Assignment P1 (Summer 2021)

Veronica Shei
veronicashei@gatech.edu

1 DESIGN PRINCIPLES

This section goes through Human Computer Interaction design principles. We will first go through the processor and predictor models in relation to the Piazza interface and compare the insights from these two models. We will then examine one way to use the Spotify application in different contexts, discuss the effect of contexts on how we use Spotify, and describe how we can improve the design to overcome any restraints.

1.1 Piazza from the perspective of the processor model

A processor takes input in, and spits output out. When designing with the processor model in mind, we are concerned with ensuring that the interface fits within known human limits or more specifically that the interface is physically usable. We can determine whether the interface is usable via quantitative experiments like reaction time.

Piazza is a forum interface. It is a space to ask questions of our professors, teacher assistants, and peers, and to go through previous responses in search of an answer to our questions. Posting a question is simple: the user clicks on the blue 'New Post' button in the upper left and a new question interface appears. The user then writes the title, question, adds any tags, and presses the orange submit button to post their question. This action took around 20 seconds for me but may take longer depending on one's familiarity with the interface (i.e., the amount of time it takes to find the 'New Post' button) and the length of one's question.

1.2 Piazza from the perspective of the predictor model

A predictor can map input to output or predict the outcome of an action they take. Thus, the predictor model aims to design interfaces that fit with human knowledge and typically uses qualitative assessments like cognitive walkthroughs to do so.

In Piazza, we are shown an interface with many different posts and buttons and tags. The amount of information provided in the interface might make it difficult to find the specific buttons we need to use. However, each button is clearly labeled to allow for easy prediction of what pressing on the button would entail. For example, as a user, I would assume that pressing on the “New Post” button would lead to a new post interface appearing on my screen. This assumption proves to be correct as the “New Post” button opens an interface that lets me post a question, note, or poll/in class response. While I might not have been able to predict that I was able to make three different types of posts, I was able to predict that this button click would lead to some sort of new post interface.

1.3 Comparing the insights from the processor and predictor models

From our discussions above, the processor and predictor models focus on different aspects of design. The processor is concerned with quantitative features like efficiency, reaction time, while the predictor is concerned with qualitative assessments like the user thought process.

Improvements we might derive from the processor model is ways to make the post a question flow faster. One way to do this is to make the “New Post” button larger, and to choose a brighter color so the button stands out from the rest of the interface. These two actions would increase our reaction time by making the button more obvious to the user.

Improvements we might derive from the predictor model is to update the “New Post” button to be a drop down. By turning the “New Post” button into a dropdown, the user will be able to clearly see that there are three different post options. Then when the user clicks on the specific option, their experience will exactly match their expectations.

1.4 Using the Spotify application in different contexts

I use the Spotify application in many different contexts. When I’m doing work, I oftentimes have music playing in the background. I may have the music playing from my laptop or from my headphones depending on where I’m working. If I’m in the office I use my headphones, so I don’t disturb my coworkers with my music. I generally play music that is calming so it doesn’t distract me from doing my work. I also try to play from one large playlist, so I don’t look like I’m playing with my phone at work. If I’m doing work at home, I’ll likely just play music directly from the laptop.

When I'm driving, I plug my phone into the auxiliary cable and use the Spotify application via the Google Maps integration. Since I'm driving, I won't be able to fiddle with the music, so I need to select the playlist before I start driving. My car is older and doesn't let me use the car controls to change the songs. This means that I need to change songs via my phone which is distracting. The Google Maps integration doesn't always work so sometimes I'll even need to switch between the two applications.

When friends are over, I'll connect my phone to my Bluetooth speakers and play from our collaborative Spotify listening session. This feature is great because it allows my friends to control the streaming so they can queue up songs they want to listen to, control the volume, or skip songs they might not like. The main issue with the shared streaming session is if too many folks have opinions on the music, and everyone keeps changing the songs.

When I'm walking my dog, I like to listen to Spotify via my Apple AirPods. I usually listen to it in the non-noise cancelling mode so I can still hear outside noise.

1.5 Improving the Spotify application based on context

An improvement to Spotify when I'm doing work might be to sense my location and automatically suggest connecting with my headphones if I'm in the office. Another improvement could be to suggest playlists that I tend to listen to when I'm at this location. Because I try to listen to more 'elevator' type background music when I'm working, it would be a great feature to not have to search for the playlist every time.

An improvement to the Spotify experience while driving is to ensure that the Google Maps integration always works. Sometimes when I click on the integration button, the interface doesn't change and the authorization to use Spotify fails. Instead of needing to authorize the usage of the Spotify application via Google Maps every time, that should be a one-time activity that we can later disable in the settings.

For the shared listening session with friends, a good improvement might be to somehow limit the number of songs changes one can do. Or to designate an administrative user to resolve any song conflicts.

For the dog walking context, a good improvement would be to have a predefined setting for ‘walking’ so my volume, Bluetooth preferences, etc. are saved and I don’t need to toggle through all the different modes independently. These types of predefined settings could also work for other contexts like ‘work’, ‘at home’, etc.

2 FEEDBACK CYCLES

This section goes through Human Computer Interaction feedback cycles. We will examine the Canvas interface via the lens of feedback cycles to better aid our understanding of how feedback cycles work. We will then examine two different types of online reservation systems from the perspective of the gulf of execution and how this might help us come up with improvements to the system.

2.1 Canvas and the three stages of the gulf of execution

The gulf of execution is defined as the difference between what the user thinks they must do and what they need to do to achieve their goals. The three parts to the gulf of execution is to identify intentions, identify actions, and execute in the interface.

