

Calendar Delegation with Alexa

Priya Andurkar
Department of Computer
Engineering
MKSSS'S Cummins College of
Engineering for Women
Pune, India

Akshat Pandey
School of Computer Science and
Information Technology
Manipal University Jaipur
Jaipur, India

Pranjali Deshpande
Department of Computer
Engineering
MKSSS'S Cummins College of
Engineering for Women
Pune, India

ABSTRACT

Electronic calendar applications currently are ubiquitous and used by most organizations to schedule events and meetings. Common commercial applications are Google Calendar, Apple iCal, and Microsoft Outlook, which are primitive in comparison with the state-of-the-art scheduling algorithms. Meetings are an essential aspect of collaboration and decision making within organizations and scheduling them is often a time-consuming task involving a lot of communication and tedious information processing. Proper scheduling of meetings using calendars of employees in organizations can lead to efficient time management of employees and the completion of more tasks. The system aims at implementing an intelligent and autonomous calendar delegator that uses the popular virtual assistant, Amazon Alexa, entrusted with the task to schedule meetings of participants without their intervention. Voice-user interfaces (VUI) use speech recognition technology to allow users to interact with their peripherals with their voices. VUI enables hands-free, efficient interactions that are more human in nature and has led to the development of famous Virtual Assistants such as Amazon Alexa, Apple's Siri, Google Now and Microsoft's Cortana.

General Terms

Intelligent Systems, Enterprise Bots, Calendar

Keywords

Alexa Skill Kit, Amazon Web Services, Intelligent Virtual Assistant, Meeting Scheduler, Voice User Interface

1. INTRODUCTION

In an organization, everyone has multiple meetings and commitments to adhere to. A calendaring tool stores details of people's day to day schedules. Scheduling meetings is an important activity in all organizations and usually employs people such as personal assistants for the tedious, time consuming and iterative task. A meeting scheduling task usually requires a lot of manual efforts in communication and negotiation among attendees since they may have different calendars, constraints, and preferences. When setting up a meeting, the participants need to reach a mutual agreement to hold the meeting which is subject to their personal constraints and preferences. A variety of calendaring applications are used to assist people to schedule their meetings. Information workers are people who work daily with information and process it to derive actionable tasks such as:

A human resource personnel needs to sort candidates, schedule interviews and coordinate the overall hiring process. A product manager needs to coordinate product requirement gathering, engineering resource planning, marketing strategy, etc. Many of these activities rely heavily on using the calendar

in order to communicate with people, leading to the task of organizing their calendar. Information worker spends a significant amount of their working hours in organizing calendars and scheduling meetings. With the advent of Intelligent Virtual Assistants like Alexa, Siri, Cortana, the responsibilities and benefits of personal assistance can be automated for every worker in an organization. Delegation of the task of meeting scheduling and calendar organization can be efficiently performed by Intelligent Virtual Assistants. This removes the manual dependency. Meetings are an essential aspect of collaboration and decision making. A well sought after schedule with proper time slots increases the efficiency of people participating in those tasks. The use of Alexa as a personal assistant in the enterprise domain is attributed as the motivation for this system. The system implements methods to organize and schedule meetings within an organization using Amazon Alexa.

For example:

1. Alexa, what is on my calendar - will read out the events on the calendar.
2. Alexa, schedule a meeting on Monday, 3 pm – will create an event on the calendar.

Alexa currently provides calendar integration with Google, Office 365 and iCloud. It is inherently reactive; this means that the user needs to pass command through voice to perform certain tasks. The system would try to make Alexa intelligent enough, so that the user can delegate tasks to their Alexa enabled applications and devices, and from there on Alexa will be able to perform actions without the user instructing it to do so. The delegate commands are predefined rules which will be stored in the system by the user according to their preferences and will be acted upon before scheduling an upcoming event.

2. LITERATURE SURVEY

There have been various contributions in the field of intelligent enterprise calendar management tools. Finding a common free timeslot for all the participants in a meeting is the most basic goal of a meeting scheduler. The research done in the field of decision making in a distributed meeting scheduling system by using multiple agents [1], the concept of Quorum and using a two-way negotiation mechanism to reach the quorum is proposed. In this method, the process of scheduling meetings consists of a host agent sending a request to all the invitees. This invitation consists of parameters such as start time, end time, durations and members for the event. Each meeting consists of certain fixed or compulsory members and others are flexible members, whose presence may or may not affect the commencement of the meeting. The requesting or the host agent decides on certain time slots

