Assignment M3

Piyali Banerjee
pbanerjee32@gatech.edu

***Abstract*—**Microsoft Outlook is a popular email and calendar management tool used by many institutions across academia and the industry. When Outlook users schedule meetings, there is often a need to compare schedules to find an optimal time for the potential attendees. There are several methods a user can use to schedule a meeting, such as the Scheduling Assistant add-on and importing calendars from a directory and overlapping schedules. This project explores a potential redesign of the Outlook Calendar interface to improve the meeting scheduling assistance functionality via the design lifecycle.

1 BRAINSTORMING PLAN

I will conduct a 30-minute individual brainstorming session directly in Section 2 of this document. At the 15-minute mark, I'll take a short break. Prior to the brainstorming session, I will review the survey and interview responses from the needfinding exercises done previously, because it would serve as a strong foundation for generating ideas. The main goal of the brainstorming activity will be to think of ways to make the experience of scheduling meetings on Outlook more intuitive for novice Outlook Calendar users and to make scheduling features more discoverable.

Through the needfinding activities, I learned about a few pain point areas that Outlook Calendar users face when scheduling meetings:

- Inconsistency between the Mac desktop application and web browser
- Learning curve for novice Outlook Calendar users
- Lack of discoverability for more advanced functionality for expert users
- Calendar overlap visualization may be confusing

I will consider my brainstorming session complete once I have at least 30 ideas spread across these pain point areas (~5-10 ideas per category).

2 BRAINSTORMING EXECUTION

- Inconsistency between the Mac desktop application, mobile application, and web browser
    - New advanced version with feature parity
    - Ability to invite optional attendees in mac desktop app
    - Add scheduling assistant in mobile app
    - Add overlapping calendar in mobile app
    - Add optional invitees to mobile app
    - Separate outlook calendar mobile app
- Learning curve for novice Outlook Calendar users
    - Voice activated calendar helper/schedule assistant
    - Intro tutorial easily accessible from pop-up toolbar
    - Save event drafts automatically
    - FAQ accessible from help section
    - Help text for various icons in event creation window
    - Interaction with user groups - pop-up suggestions
    - Schedule assistant open by default
    - Make scheduling with groups work with scheduling assistant
- Lack of discoverability for more advanced functionality for expert users
    - Suggestions for shortcuts to repeated actions
    - Pop-up toolbar with additional functionality/extensions
    - Scheduling meetings with group - tutorial
    - Frequently invited ppl - suggested invitees list
    - Charms help guide/use cases
    - Suggest add-ins to help with frequently used actions
    - User group formation/invitation add to scheduling assistant
- Calendar overlap visualization may be confusing
    - Alternate color schemes for color blind individuals
    - Pattern choices for availability categories
    - Voice activated scheduling assistant
    - Automatically show next available time
    - Ability to exclude optional invitees from schedule overlap requirements
    - Toggle for optional invitees
    - Month to month view
    - Minimal categories -- ie: only “free” and “busy”

- Make design of scheduling assistant calendar overlap consistent with the default overlap in the meeting invite window

3 SELECTION CRITERIA

Improving learnability, particularly for novice users, will be the driving factor and main requirement behind redesigning the interface, since this was discovered to be a major pain point based on the needfinding exercises.

To guide the selection of three ideas from the list of ideas generated in the brainstorming session in Section 2, I've created two personas based on what I learned about the users from the needfinding exercises and the data inventory: 21-year-old Ashley who is a fresh college grad who just started her career at a tech company as a software engineer and 45-year-old Nathan who has moved to a new company as the head of HR. Ashley considers herself to be very tech-savvy, having been surrounded by electronics and computers since she was a toddler and having just completed a BS in CS. She has primarily used Gmail throughout her educational journey and for her personal use and has never used Outlook prior to her new job. Ashley uses a MacBook Pro laptop. Nathan is also new to Outlook and had 20+ years of experience working with various meeting scheduling technology and people and resource management technology. Nathan is not as tech savvy as Ashley since he grew up in a rural area without much technology beyond a home telephone and household appliances. He typically relies on tutorials, documentation, and help from an interface to become familiar with it. Nathan uses a Windows PC. With these personas in mind, the goal is to select three ideas to improve the learning experience for Ashley and Nathan as they navigate scheduling meetings with users and user groups with Outlook Calendar.

