Gurubasavaraju M N

gmn@gatech.edu

Question 1

PROCESSOR MODEL DISCUSSION:

In terms of processor model, the interface must adhere to human limitations. As soon as the user logs in, the user should be able to see the screen like below fig1 in the Canvas interface. By default, a user is shown the dashboard, which is also highlighted in the left pane, from which he can click on the courses he has opted to learn

On the left pane, the user has several options to choose from, all of which are self-explanatory. For example, the user can click on courses to see a list of courses, calendar to check for events over time, inbox to read messages, history to see what the user did previously, my media to see the media files the user loaded and help to get help with the tool. By looking at this interface, the user can understand what to do and easily click on what needs to be done.

On the right-hand side, we have the To Do list, which displays a list of pending tasks for the user to complete. The interface provides all of the options to choose from and provides the user with the necessary information to get to the desired location. The interface is very simple to understand and use, whether it is course content, tasks, calendar items, inbox, or media.

The interface is useful because it contains all the course content that the student needs in one location. Lessons, videos, reading materials, announcements, assignments, projects, and other course materials. The interface, on the other hand, is very usable due to its simple and human-readable format. For the first time, a student can directly pitch in and understand without any tool-specific knowledge to perform his task.

The efficiency of the interface can be measured by how quickly a user can launch his course, how quickly he can get directly to doing his tasks, and so on. In this interface, the behavior/output of each user click is more obvious.

PREDICTOR MODEL DISCUSSION:

According to the user predictor model, users have certain goals in mind when they interact with an interface. The student anticipates receiving all course materials, including assignments, projects, discussions, lessons, grades, tests, and other activities, through Canvas.

The dashboard items are exactly how the student expects them to be. Since it gives the courses on the dashboard it is easy for the user to predict where each of these would take them. Further inside the course, Announcements, assignments, Ed Discussion, Ed lessons, Peer Feedback, Grades, Files, Syllabus, quizzes tabs are self-explanatory which will exactly provide the details of what student can predict by looking at these options. As shown in the fig2 by looking at the

scroll bar user can expect there is more details below and scroll down to see it. Bingo! User will see what he is expecting to see. User can interpret the interface unless it is Jim Watson and Honorlock which user click with curiosity.

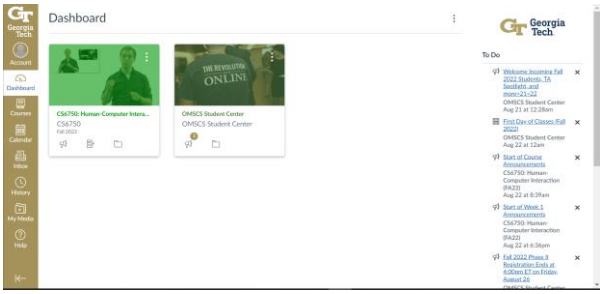

Fig1: Default interface when user login

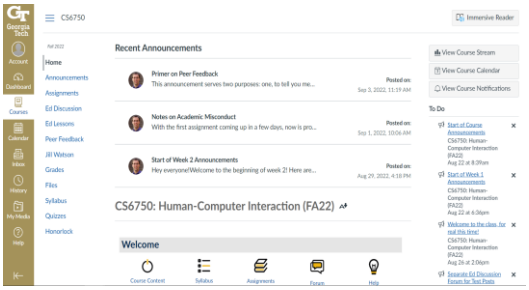

Fig2: UI when course is selected

INSIGHTS AND COMPARISON:

The user can grasp the navigation to the desired course content and accomplish their goal with the aid of the processor model. However, as illustrated in fig3, there is a potential method of optimization to name the icons with descriptions. This will make it easier for users to access Files, Assignments, and Announcements directly from the main menu. As a result, fewer clicks will be needed to go where you're going.

