

Assignment-P1

Arti Chauhan
achauhan39@gatech.edu

1 QUESTION-1

In this section I will be discussing *Piazza* web interface from the perspective of user as a *Processor* and as a *Predictor*.

1.1 Processor Model

When we evaluate an interface from Processor perspective, we care about *what* information is communicated to user, *when* and *how* ^[1]. Piazza web interface is

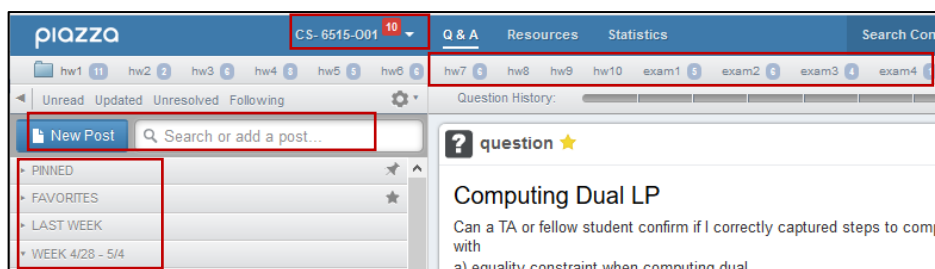


Figure 1a. Piazza web interface for an enrolled class

quite intuitive and well thought of. Information is organized effectively for students to accomplish various tasks. Interface presents necessary information in succinct and clear format as shown in Fig-1a. E.g. Top drop-down menu lets user select desired class or add another class. If user opts to enroll in another class (assuming user is logged in), App remembers what school user is in and which classes user has already joined for current term. As user starts typing in class name or number, interface presents list of relevant classes offered by user's school. Thus, interface provides all necessary information at each step to optimize user's enrollment process. I evaluated this interface with some quantitative experiments to measure task-completion efficiency in terms of time /clicks. Table below shows the summary, along with insight gained from each experiment.

Assumption: user is already logged into his desired class.

1. Creating a new post

Actions	a) Click on 'New Post' button in left pane b) Choose appropriate folder. Change default 'Post-To' & 'Post-type' (question, note or poll), if need be. c) Fill in 'Summary' and 'Details' of question/note and hit 'Post' button
Efficiency	Number of clicks: 5-7 Time taken: 4-6 seconds (excluding step-c)
Key insight	Time for post creation can be minimized by providing default setting for as many selection parameters as possible. Piazza already has default setting for 'Post-Type and 'Post-To'.

2. Filtering by Keyword

Actions	a) Navigate to search-bar in left pane. Enter the keyword. [<i>Post(s) containing keyword appears in left pane. Results are sorted by 'Relevance'.</i>] b) To sort results by 'Date', click on 'Search Tips' underneath search-bar, select 'Date' and click on 'Search Tips' to collapse the tips window. c) To see where keyword appears in selected post, click 'Ctrl-F' and type in the keyword again.
Efficiency	Number of clicks: 1-5 Time taken: 11 seconds
Key insight	<ul style="list-style-type: none"> Making <i>sort</i> option available on main ribbon (eg: underneath the search bar) will speed up step-b and will make option discoverable. Step-c can be eliminated if interface keeps the keyword highlighted in the selected post as well.

3. Filtering by Topic

Actions	a) Navigate to 2 nd ribbon from the top, click on desired folder. [<i>Posts for selected folders are displayed in left pane.</i>]
Efficiency	Number of clicks: 1 Time taken: 3-7 seconds
Key insight	Time for folder selection can be reduced if folders are alphabetically sorted.

1.2 Predictor Model

From user as a Predictor perspective, Piazza's interface is mostly in line with user's *experience, expectation* and *knowledge* ^[1].

- When 'New Post' button is clicked, a form is presented to user to fill in content and specify recipients, which matches user's expectation.

- User can efficiently draw upon his prior experience with other email interfaces (like Gmail or Outlook) to organize/view posts by topics (folders) in Piazza.
- Interface allows user to quickly learn just by glancing at left pane if a post is a question, poll or note, which posts have instructor's response, how many updates have occurred to a given post since user last viewed it.