depending on the duration, the starting, and ending time as specified in the request. Individual invitee agents send their bids for each of these requested time slots by scrutinizing their existing calendar. These bids are evaluated and in case of ambiguity, the negotiation phase begins. Negotiation is done in two rounds, in first, only the bids of the compulsory members of the meeting are negotiated and in second the flexible members of the meetings are considered. If a common consensus is met following the negotiations, then the meeting time is booked and is sent to all invitees. In this way, the system achieves scheduling meetings in a manner that is most convenient for all members of the meeting. In another approach Kozierok and Maes [2], suggest providing the user with a learning agent to schedule meetings on their behalf. A trust relationship is built with the agent gradually by observing the user's activities and receiving direct feedback. Advantages of this approach are that it requires less work from the end-user and application developer. Further, the agent is more adaptive over time and the agent can be customized to individual user preferences and habits. The user can decide how far to delegate tasks and can even decide not to delegate any tasks at all until he or she becomes sufficiently confident in the agent's predictions. Memory-Based Learning [3] and Reinforcement Learning are used for implementation of the above method where the agent it keeps a memory of everything the user does and stores them as situation-action pairs. It also maintains a set of priority weightings for meeting topic keywords, and for the relative importance of other participants. Meeting scheduling problem can also be viewed as a Constraint Satisfaction Problem [4] where solving the problem consists of scheduling all the meetings while satisfying all the constraints related to both the users and the meetings. Constraint Satisfaction Problems are a special type of search problems in which a state is defined by variables X_i with values from domain D_i and goal test is a set of constraints specifying allowable combination of values for subsets of variables. Many real problems in AI can be modelled as Constraint Satisfaction Problems. Bots, Virtual Assistants and Virtual Agents are known widely in personal spaces and have increased accessibility and productivity. Although a different set of rules apply to them in the enterprise context. A trend towards automation in both domains is observed. An Enterprise Bot is an automated user service that provides casual and conversational interactions with complex enterprise systems and processes. Enterprise Bots can for example answer questions or perform smaller tasks. A user can interact with it by just typing or speaking a request in natural language. It can only act in a passive way, meaning it must actively be triggered by the user [5]. Enterprise Bots support companies to automate processes and make the user more efficient around the digital workplace. The referenced model, helped in answering the questions relating to the factors that influence the

intention-to-use of Enterprise Bots for users and how strongly do the employees trust the Enterprise Bots in the usage intention? Virtual assistant capabilities are commonly called "skills" [6]. Skill functionalities range from basic (such as timers, jokes, and reminders) to advanced (such as music playback, calendar management, and home automation). All major assistants have provided development kits that have empowered third-party developers to create their own skills. Skills are provided with the capability to be invoked independently, they can be linked together within a single voice command to invoke a routine which has been pre-programmed by the developer, or in a sequence as required by the related skills for completion of complex tasks. Unlike

popular smartphone apps, assistant skills are most likely invoked on headless devices which lack display. To enable a Skill, a user needs to simply invoke it on their Alexa device. The Alexa devices act as an agent between the cloud-based backend of the developed Skill and the user; hence these invocations do not require any prior installation as the Skills are not installed on the devices. Some Skills need the user to link their Alexa account to another account which exists on another service. For example, when the user uses the Skill to access Uber with Alexa for the first time, the user is needed to link to their existing Uber account, which will further allow the Skill to identify and serve the user whenever they access it [7]. Discovery of virtual assistants' capabilities has been a challenge. Discovery of a required skill has been posed by two major challenges, the first is how capable can your skill be with respect to its actions and second is the abundance in the number of skills which are being developed rapidly. Keeping these challenges in mind the system has been implemented to overcome the existing limitations and to increase the footprints of enterprise bots.

3. SYSTEM DESCRIPTION

The system is implemented in the form of an Alexa Skill. To be able to use the skill and delegate tasks to Alexa, the user must link their Google calendars to the skill. OAuth 2.0 is an authorization protocol that allows the skill to gain limited access to restricted resources. Upon enabling the skill, the user is redirected to Google's login page whereupon they can log in with their credentials to link their google calendars to the skill. After successfully linking their calendars, the user can interact with their Alexa application or Alexa enabled device to query their own calendars, create events, update events, query others' calendars and delete an event. The skill is also enabled to provide the user with the ability to find free slots for desired calendars, this way they can decide on an appropriate time for scheduling an event without having to compare each other's calendars manually. Each of these voice commands are taken as inputs, and the intents, which represent the action which needs to be taken to fulfil these requests are invoked. These intents provide details that are necessary to use the Google Calendar APIs to retrieve and store information. The skill also enables every user to store their preferences for scheduling meetings, such as: "Do not accept meetings from 4 pm to 5 pm on Friday evenings", "Always accept Project Design meetings".

The preferences of every user are stored in DynamoDB by invoking the intent to add rules. The user can store rules for accepting, declining or avoiding (accept/decline) the meetings. These rules can contain information regarding the day, time, topic, and/or the organizer of the meetings. They can also be given implicit priorities for resolution of conflict during the decision making for scheduling meetings. A conflict can appear when the premise to accept an event matches the premise to decline an event and the conclusion with greater priority is chosen for the same.

