Personal Scheduling Assistant: Automated & Ubiquitous Event Scheduling for Digital Voice Assistants

JOHN ENGLISH* and DANIEL C. RUIZ

1 INTRODUCTION & MOTIVATION

For our ubiquitous systems project, we chose to focus on the problem of modern schedule planning, and how current systems could be integrated and elevated to become more ubiquitous. In the modern day, one could argue that daily planning using digital calendars is already ubiquitous, due to the ease with which smartphones can be accessed and new events can be added to an application. What has not become ubiquitous, however, is digital assistance with the behind-the-scenes mental logistics that motivate the addition of a new calendar event. For instance, a user may have to consider conflicting events, traffic conditions, or prices at different stores before scheduling a grocery run. While performing these mental calculations, a user may consult several other applications before making a decision, which wastes valuable time in the process. We see this issue as an optimization problem central to the act of daily planning. We wish to push this problem towards a solution by proposing a new system, motivated by current literature and integrated with cutting edge voice assistant technology, that automates this optimization. We envision a system that receives a desired event from the user and returns a recommendation for an optimal place to add it to their calendar. We believe such a system, once properly networked and integrated, would save time for users and reduce physical interaction with technology while planning daily events.

We began our project by conducting a literature review that examined the art and science of planning via digital calendars to ascertain benefits, detriments, design methodologies, and cutting edge systems in development (see Related Work for a summary of significant findings). For the second half of our project, we conceptualized, designed, and implemented a software prototype that explores the potential of ubiquitous planning systems. Specifically, our prototype is tailored towards integrating automated planning with digital voice assistants (DVAs), such as Amazon's Alexa, Google's Assistant, or Apple's Siri. Dubbed "Personal Scheduling Assistant" (PSA), our prototype is designed to emulate the interaction that would occur between a user and a DVA, while performing the internal logic that results in a better-than-random scheduling recommendation (see Description section).

2 STATEMENT OF CONTRIBUTIONS

Both authors contributed equally to this research, despite slightly different responsibilities. John English's most notable contributions to the project were formulation of the original idea, literature identification/information extraction, and prototyping in Python. Daniel Ruiz's most notable contributions were report writing, poster design, and initial prototyping in the Amazon Alexa Development Environment. Both authors contributed significantly to the overall concept, design, and final implementation of the PSA prototype.

*Both authors contributed equally to this research.

Authors' address: John English; Daniel C. Ruiz.

3 RELATED WORK

To conduct our literature review, we began by looking back in history for examples of early digital calendar systems and applications. We found rudimentary scheduling systems reaching as far back as the 1980s, when personal computers first started to become ubiquitous in business offices due to the rise of word processors. For instance, Japanese researchers created a Prolog-based system for managing the schedules of hospital nursing staff in 1988 [7]. Though the system would be considered primitive by today's standards, it was still capable of simultaneously considering a variety of requirements when when assembling a day-to-day schedule for groups of nurses. This helps predict a key conclusion we made during our literature review. Concretely, we found that the vast majority of digital scheduling research has been focused on coordinating the efforts of large groups or organizations. Examples of such research vary wildly, and are often tailored towards specific tasks or events, such as Abbaspour et al's tour planning system for cities, or Cohen et al's scheduling system for music festivals [1, 5]. An interesting project published by Microsoft Research blurs the line between human and computer scheduling by creating a digital agent that uses workflows to automate the scheduling of group meetings while allowing the intervention of humans when scenarios become unusual [6]. Papers such as these tend to focus on the needs of many individuals and leverage them against each other to find an acceptable solution for the group, rather than an optimized solution for a single user. This research has been enhanced over the years by systems that specifically account for users with conflicting priorities, while even providing an interface for multiple users to collaboratively resolve global constraints [9]. While less relevant for our purposes, these "crowd-sourced" coordination systems have been shown to rival the schedules produced by digital expert systems in terms of efficacy [2]. More important, however, is the fact that we found little to no research seeking to enhance the digital experience of personal scheduling, especially within the context of ubiquitous computing.