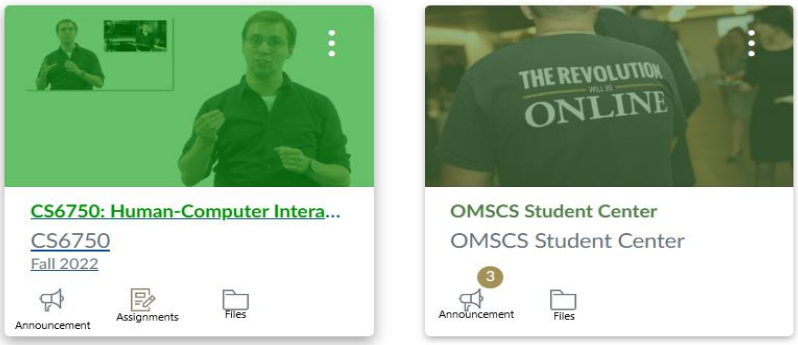

Fig 3: suggested change to optimize the user interface

The predictor model aids the user in making assumptions about what might supposedly occur when they click a specific item in the user interface. Here, there is need for general design improvement for improved user experience. By including a search box on the home page, users may readily predict that by conducting a search and clicking on the results, they will arrive at their goal.

Question 2

I'd want to think about utilizing a fitness app that records users heart rate, sleep, walk, run, and track steps for participation perspective of the user. The app is widely used in a variety of circumstances, including at home, while walking or running, sleeping, and driving.

Let's say a user has an application on his wristwatch that constantly tracks his steps while he is driving, walking, running, sleeping, or doing any other activity that requires movement. The user's movement from one point to another and the hand's motions will be taken into account by the app as input in the interface. In certain circumstances, the application interface acts and responds differently.

Fig 4: Example of fitness tracker interface with the task details

Here are a few examples:

1. At home: The software is made to comprehend the situation when the user is sleeping. All notifications and alerts are set to silence, screen mode is changed to low light, and sleep tracking will be activated based on the user's movement and posture during the user's sleep cycles. The interface's intent is aligned with the context.

   While at home, a user must move around to watch TV, have lunch, do the laundry, make the dishes, make phone calls, etc. In this situation, users' cognitive resources are more heavily focused on carrying out the daily tasks listed above rather than on the actual task that app is tracking. However, the app must be designed to track the steps taken while performing all of these jobs rather than adding to the overall workout plan because it lacks the intelligence to comprehend this situation.

2. Running/walking: In this scenario, the app tracks the user's steps and uses that information to calculate the distance travelled, calculate the number of calories burned, and show various health benefits of the exercise. The fitness tracker app was created with this as its primary goal, and it recognizes context based on user gestures and displays the same information on its UI.

   If a user is sitting still and moves his hand, the tracker may believe that the user is moving, and the interface may display false positives.

3. Driving: While driving, the user is actually moving, but because the software is able to understand context and doesn't count the motion as steps, the interface remains same.

   Even when the user is not moving his body, we can still see that the number of steps may increase. This is a result of the user moving their hand around inside the vehicle for various reasons. The tracker continues to record and display the number of steps taken.

   The user may spend a lot of time at home working on a computer or performing other duties in one spot. It is possible to restructure the user interface so that it recognizes the context and prompts the user to walk around, extend his arms, or engage in physical activity. The software may even interpret the user's hand position when working at a table and indicate whether the position is ergonomically sound or whether some height adjustments are required.

   To prevent false positives about the user's physical activity, the interface can be revised to display more information from the motion sensors on the screen or to ask the user to indicate whether he or she is walking or running.

   The fitness tracker app on the wristwatch is Bluetooth-connected to the phone; when the user gets in a car, it may be possible for the interface to detect that the user is in a car or to request the user's input if he or she is on a vehicle and activate the driving features, such as how far the user is from home or work, how long the user is driving more than usual days, when the user needs to take a break, etc.

Question 3

Assume a student wished to ask a question in class. The student must first comprehend his own objectives, the question he wants to pose to the professor, and the appropriate channels to use. For instance, "Do I receive all the credits after doing the assignment?"

To post questions and receive responses from classmates, teaching assistants, or instructors, use the Ed discussion tool. Therefore, the user must now comprehend that the system's goal is to allow them to post a question on the Ed discussion tool. The Ed discussion tool (Fig. 5) is what the user will access after logging in. The user might enter the categories on which he has questions to see if he has the opportunity to write a question but entering the categories first will help him realize that he can find the prior postings there. Even though it doesn't lead the user to their intended goal, achieving the goal isn't difficult and he can still figure out how to get there.