Thus interface helps user learn what can be accomplished and efficiently build upon his prior experiences. [1].

1.3 Comparison

To evaluate this interface from Predictor perspective, I tried to simulate cognitive walk-throughs of some tasks, which gave me insights into cases where interface's output didn't fit with user's expectation. This helps in highlighting advantage of taking Predictor perspective over Processor in designing better interfaces. While Processor view enabled *evaluation of task-completion speed*, Predictor view gave me better *understanding of user's needs and expectation*. I have presented two such examples below.

Case-1: red-box in Fig-1b shows what user entered in latex editor while formulating a post and Fig-1c shows how content was displayed to him in the post form, which was not in line with his expectation. User had to click 'preview' button (which shows correct formatting) to proof-read his post. This can lead to frustration, if use of latex is frequently required, such as in Math-heavy classes.

Case-2: Interface provides a button for image insertion, which gives user impression that he can perform this action by more efficient 'drag and drop' method. This expectation is consistent with his experience with other interfaces. However, 'drag and drop' is not supported by Piazza interface and fails to build upon user's prior experience with other interfaces for similar task.

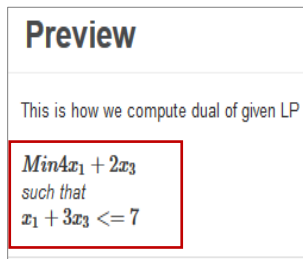


Figure 1b

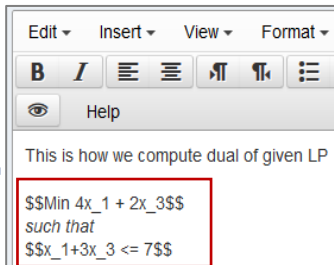


Figure 1c: Latex usage



Figure 1d: Image Insertion

2 QUESTION-2

2.1 Multi-Context Activity

For this question I have chosen my home-security app, which I use on a daily basis to control my home intrusion-detection system, smart lock at front door, video door bell and surrounding cameras. I use it in many different environments – from my bedroom as I am getting ready to hit the sack, from my car as I am heading out to work or gym, from my work as my Tweens get back from school.

2.2 Current Limitations/Constraints

App is adequately designed for users who are stationary. I can access each component of my home-security network. This requires clicking on different tabs (corresponding to each component), which is not a big deal when user is stationary but becomes cumbersome and distracting if user is driving. It is almost impossible to interact with the app without halting his current activity, if user is biking or running.

2.3 Interface Enhancement

Following enhancements can be made to the App to accommodate user's varying contexts, in particular, when he is **mobile** (driving, biking, running) or **busy in a meeting**.

- 1) If there's a visitor at my door, App determines my context and uses appropriate modality to notify me.
 - If I am driving (which App can determine by sensing my rapidly changing GPS location), App plays a *voice* notification that 'someone is my door'.
 - If I am in a meeting (App can learn this from my calendar), it sends a *text* notification. It mutes the notification-volume and silently flashes notification on my screen to get my attention.
- 2) We can take it a level up, where App instead of telling me that *someone* is at my door, it tells me *who* is at my door, so that I can make a decision whether to ignore the call, without any physical interaction with the phone/App.
- 3) Build a voice recognition system where user is authenticated by his voice, thus eliminating the need to enter password or finger-print every time he

needs to use the App. E.g. User can command the App to arm the system as he is driving.

- 4) Even better, if App learns from user's usage pattern, just like NEST. Currently user has to explicitly configure in the App day and time when security system should be armed. App should learn that it's time to arm the system as I head out for work every morning.

3 QUESTION-3

3.1 Gulf of Execution

Identify Intention: User has a goal of submitting a response to an assignment. So, he forms his intent for submission and tool provided to him for this task is Canvas web interface.