To understand the gulf of execution for assignment submissions in Canvas, we will define it across the three stages. Firstly, my intentions are to submit an assignment in Canvas. Secondly, these are the actions necessary to achieve this task:

1. Navigate to the Canvas URL
2. Click on the ‘Login’ link in the right panel
3. Log into Canvas
4. Authenticate with Duo
5. Click on the CS 6750 class
6. Click on ‘Assignments’ on the left
7. Click on the assignment you want to submit from the list
8. Click on ‘Start Assignment’
9. Upload a file
10. Agree to the terms
11. Click on ‘Submit Assignment’
12. Verify that the assignment was successfully submitted

Finally, I need to execute the above actions in the Canvas interface.

In total, there are twelve steps one needs to undertake to submit an assignment in Canvas. The good thing is that the interface is well laid out, with links and buttons in the area you expect them to be. The labeling is also straightforward so the user can navigate through the website logically. Some of the labeling is repetitive as well so the user can find it in multiple places (i.e., 'Assignments' is in both the left panel and in the top icon bar).

At any steps where the action might be confusing, Canvas provides more detailed instructions. For example, during the two-factor authentication action, the user has three different options to authenticate and a small explainer of what each option entails.

One way the user might fail in submitting an assignment is using the wrong format (i.e., a Word document instead of a PDF). While Canvas does specify only PDFs are accepted, making the wording bigger and more obvious could be useful. If you do try to submit the wrong format, Canvas displays an error to remind you to submit file as a PDF. If Canvas automatically converted any text-based format, that could be a great way to remove the mental burden of remembering the correct file format.

Finally, Canvas does provide a shortcut for users who don't want to click through so many links to get to the assignment. On the 'Dashboard' screen, there is an icon for the 'Assignments' link that takes the user directly to the assignment list. Canvas also has a search bar at the top of the 'Assignments' page so users who don't want to scroll through all the assignments can quickly filter out the specific assignment they are looking for. These are both good shortcuts for the more advanced user.

2.2 Canvas and the three stages of the gulf of evaluation

The gulf of evaluation is for the new user to determine the new state of the interface based on the actions they took. There are three parts: interface output, interpretation, and evaluation.

The interface output we received in response to our assignment submission is a loading icon next to the assignment button. This loading icon indicates that the assignment is being submitted. If the submission is successful, the user is redirected to the assignment submission page with an update to the submission's header on the right with the new submission time. If the upload task fails (i.e., wrong document format) the error is communicated on file upload in a red error text. If the submission itself fails, the interface most likely would not redirect to the assignment page. I was unable to verify this because I could not replicate a system submission error.

The interface outputs are all straightforward to interpret because they are text based so they inform you of exactly what happened. Finally, the evaluation is clear because if the submission was successfully the screen is updated to say 'Submitted!' on the right side. However, the successful submission text is not obvious. The interface may benefit from some sort of alert or banner for both success and error on submission to make the outcome explicit to the user. The banner will also reduce the reaction time for interpretation and evaluation.

2.3 Hanauma Bay reservation system and the gulf of execution

On Oahu, there is a very popular snorkeling spot called Hanauma Bay State Park. Due to COVID-19, they implemented an online reservation system to limit the number of people at the park every day. When you go to the website, it states that guests can book tickets 48 hours ahead of their planned visit. It then has a large 'Reserve Now and Pay Later' button to make it clear to users that this is where you can make those reservations.

However, when I click on that button, I don't get results the match the instructions given. When I first attempted to make reservations, I was only given the option to make reservations for the next day. However, all the dates were sold out and this did not match my expectation of having time slots be available 48 hours before.

I then reasoned that maybe spots opened at midnight. But when I tried to make reservations at midnight, there was no date option to select from the reservation dropdown and the website just said, 'No sessions available for this day!' even though I had not selected a day yet. Furthermore, I was confused as to what 48 hours ahead of my planned visit meant. Did that mean at midnight or specifically 48 hours ahead of the time slot I wanted. If it meant by time slot, why didn't it tell me what time slots I could choose from? Eventually I gave up on making a reservation because the gulf of execution was too large. Even though I identified that my intentions were to make a Hanauma Bay reservation, I could not identify the necessary actions to accomplish that task.

2.4 Pearl Harbor reservation system and the gulf of execution

Concurrently with the Hanauma Bay reservation system, I also wanted to book some tickets to the Pearl Harbor National Memorial.

The website made the reservation link easy to find from the homepage. It then asked for the date I wanted (with a calendar popup and different colors and a legend to indicate 'Available', 'Not Available', or 'Not Yet Released' tickets). It also asked for the quantity of tickets I was looking for and greyed out timeslots if the quantity I wanted was more than what they had available. The calendar popup was very useful in helping me determine which dates still had available tickets. Also, if I clicked on the 'Not Yet Released' dates, it told me the exact date and time those tickets would be released. That way I could plan when I needed to be online to book my tickets. The ticket booking process was a simple checkout process that consisted of logging in, selecting my date and quantity of tickets, adding the tickets to my cart, providing my credit card information, and then checking out. The steps aligned with my expectations for how to book a ticket and it was easy for me to execute.

2.5 Lessons from the Pearl Harbor reservation system

The key feature the Pearl Harbor reservation system implemented that the Hanauma Bay reservation system did not was informing us which dates were available and at what time tickets would be released. The Hanauma system was unclear in when tickets were released, and the interface did not match the instructions (i.e., not even showing any dates from the dropdown list). Even for dates that Pearl Harbor had no tickets available, it clearly wrote out that information in the calendar, so the user knew the date was sold out. Because I couldn't even select the day in the Hanauma page I didn't know if the date was sold out or if the website was faulty. Thus, the Hanauma reservation could benefit from clearer instructions (i.e., tickets released at 3 PM 48 hours before) and an interface that correctly matches the instructions given.