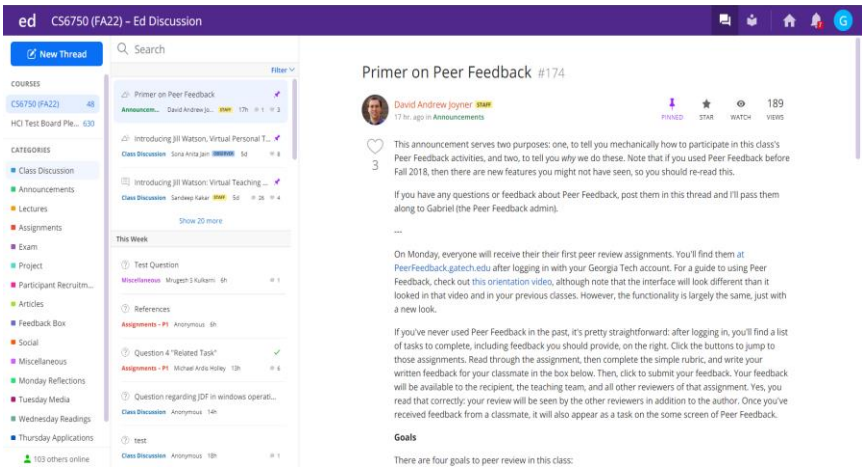

Fig 5: default Ed Disscussion UI

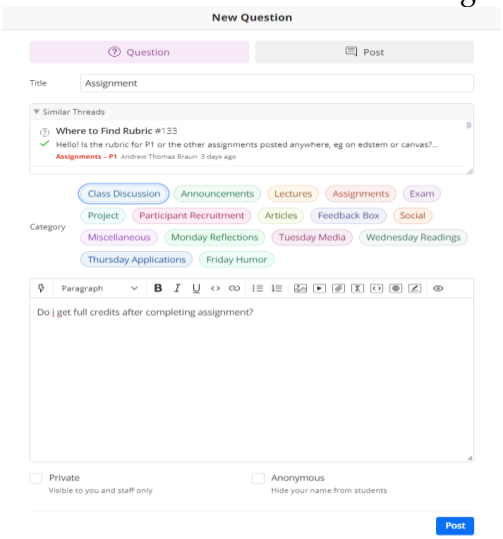

Fig6: UI for user to write message

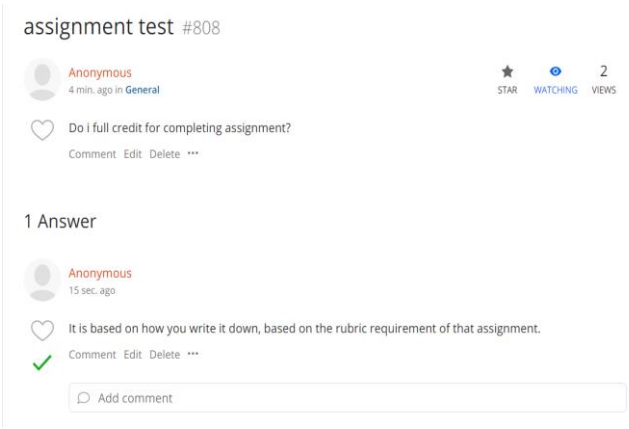

Fig7: Example question and answer in the forum

The user now knows that he can post a question by selecting the "new thread" symbol. The UI makes it easy to find and use the posting function. Additionally, the "new thread" button is aligned with the "compose" icon in the Gmail tool UI, which serves as the default spot for all Gmail users to type down messages. The user can relate to it and simply get the desired outcome. This is the procedure for comprehending the action to achieve the objective, which is to ask a question and receive a response.

Users need to click on the new thread to execute the goal of posting his question and getting the response.

The user will now be aware of where to type the question, the category to choose, and the paragraph window where the actual question can be typed after completing the operation. Finally, the operation can be carried out by posting the query by pressing the post button.

Instant feedback was given, and the list in the middle pane of the screen indicated that the question had indeed posted.

