Assignment-P2

Arti Chauhan
achauhan39@gatech.edu

1 QUESTION-1

1.1 Current course enrollment process

Below are the steps I follow as an OMSCS student when I enroll in a course. I find out CRN of courses I'm interested in beforehand, as it reduces number of steps required for registration process when time-ticket window opens.

Finding CRN

1) Go to Oscar->Buzzport and complete two factor authentication.
2) Click on '*Registration-OSCAR'* icon -> '*Student Service & Financial Aid'* -> *Registration* -> *lookup classes* -> Select appropriate *term* from dropdown menu.
3) click on '*Advanced Search'* -> select '*Computer Science & Engr'* and '*Computer Science'* -> Select Campus = '*Online'* -> click on '*Section search'*
4) List of relevant courses along with CRN is displayed. Note down CRN of desired courses.

Registration

1) When time-ticket window opens, login into Buzzport.
2) Click on '*Registration-OSCAR'* icon -> '*Student Service & Financial Aid'* -> *Registration* -> select *term* -> Enter *CRN* -> *submit.*

1.2 Redesign

Following redesign strives to reduce cognitive load by allowing user to directly manipulate the object of interest (course). Visualization of redesign is shown in Appendix.

Given that user has authenticated himself in step-1, interface knows who the user is and hence can create a personalized registration view for the user.

1. Interface determines user's specialization, how many mandatory courses he has taken so far, what other electives are available to him and, if user is OMSCS student or not.(It can query *DegreeWorks database*, which already maintains this info). Based on this information, interface will prepare a curated list of courses that are relevant to student.

2. This list will have 2 sections . Top section will display courses mandatory for his specialization. This will include core courses and electives. Bottom section will show list of available free-electives. Each course will show available seats, waitlist status and instructor info.
3. Courses that user has already taken (and had passing grades) will be grayed out. This will not only give user a quick summary of what courses he has taken so far, it also eliminate user erroneously selecting that course.
4. Interface will also allow user to mark certain courses that he plan to register for as 'favorite', which can be used as filtering criteria when displaying results.

Above changes enables **continuous representation of object of interest** (courses) before and during the process of registration.

5. From this personalized list, user will select desired course(s) via a check box and hit *Register*. Thus, **task object (course) is manipulated by physical actions**, such as clicking, **rather than by entering complex syntax** (entering CRN number).
6. If class is full, user will be notified. User may choose to add himself to wait list. If registration is successful , user is informed about task success and course in question will now appear under '*Registered'* section. Thus, **effects on task objects are immediately visible** to user.

1.3 Benefits of redesign

- **Minimizes clutter and shrinks gulfs of execution:** This redesign presents curated course list and avoids displaying extraneous information. This reduces user's cognitive load and shrinks gulfs of execution. Information presented in this manner helps user quickly map his goals into action , which reduces semantic distance. Once user determines which course he wants , interface make it easy to accomplish his goal (i.e. complete registration). This helps in reducing articulatory distance.
- **Efficient :** This redesign allows user to complete registration in two clicks, there by speeding up the process. It eliminates the need of user having to remember CRN number, which makes it less error prone.
- **Better understanding of user needs :** By displaying courses that are relevant to user's specialization and his status (OMSCS vs OnCampus) interface

proactively determines user's needs. By allowing user to mark certain courses as *favorite* (before registration opens), interface tries to understand user's intention.

2 QUESTION-2

2.1 My past experience

My job as an optimization engineer requires extensive analysis of data. For this purpose, I use Excel , Python or R depending, depending on task at hand. For quick visualization, I started exploring Tableau. Despite having a background in data analysis, Tableau interface was not intuitive to me. I would spend great deal of time trying to figure out simple tasks such as super-imposing a moving average on a trend-line. In Excel, I would just right click on line-chart and select *Format Trendline*, which would bring up different trend line options. To accomplish same, I searched all menus and half an hour later figured out that it is accomplished via complex syntax shown in Fig 2b below.

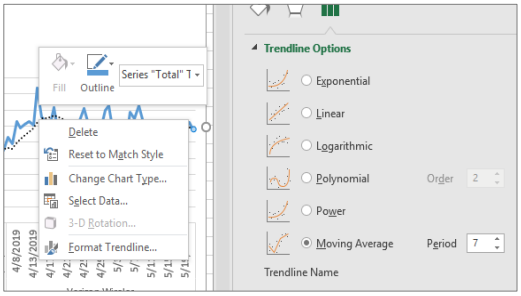

Fig2a : plotting moving-avg in Excel

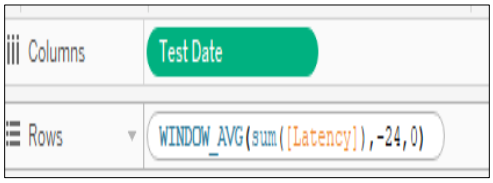

Fig2b : plotting moving-avg in Tableau

Another example I vividly remember is when I tried to create a jitter plot or 3D in Tableau. In Power-BI , I would click R button on right pane, write few lines of code in R-script window and run it (Fig 2c). To accomplish same task in Tableau, user first need to configure a R-server (Fig 2d) as R-server is not configured by default, unlike PowerBI . This in itself was a chore.

