

Alpha - A Proposed Meeting Scheduling Applications

Ryan Marks
001406077
marksr2@mcmaster.ca

Nick Morrison
001426613
morrin2@mcmaster.ca

James Taylor
taylojlp@mcmaster.ca

Trong Tran
trantp2@mcmaster.ca

ACM Classification Keywords

H.5.m. Applied Computing: Enterprise applications

ABSTRACT

Alpha is a proposed meeting scheduling application. Existing ‘meeting request’ style approaches have issues when scheduling meetings with more than 5 participants. Purpose built meeting schedulers can still be clunky leading to slow responses from participants. Our final product will be a functional chat-bot effectively mimicking a human focused on scheduling. There will be many facets of design to be considered when creating the product, these duties will be divided among the four group members.

Author Keywords

Consumer Applications; Calendaring; Novel Interfaces, Natural Language Processing

TOPIC OVERVIEW

A common feature of almost all organizations is designated periods of synchronous auditory communication typically called ‘meetings’. When scheduling a meeting it is necessary for all relevant participants to be available for the same block of time. This gives rise to the ‘Meeting Request’ a ubiquitous feature of email+calendaring systems which allows a meeting host propose a block of time to a number of participants, who can indicate their availability. This methodology falls apart once the meeting has five or more participants because of the likelihood a proposed reschedule has conflicts for another participant. We propose a system that will leverage natural language processing to minimize difficulty for users.

EXISTING PRODUCTS

There are issues in current processes that one has to undergo when responding to meeting invites and planning meetings with others. Tools such as Need To Meet and Google Calendar, though widely used, could be improved on.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI’16, May 07–12, 2016, San Jose, CA, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 123-4567-24-567/08/06...\$15.00

DOI: http://dx.doi.org/10.475/123_4

Need To Meet, though simple to use on desktop, lacks a mobile friendly site. This causes friction in the interactions necessary to use the app and adds more work for the user; for example, the user now needs to zoom into the website and drag to navigate the input forms.

Google Calendar and other systems like it (e.g. Microsoft Outlook) are not the most intuitive to use on desktop. The user creates an event and then edits the event information. When editing the event information there is a guest list where you can enter the email address of all the guests you would like to invite. It is assumed that adding an address to the list would send an invite to the corresponding guest, but nothing happens. Instead, you have to press ‘email’ to the right of the guest list for the invitees to be notified of the event. Then, from the email sent to the user, no action can be done. The visual affordance of the main content of the email is a clickable element that would take me to Google Calendar to see more information —unfortunately hovering over the element does not cause the cursor to change to a pointer, and clicking causes no action.

FUNCTIONALITY OF FINAL PRODUCT

The final product will offer the following features:

Creation of Meeting Invite

Meeting invites will be sent from a web interface. The host will open the page, enter a name for their meeting, and some proposed times. The host will then enter the email or mobile numbers for their invitees. Invitations will then be sent to the invitees requesting they indicate their availability.

Response to Meeting Invitation

Email: After opening a link in the invite the user will be presented the proposed times and responses. They will be able to indicate if they cannot, can, or can if necessary attend each given time. They will also have the option of proposing a new time.

Text Message: People invited by text message will receive a message containing the name of the meeting, the proposed times, and a link to the same web interface as the email. For a speedier response, users can reply with a text response to indicate availability. Each option will be lettered, and the response will just include the relevant letters. Proposing new times, and indicating “can if necessary” will not be available over text.

Calling of Meeting

Once all invitees have responded the host will be informed and will have the ability to call the meeting for a certain time. Once the host has chosen a time all invitees will be informed by the same means they were invited.

WORK BREAKDOWN & DUTIES

Expected work

Interaction design

Interactions in the application will be designed in such a way as to strike an optimal balance between maximizing speed and minimizing unwanted invocations/errors.

Asset design/creation

Assets (icons, sound effects etc...) will be designed to in accordance with industry best practices and design language. Where appropriate.

Application flow design

Application flow will strive to be designed in such a way that the application can be navigated intuitively by as many users as possible.

Natural language design

Choosing keywords/hooks essential to the natural language portion of the application should be made keeping in mind as

large of a set of end consumers as possible (these may include non english first language speakers etc.)

Conversation design

Conversations with our chat bot will be designed to be as 'human' as possible. We are not claiming artificial intelligence is a main goal of the project, but we will design the bot to mimic something of that nature.

Proof of concept implementation

Seeing as we are to be providing a proof of concept to demo at the end of this project, we will inevitably need to do some implementation of the design. This may, but is by no means set a frontend implementation, a backend implementation, and api design.

Duties

James Taylor - Interaction design, Asset design/creation, Proof of concept implementation

Ryan Marks - Asset design/creation, Application flow design, Proof of concept implementation

Nick Morrison - Application flow design, Natural language design, Proof of concept implementation

Phillip Tran - Natural language design, Conversation design, Proof of concept implementation