coffee" message.

Redesign the weaker interface

When we look at the way that the coffee maker ensures linear and consistent steps with solid signifiers and clear feedback throughout the complete coffee brewing process we can likely find some ideas that would help bridge the gulf of execution and gulf of evaluation in the toaster oven.

To improve *identifying actions* the toaster oven could distinctly separate cooking and toasting tasks with a toggle switch. Forcing a path of execution similar to the coffee maker. With toast selected the set temperature operations would be disabled and with oven chosen the toaster setting would be disabled. This adds a button but improves the ability to *execute in the interface.* Additionally, to support *interpretation,* the oven would provide feedback only when an operation was actually executing. And finally, to promote *improved evaluation,* the feedback could be more explicit (by providing a cooking light and toasting light) and potentially a digital timer display.

Although these designs would likely improve the interface, we must understand that designs do have trade-offs with costs, durability, simplicity (hidden shortcuts) and usability (discoverable) among others.

References

Joyner, D. (2017). Human-Computer Interaction. Retrieved from https://www.udacity.com/course/human-computer-interaction--ud400

Norman, D. A. (2013). *The design of everyday things* (Revised and expanded edition. ed.). New York, New York: Basic Books.