As a result of our perceived lack of ubiquitous research, we set out to envision a system that not only alleviates some of the logistical hassles of personal event planning, but also does so in the most omnipresent manner available. To this end, we identified digital voice assistants (DVAs) as a primary contender for hosting our system. Firstly, however, we sought evidence to support our assumption that people, generally speaking, prefer to schedule their personal events using digital tools, rather than analogue. This sentiment was confirmed by a recent study that examined how Microsoft employees manage their personal and household scheduling needs. While not the most diverse group in terms of ages and annual income, we assumed for the purposes of our research that average Microsoft employees would also be more likely to utilize a digital voice assistant in their home. The study concluded that the majority of individuals use digital calendars to plan personal events, while only 38% preferred analogue methods [4]. Given that this study took place in 2005, we estimate that the majority has only grown in size as scheduling applications have become more robust and widespread. Next, we derived some conclusions from studies that focused on voice assistants themselves. Most notably, a 2018 paper by Bentley et al informed us that smart speakers (in this case Amazon's Alexa, the largest market share holder) are vastly underutilized by their owners, despite their numerous capabilities [3]. This assessment can lead one to believe that the voice assistant market is ripe for the implementation of additional utility that will have a larger impact upon people's lives. Furthering this notion, a 2019 paper published by the Head Scientist of Amazon Alexa cited enhanced "daily convenience and innovation" as the core mission of Alexa devices [8]. This statement is very much in line with the intent of our Personal Scheduling Assistant program and reinforces our belief that DVAs are an ideal enabler for ubiquitous digital planning.

4 DESCRIPTION

Based on the research above and our derived conclusions, we set out to conceptualize a system that took advantage of DVAs and their inherently ubiquitous nature. We envisioned a program, running on an abstract DVA, that prompts the user for information about a personal event they wish to schedule. This information is then optimized against a variety of external factors through the program's internal logic. Through a minimal amount of interaction with the user beyond the initial queries, the program will return a recommended block of time for the user to place the described event on their calendar. This recommendation must be better than a completely random recommendation, and maximize the amount of resulting "white space" on the user's calender. Given a robust enough implementation, PSA could even add the event to a calendar the user has connected by the DVA. The list of external factors that can be taken into account when estimating the best time to perform an event is admittedly long. Our hope, however, is that by taking into account as many of these external factors as possible, our system's recommendations will result from logistical considerations similar to what a human would ponder, and perform schedule event placement that is strictly better than random. Examples of external factors the program could be designed to consider are captured below, along with the general flow of the system (see Figure 1).

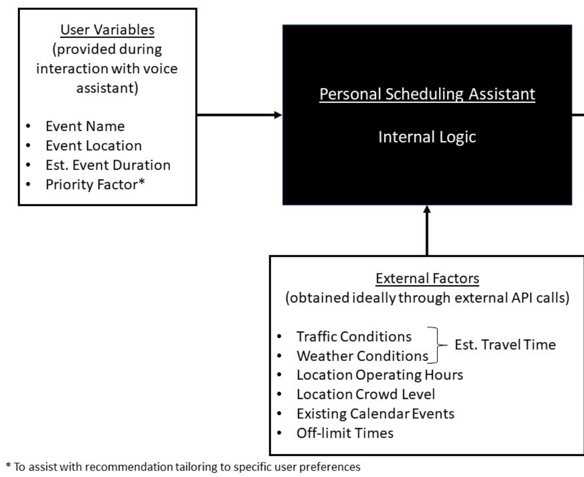

Fig. 1. PSA Concept

To illustrate what makes a system such as this ubiquitous in nature, one only needs to envision some potential use cases. For instance, consider Tina, a busy single mother who must schedule a trip to the gym around work, grocery shopping, and picking up her children from school. Assuming the aforementioned events are already on her schedule, she could then initiate the PSA program using her DVA. After recording the gym name, location, and workout duration as input from Tina, PSA would then consider several factors. Examples include estimated driving time from Tina's home or work, estimated weather conditions, the gym's hours, and the gym's peak crowd level, to name a few. Based on these factors, and Tina's preference for low crowd levels at the gym (recorded via the "priority factor" input metric), PSA could help Tina realize that she could retain white space on her schedule by going to the gym later in the afternoon, rather than directly after work during peak hours. For illustrative purposes, two other potential interactions with PSA-enabled DVAs are shown below (see Figure 2)