The first idea selected from the design alternatives is the voice activated Scheduling Assistant. According to the surveys and interviews I conducted for the needfinding exercises, the Scheduling Assistant is the most-used Outlook Calendar tool to help find optimal times for meetings based on the availability of the invitees. Ashley often uses the voice-activated virtual assistant on her laptop (Siri for her MacBook Pro) to check the weather, ask for quick facts and trivia, and take notes. Nathan doesn't regularly use Cortana on his Windows PC, but has some familiarity with it and has used it a few times. If Outlook Calendar leveraged the virtual assistants to allow users to quickly schedule meetings

without even having to directly interact with the interface, the interface would become more invisible and the learning curve would be practically non-existent, especially for users such as Ashley. For users such as Nathan who have none to little familiarity with using virtual assistants, the learning curve would potentially still be significantly lower since they would not have to learn the many subtasks involved with scheduling meetings with the Outlook Calendar interface. This idea therefore was selected due to the potential for improving learnability for novice users such as Nathan and Ashley by making the interface more invisible and leveraging existing technology.

The second idea selected from the design alternatives is minimizing categories representing a user's status for a scheduled time. One subtask discovered through the needfinding exercises is for a user to select their status for the event. They could be shown as being Free, Working elsewhere, Tentative, Busy, Away, or Private, as shown in Figure 1. I learned from interviews that these statuses are typically confusing for novice users, and users often forget this subtask entirely which may cause unwanted outcomes -- for example, if Ashley/Nathan creates an all-day event, the status is set automatically to "Free" indicating that Ashley/Nathan are free to meet with other users on that day. As novice users, Ashley/Nathan may not be aware of this subtask and may intuitively think that if they are scheduling an all-day event, they are naturally busy that day. Reducing the statuses to "Free", "Busy", and "Away" and having the status be "Busy" by default for any event being scheduled regardless of duration would improve learnability of the interface significantly by narrowing the gulf of execution of the subtask and potentially removing the need for novice users to worry about the subtask entirely.

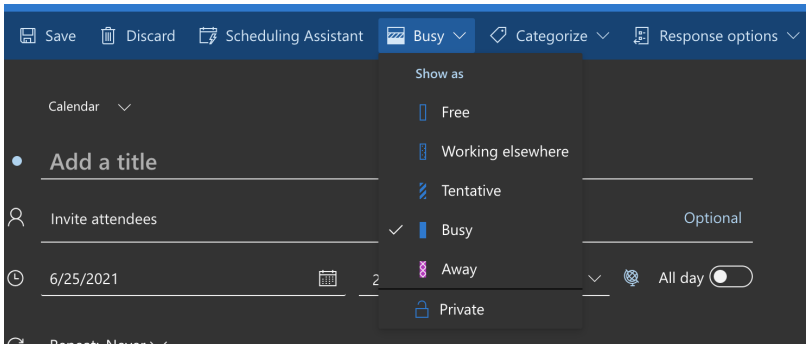


Figure 1—Outlook Calendar meeting status options.

The third idea selected from the design alternatives is a pop-up window or text box showing suggestions on how to schedule meetings with user groups easily. Outlook has a user groups feature that teams can use as both an email list and a way to create team resources such as team calendars where they can schedule recurring meetings. There could be a scenario where Ashley wants to schedule a meeting with a different team for a cross-team collaboration sync. Having pop-up help texts guiding her through the subtasks would ultimately improve the learnability of the interface and would reduce the gulf of execution.

4 PROTOTYPE 1: WIZARD OF OZ

4.1 Prototype

The premise of this prototype would be to provide a script to the user with commands they can give to the virtual assistant on their device to schedule meetings with other users. I would pretend to be the virtual assistant and would guide the user through scheduling and confirming the details of the event.

1. **Script to user:** Hello! We can assume you are logged in to your Microsoft Office account prior to this exercise. Please say the following to your virtual assistant: "[Virtual assistant greeting such as "OK Google" or "Hey Siri"], I want to schedule a meeting on Outlook Calendar."
2. **Virtual assistant response (from me):** Sure, who would you like to invite?
3. **Script to user:** When prompted by the virtual assistant for users or user groups you would like to invite, provide their full names one at a time. Let's say you want to invite John Smith, Riley Kane, and the user group called Team Dynamite. You would start by simply stating "John Smith"
4. **Virtual assistant response:** Would you like to invite John Smith, email address jsmith@work.com? Please respond with "yes" or "no."
    a. **Expected user response:** "Yes" since John Smith is their intended invitee.
    b. **Virtual assistant response**: Great. Would you like to invite anyone else? Please respond with "yes" or "no."
    c. **Expected user response:** "yes"
5. Repeat steps 2-4 for Rile Kane and Team Dynamite.