Tableau's features are just as powerful, but some of them are less intuitive, hidden behind menus. Hence in the beginning I spent more time on learning about the interface than on actual task at hand.

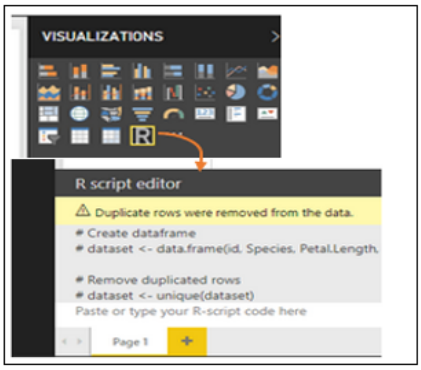

Fig2c : PowerBI – R is accessed by. simple click of a button

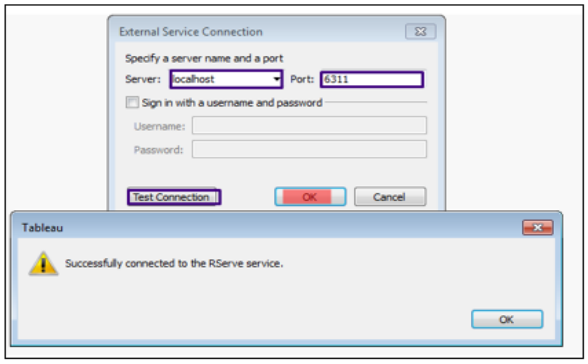

Fig2d : Tableau – Setup R server first.

2.2 Current thought process

Having worked with Tableau for quite some time now , Tableau interface has started to become invisible due to practice. Overtime I have picked up tips and trick for task that I frequently perform. So, I don't struggle as much as I used to, to identify actions for accomplishing my task. Now I subconsciously know which menu to click , which is a result of habituation and user developing more competence due to repeated exposure.
To an experienced user, using Tableau might be straightforward. But layers of menus, heavily reliance on '*Calculated Fields*' even for simple tasks, absence of prompts poses a steep learning curve for beginners.

2.3 Redesign

Following changes can drive user to point of invisibility more quickly.

Let interface teach : Interface could present pop-up messages or prompt in context of user's task. For example, if I want to display 'Date' field on x axis of a line-chart, interface presents several options to cast 'Date' – as an *attribute* or as a *measure* or *discrete* or *continuous*. There is no clear way for me to know what each option does until I try it. A pop-up message explaining the option as user selects it, would help a novice user quickly understand impact of each of those options.

Provide multiple ways to perform a task : Tableau heavily relies on user abilities to write formulas to create new *measures* , which caters to experienced user but not to novice. For example, there is no easy way to impose a polynomial trendline over a scatterplot (which is a trivial task in Excel) , unless user creates a formula for it.

3 QUESTION-3

To answer this question, I have chosen advanced treadmill.

3.1 Different human perceptions

Visual : Most of the modern treadmill have a digital display which allows user to set his goals in terms of time, calories or miles covered. As user is exercising, treadmill continuously provides visual feedback regarding time spent , calorie burnt , heart rate variation over time etc., using the display.

Auditory : In addition to visual feedback , advanced treadmills also provide auditory feedback as machine slows down when user is close to finishing his exercise routine. This is primarily for safety reasons. It alerts user to be prepared for swift change in the tread-belt speed and be attentive to his body balance.
Some machines also give a happy chime when workout routine end, which gives user a sense of accomplishment. For me, it serves as a reminder to wipe off handrail & display for next user.

Haptic : As user begins his workout, treadmill starts off slowly and then builds up speed as per user's setting. Though these changes are reflected on display, user receives an inherent haptic feedback, where his legs can feel that belt beneath him is moving faster, and hence he needs to adjust his speed accordingly.

3.2 Redesign

Visual : Current treadmills are advanced but not dummy proof. I would like to develop a capability in treadmill's interface where it can determine its inappropriate use (Fig-3a) and inform user how much of a safety hazard they are being to themselves. Conveying the consequences via pictures or short video on display can be very impactful and deter misuse.
Another enhancement that I would like is to make treadmill understand my long-term goals. Say, I let machine know that I am preparing for half-marathon. So instead of me choosing resistance, speed, duration for a given day's work out, machine should show me what workout settings I should be using. In other words, it tracks my exercise routine over time and displays appropriate workout routine to build my endurance for long runs.