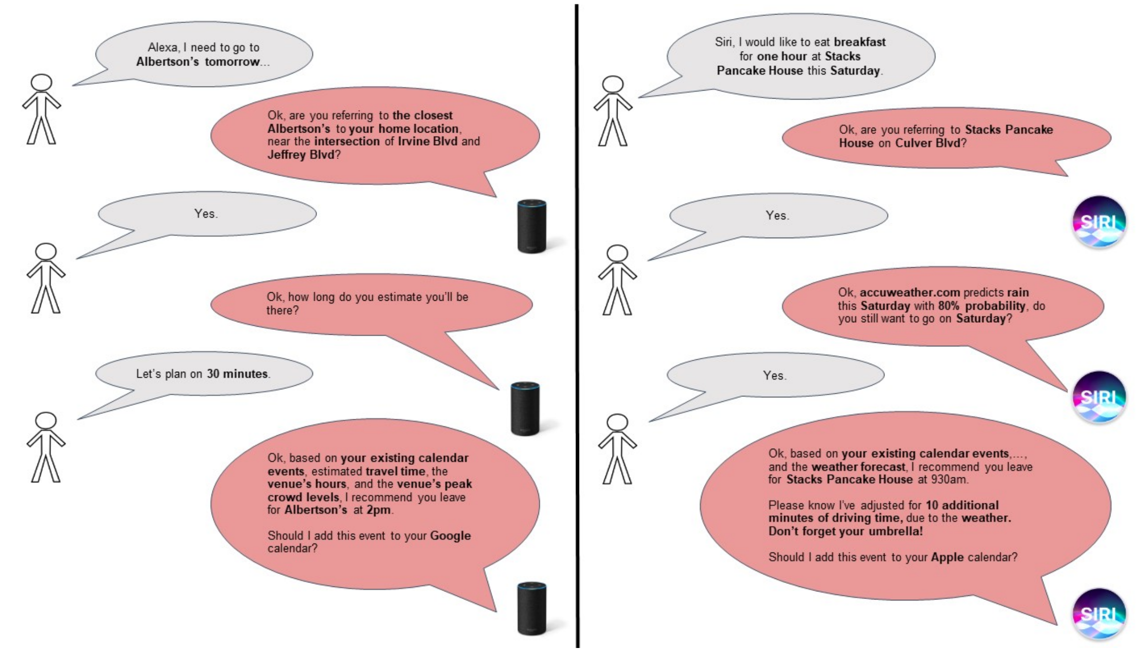

Fig. 2. Ideal Sample Interactions

An ideal version of this system would account for multiple external factors, query the user appropriately for missing information, and inform the user when certain factors are considered in the calculation. More importantly, users would often add events to their calendars in a fraction of the time it would take otherwise. Over time, the cumulative time saved over a week or month of using PSA could be significant, not to mentioned the added general convenience.

To implement the concept described above, we initially focused on Amazon Alexa systems, due to our personal familiarity with these devices and the relatively accessible development environment offered by the Alexa Skills Kit (ASK). However, we soon realized that by limiting ourselves to the ASK, the optimization problem at the core of our vision soon devolved into a data representation problem. More specifically, we found the difficulty of implementing a prototype for our concept was amplified by the need to use ASK's preferred language, parameters, and workflow. As such, we pivoted our research by developing PSA's logic outside of any corresponding DVA development environment. By doing this, we reframed the scope of our system as a prototype that could be translated to any digital voice assistant by those with the time and expertise required. Thus, we could more readily explore the potential of such a system, rather than limiting ourselves to one platform. We implemented our prototype in Python, and made it available for review at the provided Github Repository. [1] For a discussion on the capabilities and limitations of our prototype, please see below.

[1] https://github.com/JohnEEnglish/PersonalSchedulerUbicomp2019

5 DISCUSSION & FUTURE WORK

In order to follow along with the internal operation of our prototype, the following flowchart is provided (Figure 3).