6. Script to user: When prompted by the virtual assistant for the meeting duration, please provide a response in minutes and hours (ie: 1 hour 30 minutes)
7. Virtual assistant: What is the intended duration for the meeting?
    a. Expected user response: 30 minutes
    b. Virtual assistant: The duration is 30 minutes, correct? Please answer "yes" or "no"
    c. Expected user response: yes
8. Script to user: When prompted by the virtual assistant for preferred meeting days, say "No preference" if you have no preference or say a preferred date with the month followed by the date (ie: July 5th)
9. Virtual assistant: Which day would you like to have this meeting? If you have no preference, please say "No preference"
    a. Expected user response: "No preference"
        i. Virtual assistant response: "Great, the earliest available time is today at 3-3:30 PM. Should I schedule the meeting for you at that time?
        ii. Expected user response: Yes.
        iii. Virtual assistant response: I have scheduled a meeting for you, John Smith, Riley Kane, and Team Dynamite for 3-3:30PM today.

4.2 Evaluation

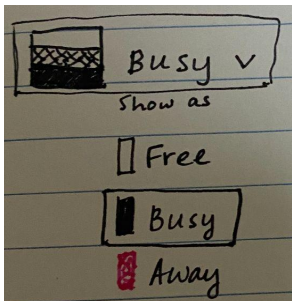
The prototype meets the learnability requirement as it would narrow the gulf of execution for novice users by making the Outlook Calendar interface more invisible as the user would only need to interact with the virtual assistant to schedule meetings. This capability would minimize the need for users to learn the various subtasks for scheduling meetings as well since the virtual assistant would automatically walk them through all the steps necessary. It also meets the compatibility requirement as virtual assistants are available across devices and platforms. For example, Siri is available on iPhones as well as Mac laptops -- therefore, this capability to leverage virtual assistants for scheduling Outlook Calendar meetings should work on any device with the virtual assistant. It meets the functionality requirement as well -- the core functionality of enabling users to schedule meetings will not be altered with this new capability. This prototype and design alternative would be valuable for both novice and expert users due to the resulting improved learnability, as well as the potentially improved efficiency

resulting from the automation the virtual assistant will provide in finding optimal times that expert users may be interested in leveraging.

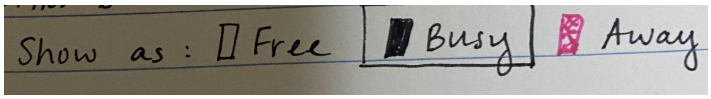
5 PROTOTYPE 2: PAPER

5.1 Prototype

Option A: re-design the "Busy" drop down shown in Figure 1 to only have Free, Busy, and Away options and always default to Busy:



Option B: rather than the "Busy" drop down, add the following under the "Invite attendees" entry section:



Ask users which design they prefer and their thoughts on the designs.

5.2 Evaluation

The prototype meets the learnability requirement as it would narrow the gulf of execution for novice users by minimizing confusion around the status options and making the subtask of selecting a status more explicit. It also meets the compatibility requirement as the functionality would be added to the mobile app version and Desktop app versions of Outlook Calendar. It does slightly modify functionality by removing status options so this design does not meet the functionality requirement; however, this can easily be addressed by offering settings for the status from which a user can add the other statuses to their available options. This design and prototype is applicable for all Outlook Calendar users, including novices and experts.

6 PROTOTYPE 3: TEXTUAL

6.1 Prototype

From the interviews I conducted as part of the needfinding activity, I learned that there are multiple ways to schedule meetings with user groups. One of the intuitive ways is to use the same method as inviting users individually by finding the user group from the active directory in the "Invite attendees" section in Figure 1. However, currently this method erroneously shows group calendars as being constantly unavailable even when that's not the case. First of all, this functionality should be fixed to present the true availability of the user group. Then, a pop-up textbox or window showing suggestions on how to schedule meetings with user group would help guide novice users such as Nathan who are not as tech savvy easily understand the subtasks necessary to schedule the meeting. The help text box would appear next to the "Invite attendees" section as the user is searching for a user group to invite. The text would offer options on how to schedule a meeting with a user group (ie: providing a link to the user group's calendar, method of finding a group a specific user belongs to, methods of finding optimal times with a group), and would be structured as an FAQ page. There would also be a "Don't show this again" checkbox at the bottom of the pop-up if the user feels that they are familiar and don't require the help text anymore. However, if they need to access the FAQ page again, they would click a "?" icon that can be added next to the "Invite attendees" section. This pop-up should appear on the mobile app and Desktop app versions of Outlook Calendar as well and should work across all devices.

6.2 Evaluation

The prototype meets the learnability requirement as it would narrow the gulf of execution for novice users figuring out how to find and invite user groups and find optimal times for meetings with them by providing a pop-up FAQ page with multiple options. After reading the FAQ page, users can select a method that works best and is the most intuitive for them. It also meets the compatibility requirement as the pop-up FAQ page functionality would be added to the mobile app version and Desktop app versions of Outlook Calendar. It meets the functionality requirement as well -- adding the FAQ pop-up will not modify the core functionality of Outlook Calendar, but would rather guide users to learn it faster to start leveraging existing functionality more efficiently.