Fig :3a

Auditory : Most of us, when exercising , tend to keep our selves engaged in other activities , which most commonly involves watching oversized TVs in the gym . Another scenario is – using desk treadmill. This can pose some safety risks, if we were to solely rely on visual feedback because user is not paying attention to treadmill's display. Treadmill interface can be enhanced to determine if user is not looking at machine's display (may be via eye tracking) and provide auditory feedback, especially for the cases that pose safety risks such as change in steepness or speed of treadmill or if user's pulse is not within expected range.

Haptic : Continuing with same context of treadmill usage as explained in section above, treadmill can be enhanced to provide haptic feedback by adding small groves to edge of the belt. This idea is inspired by road safety , where groves on road-edges gives driver haptic feedback that he is about to fall off the road.
If user is engaged in some other task while exercising and his feet come too close to belt boundary and steps on grooves, he can feel that surface underneath him is not smooth any more. This gives him haptic feedback to reposition himself.

Different kind of human perception

- **Pressure** : As user adjusts the incline/decline of tread-belt , he can feel the pressure in his calf muscles, which allows him to evaluate the impact of his action of changing steepness or resistance setting.

- **Thermoception** (ability to sense heat/cold) : Many modern treadmills have built-in fan. When user adjusts the fan speed, he can feel if his action resulted in bringing down/up his body temperature.

4 QUESTION-4

4.1 Tip violation# 1 – Offload task

Few years back I bought this quesadilla maker. It's a sleek device, easy to store but has no controls whatsoever ('*Minimize clutter'* mantra taken to extreme). It has two lights – one for power and other for pre-heat. But it provides no indication when quesadilla is ready. User is expected to keep track of time. And it's not as simple as keeping track of a fixed time duration, user needs to mentally calculate for how long quesadilla must sit in that machine to achieve desired crispiness.

This task can easily be offloaded to machine, by incorporating a timer, which tracks cooking time . Machine would switch off or provide some kind of feedback , visual or auditory, to inform user when quesadilla is ready.

Additionally, interface could be enhanced to provide a knob to set desired crispiness. Based on this setting , machine can readjust cooking time and inform user accordingly.

4.2 Tip violation# 2 – Minimize clutter

Fig-4b below shows interface of my recently retired GE washing machine. While my quesadilla machine had too little user options, my washing machine was at other end of the spectrum, with too many user options. This busy interface is not only confusing to use, but it also fails to highlight key buttons that user need to press to perform simple task of laundry. While I could figure out functionality of two side knobs without much difficulty, I was never able to fully understand middle dial. This interface had several user-options that left me confused and even reading manual didn't help.

Below are few examples of some redundant options on this interface that I would eliminate as part of redesign.

- Middle knob requires user to select if load is *light* , *normal* , *heavy* or *bulky*. Then it requires user to select water-level - *low, medium, high, max*. The water-level knob is totally redundant, given that user has already selected appropriate load type via middle knob and hence can be removed.

- Now add gimmicky '*Deep Fill'* extra water button to the mix. Once I select a particular *Water-Level*, does '*Deep Fill'* overrides it or vice-versa ? Never understood this. Another example of redundant option.
- Middle knob has option *bulky* & *heavy* . To me, they seem the same and add to the confusion.
- Second right dial 'Options' is another extraneous dial. Who needs '*2nd Rinse*' or '*2nd Rinse + Extra Spin'*, if machine has done its job of washing properly.

Elimination of above-mentioned dials/options will help in minimizing the clutter and make interface cleaner and intuitive

Fig 4b: Check out options on Temperature dial – *Tap Cold*, Cold , Cool , *Color*, Warm

5 APPENDIX

Visualization of Registration redesign described in section 1.2.

Once student has logged in and selects *Registration icon* , he is presented with a view which shows

1. courses student has completed so far (courses that have tickmark and grayed out)
2. Number of courses remaining to fullfill his specialization requirement. (right chart)
3. If there are any registeration holds.
4. Relevant courses within the context of each requirement, which user can select via check box.
5. Second column from the right displays '*Open'* if there are seats available, '*Waitlist-x*' waitlist number , '*Closed'* – Waitlist is limit reached.

Student : **Jane D** Specialization : **Machine Learning** Registeration hold : **None**

MAJOR

Core

CS 7641	Machine Learning	Spring 2018	✓
CS 6515	Intro to Grad Algorithms	Spring 2019	✓

Electives

CS 7646	Mach Learn For Trading	Spring 2017	✓
CSE 6242	Data & Visual Analytics	Spring 2017	✓
☑ CSE 6250	Big Data Health	open	
☐ CS 7642	Reinenforcement Learnir	waitlist -53	

FREE ELECTIVES

CS 6035	Intro To Info Security	Summer 2017	✓
CS 6250	Computer Networks	Fall 2017	✓
CS 6262	Network Security	Summer 2018	✓
CS 7637	Knowledge-Based AI	Fall 2016	✓
☑ CS 6750	HCI	open	
▩ CS7637	KBAI	closed	

....

submit