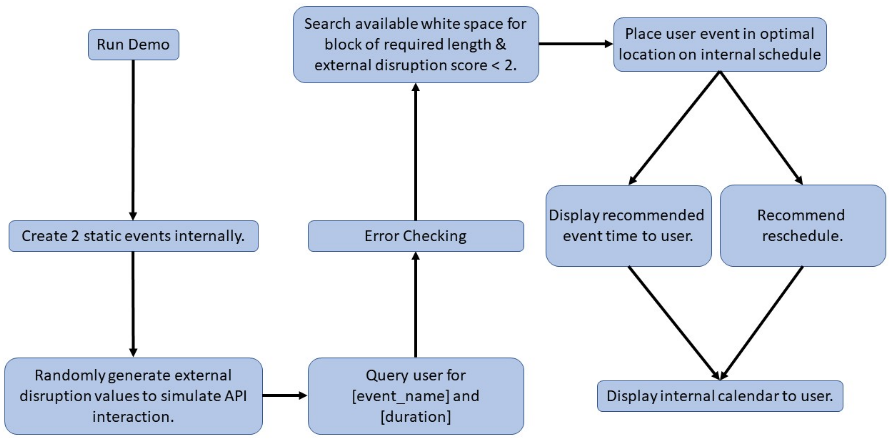

Fig. 3. PSA Concept Flowchart

By default, our PSA prototype begins in demonstration mode. It starts by initializing an internally tracked schedule that contains two static events. Users then receive a welcome message stating the intent of the program, and are prompted for the name and duration of the event they wish to add to their schedule. After receiving the required input, the PSA prototype simulates the acquisition of external factor information. For demonstration purposes, this information is represented by static values who's states are randomly assigned throughout PSA's internal schedule whenever the demo is run. These simulated external factors then aggregated into an "external disruption" score between 0 and 2 (2 representing significant delay, and 0 representing minimal). Due to time constraints and inexperience with the corresponding APIs, real-world variable states are not currently implemented in the PSA prototype, though the event class is constructed in a way that facilitates replacement of these static variables with API calls. Similarly, our prototype does not currently scrutinize external factors that affect a user's travel time to a destination (i.e. weather, traffic conditions, etc), primarily due to the inherent challenge involved in representing separate locations and routes between them, both statically and

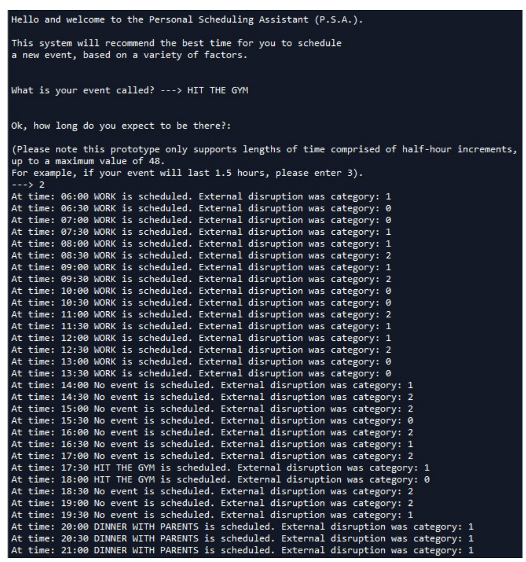

Fig. 4. Sample Output

through API calls. Based on the external disruptions for each available time slot, PSA will find the optimal place for the new event on the schedule, and make a recommendation to the user. In the event no optimal spot is found, PSA will let the user know that scheduling an event on this day is not recommended. Lastly, PSA will display a representation of the internal schedule and corresponding external disruption scores for visualization purposes. For clarity, sample output of our prototype is provided (see Figure 4). In this example, a user wishes to schedule an hour-long gym workout between their static work and dinner events. In this run, our prototype determines the optimal time to spend an hour in the gym is between 1730 and 1830.