Identify actions: Assuming user is already logged in, Canvas interface presents user with courses he is enrolled in. User selects the desired course and spots 'Assignment' tab on left pane without much difficulty. Submission page displays that file should be in pdf format. It provides several file selection options (from user's local machine, DropBox, Office-365 account etc). Each tab on Submission page is intuitive and enables user to identify further actions to be taken to complete submission (Fig 3a)

Execution: Now that user has identified actions to accomplish his goal, he needs to execute those actions. Fig-3a below shows sequence of steps/clicks user performs to submit an assignment that resides on his local machine.

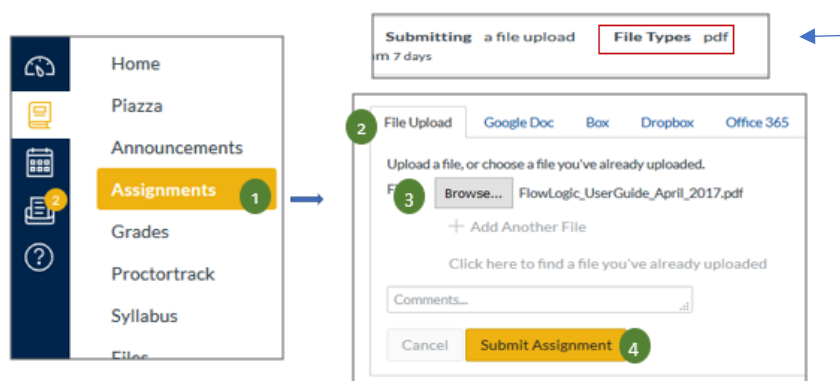


Figure 3a: assignment submission steps

3.2 Gulf of Evaluation

Interface Output: When user clicks 'Browse' button, file-manger is opened for user to select appropriate file(s). Successful selection of file is shown against 'Browse' button (Fig 3b) .When 'Submit Assignment' button is hit, user is presented with submission status as shown in Fig-3c, if successful.

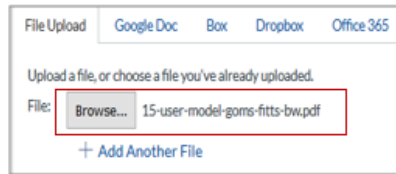


Figure 3b: successful file selection

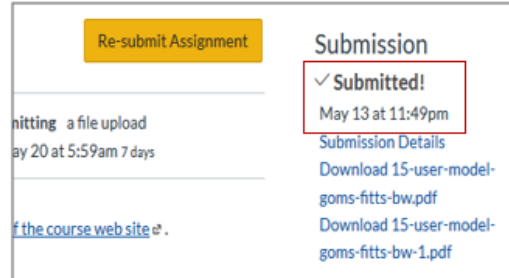


Figure 3c: successful submission

Interface has failsafes built-in, where appropriate error message is displayed

1. if user accidentally hits submit button without uploading a file. (Fig 3d)
2. If user selects a file which is not in pdf format. (Fig-3e)

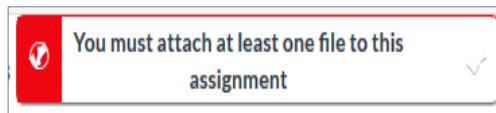


Figure 3d: error-1

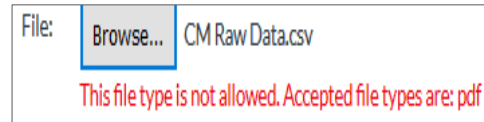


Figure 3e: error-2

Interpretation: At every stage of submission process, Canvas provides feedback which helps user interpret outcome of his action. When user selects a file to submit, interface either accepts the file if it is in correct format or spits out an error message, clearly informing user what went wrong (Fig-3d & 3e). If submission is successful, submission-details are provided to user (Fig-3c) which includes files submitted and time of submission.

Feedback on successful submission is subtle. For expert user, it is might be easy to infer little tick mark against 'Submitted!' (Fig-3c) as sign of success but a novice user might benefit from an explicit message stating 'Submission was successful' and optionally, sending email/text confirming successful submission.