The user's inquiry was posted with the expectation of obtaining an answer, which is the basis of evaluation. The user may not be aware that he has actually received the response he is expecting even though it is outputted in the interface when someone responds to the user's question and is displayed in the interface.

When a user receives a response from a professor or peer and receives a notification message on their phone, the interface immediately sends them feedback via email. Now, the consumer is able to comprehend that he has stuff to check. The user must interpret the response from the interface, which makes it clear that someone has answered the questions. You might also get emails with replies to previously asked questions, feedback on your posts, or announcements; these alerts aren't always the answers a user is seeking for.

Once the response has been understood by reading the mail content and the user has access to the Ed Discussion tool, the user must assess whether the response is aiding in the achievement of the user's objective. In this instance, the response explained that whether or not the user followed the assignment's rubric would determine whether or not they received credit for the task. which the user may assess and comprehend that in order to receive full credit, the user must adhere to the required content as specified by the rubric. The user's intention has been accomplished in this case based on the evaluation of the response.

Question 4

I'll use the example of turning on the air conditioner at home as the first activity to demonstrate the Gulf of Execution.

The interface turns on the display on the air conditioning panel with the temperature set in °C when I switch on the mains and push the ON button on the remote control, for instance, I take a value of 23 °C set on the system and the present room temperature is 35 °C. The temperature in the room is supposed to get adjusted to the desired level as soon as the system is turned on. Due to the lengthy process, the user is still unable to verify if the air conditioning system is functioning as intended or not. The user won't be able to sense the room temperature decreasing for a few minutes. After performing an action there is a gap in execution that can be felt.

Fig 8: Air condition UI example

For instance, the cooling system may not be functioning properly because it is clogged with dust, the exterior unit may have problems, or the condenser may be malfunctioning as a result of the weather. However, the user is not made aware of any problems with the system or why the temperature cannot be reduced to the desired level. The interface just indicates that the system is ON and that the temperature is set to 23 °C, even if the room's actual temperature is still between 30 and 32 °C. User evaluation of this is only possible after some time has passed; user will not receive quick indication that the system is doing as it should.

Now let's think about watching Amazon Prime shows on any device. A membership and internet connection are the prerequisites. Suppose we have both.

Users can anticipate seeing a display with the movie's content when they attempt to launch a movie on Amazon Prime. If the user encounters a slow internet connection and the video does not load, the interface is intelligent enough to display a buffering action so that the user can see that the material is still loading and has not crashed. Alternatively, if internet access is lost, a prompt "No connection" will appear on the screen, letting the user know that there is no internet. The user's execution gap is reduced as a result.

Fig 9: Interface showing the Video is still buffering

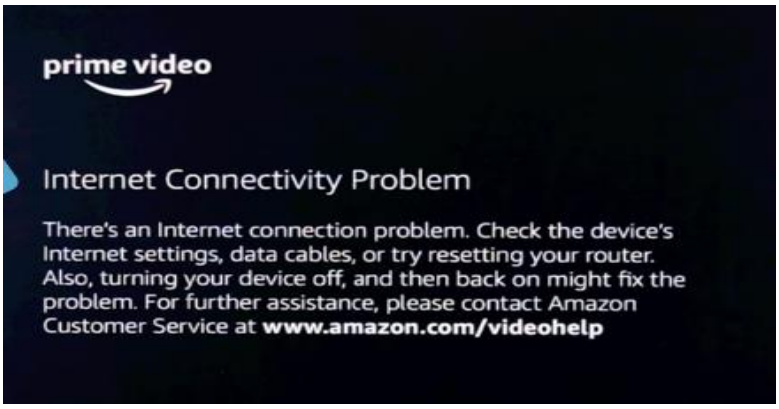

Fig 10: Internet connection error on Amazon prime

The Amazon Prime interface teaches us that a better user experience will result from a better user interface for any task. It is clear that both Amazon Prime interface and the air conditioning system are helpful. However, the Amazon Prime interface is easier to use than the previously described air conditioning system due to the interface's rapid feedback.

By redesigning the user interface, the system will give the user rapid notice that there is a problem with the system, which the user may fix. If there are no problems, the user can be confident that the system will function as intended and can even watch the temperature change to confirm that what they did would actually happen after they took their action.