Given the limitations of our system described above, we believe a reiteration of its capabilities to be worthwhile. After accepting user-defined data concerning a new event, the system simulates the consideration of external factors. These factors are used to identify the optimal time to schedule the event, and a recommended time is returned to the user. Despite the inherent simplicity of the prototype in its current iteration, it demonstrates how, through the automation of planning around external factors, users can save themselves time and enjoy a higher level of convenience.

6 CONCLUSION & FUTURE WORK

With more time to dedicate to this project, all future work would have focused on improve the robustness of our prototype. Beyond the inclusion of travel-time affecting variables and the substitution of static variables for real-world states, we would implement the connection of users' existing digital calendars to enable PSA to review existing events and add new ones. Next, our primary goal would be translation of the system into an existing DVA. Afterwards, we would look to open-source our work and release the program on the respective DVA's platform (i.e. Amazon Alexa's Skills Hub) to encourage community collaboration. We envision many interesting extensions to the baseline capabilities of our concept. For instance, incorporating a visual interface or GUI could enhance the way users interact with PSA, possibly by viewing recommended events on their calendars or reviewing a representation of PSA's internal logic. Of course, this would detract from the ubiquitous nature of simply speaking with digital assistants in one's home or on a smartphone. Another potential application of a PSA-like system is for self-driving cars that automatically adjust their routes based on minimizing driving time in light of fluctuating external factors.

In summary, we present a concept and prototype for a ubiquitous system we believe has the potential to enhance human lives in interesting ways. Based on the central argument that personal event planning can be heavily automated, our system envisions a capability for modern DVAs that takes the planning burden away from users. By offering schedule recommendations, users no longer have to consult multiple applications to determine the best time to execute an event, especially when the event takes place in an unfamiliar location. We would be surprised to learn we are first researchers to conceptualize such a system. As such, we look forward to seeing similar and more robust planning automations hit the market in future years.

REFERENCES

[1] Rahim A. Abbaspour and Farhad Samadzadegan. 2011. Time-dependent personal tour planning and scheduling in metropolises. In *Expert Systems with Applications*, Vol. 28. https://www.sciencedirect.com/science/article/pii/S0957417411005410

[2] Elena Agapie, Bonnie Chinh, Laura Pina, Diana Oviedo, Molly Welsh, Gary Hsieh, and Sean Munson. 2018. Crowdsourcing Exercise Plans Aligned with Expert Guidelines and Everyday Constraints. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, Montreal QC, Canada. https://doi.org/10.1145/3173574.3173898

[3] Frank Bentley, Chris Luvogt, Max Silverman, Rushani Wirasinghe, Brooke White, and Danielle Lottridge. 2018. Understanding the Long-Term Use of Smart Speaker Assistants. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Vol. 2. ACM. https://doi.org/10.1145/3264901

[4] A.J. Bernheim Brush and Tammara Combs Turner. 2005. A survey of personal and household scheduling. In *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work (GROUP '05)*. ACM, Sanibel Island, Florida, 330–331. https://doi.org/10.1145/1099203.10992638

[5] Eldan Cohen, Guoyu Huang, and J. Christopher Beck. 2017. Personal Scheduling for Concert-Goers at Large-Scale Music Festivals. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS '17)*. Pittsburgh, Pennsylvania. http://icaps17.icaps-conference.org/demos/Cohen-demo.pdf

[6] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andres Monroy-Hernandez. 2017. Calendar.help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, Denver, Colorado, 2382–2393. https://doi.org/10.1145/3025453.3025780

[7] Mihoko Okada and Masahiko Okada. 1988. Prolog-based system for nursing staff scheduling implemented on a personal computer. In *Computers and Biomedical Research*, Vol. 21. https://www.sciencedirect.com/science/article/pii/0010480988900420

[8] Rohit Prasad. 2019. Alexa Everywhere: AI for Daily Convenience. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. ACM, Melbourne VIC, Australia, 3–3. https://doi.org/10.1145/3289600.3291377

[9] Haoqi Zhang, Edith Law, Rob Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. 2012. Human computation tasks with global constraints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, Austin, Texas, 217–226. https://doi.org/10.1145/2207676.2207708