Evaluation: Clear and concise feedback helps user correctly interpret state of the system and thus can evaluate if he was able to accomplish his goals or not. This

helps narrow gulf of execution as user is constantly kept abreast of his actions and can quickly correct his course, in case of an error.

4 QUESTION-4

4.1 First Activity

We recently moved to a new house, which meant learning to use all new appliances. As someone who loves to cook, I was very excited about my new professional-grade gas-range. But as I started using it, I realized that it wasn't as intuitive as I thought. It suffers from a large gulf of execution. Few examples below -

- 1) As can be seen in Fig-4a below, designer of this gas-range chose to stay allegiant to legacy gas-range design, filling up entire 48" panel with 12 bulky knobs. This has made interface too busy and leaves user confused. While I can easily figure out which knobs corresponds to leftmost and rightmost burners (by their very placement), I still can't remember which is for front and which one is for back burner, until I try both. As we move to the middle, it becomes even more confusing, with never ending array of knobs.
- 2) There is no option for timed-baking. Burden is on user to keep track of time while baking, otherwise you end up with a burnt dish.
- 3) Most recipes typically say '*bake for x minutes*'. Here we have an interface which gives user two options - '*Bake*' or '*Convection Bake*', without providing any cues on what the difference is between the two. (Fig-4b)



Figure 4a. gas-range



Figure 4b. oven control

4.2 Second Activity

In contrast, I found my new washing machine interface very well designed and had minimal learning curve. All options are laid out in a grid, with a clear

grouping for temperature, spin and soil. My-Cycle option (top right corner), aptly named, conveys user that he can save his preferred cycle settings. Interface also teaches user how to save it i.e. *by holding it for 3 sec* (displayed on the interface itself). I thought this was very clever. Thus My-Cycle enables user to run a load with his preferred setting with one touch, narrowing the gulf of execution. It also selects appropriate load-size by sensing weight of cloths put in and gives user option to start a cycle at specified time, which further narrows the gulf. It has a child lock which my gas-range lacks, a feature many parents can appreciate.

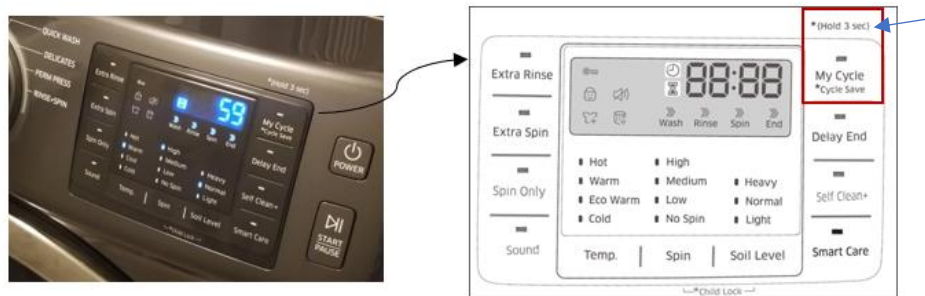


Figure 4a. washing machine's user interface

4.3 Lessons Learnt

There are several inspirations that can be drawn from washing machine interface, to shrink the gulf of execution for gas-range.

- Given that digital-clock technology is so affordable these days, it can be easily incorporated into above-mentioned \$6K gas-range. This will shift the burden of keeping track of time while baking or cooking, from human to machine.
- To control 6 burners, 2 ovens and 2 grills, use a digital display, which will not only get rid of clumsy array of knobs, but it will also make interface clean and intuitive. Eg: We can project burner locations on the display so that user can deduce the mapping easily. Digital displays also provide a medium to convey feedback (eg: error messages) to user, which physical knobs can't. Thus, it helps narrow gulf of evaluation as well.

5 REFERENCES

1. Joyner,D.(2019). Introduction to design principles
<https://classroom.udacity.com/courses/ud400/lessons/9044917867/concepts/91374998120923>