The user's stored preferences help the skill in making intelligent decisions on whether to accept or decline an incoming calendar request. A daemon service is created which gets notified of the changes in the google calendar with the help of the push notification API. The details of these changes can be extracted to create an assertion. The stored rules for the user are considered as the knowledge base and are interpreted in the "IF-THEN" format which suggests, if a certain premise is met then the conclusion would be as stated. For example, a rule stored by the user would be interpreted as shown in Table 1.

Table 1. Table captions should be placed above the table

Rule	Premise	Conclusion
“Accept design meetings on Tuesday evenings”	Topic: Design, Event Day: Tuesday, EventTime:3pm-4pm	State: Accept

With many such rules forming the knowledge base, forward chaining is used to derive maximum knowledge. This algorithm starts from facts which are known, triggers all the rules whose premises are satisfied, and adds their conclusion to the known facts. This process repeats until the problem is solved. In case of conflicts, the use priorities which are also predefined can be used to conclude whether to accept or decline the meeting. The decision is communicated to the organizer and the user without any manual intervention via email notifications. If the user wishes to change the decision made by Alexa, then the user can modify their calendar using voice commands.

4. EXPERIMENTAL ANALYSIS

The experimental setup to validate the developed system

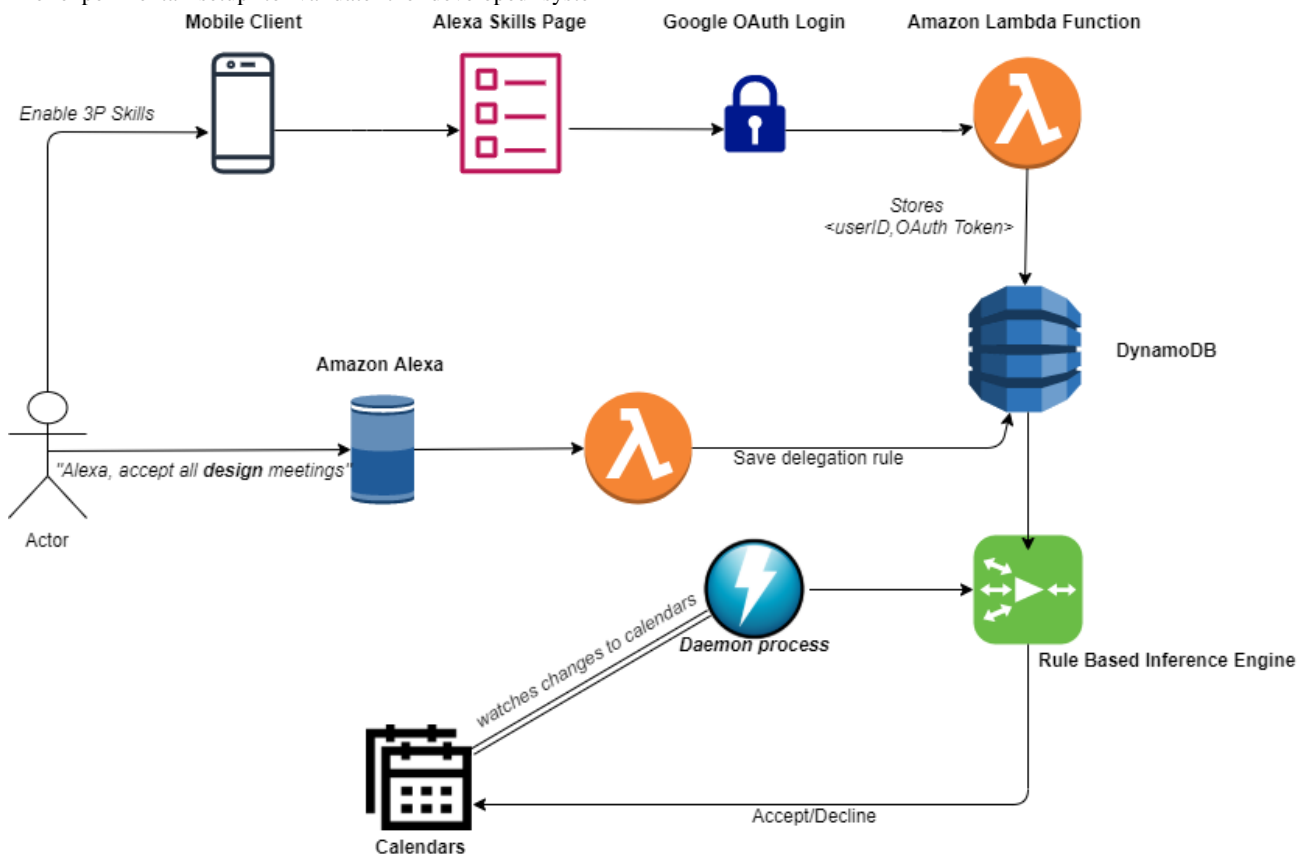


Figure 1: System Architecture

5. CONCLUSION AND FUTURE SCOPE

The system can currently make intelligent decisions of whether to accept or decline the incoming change to the user’s calendar. This is achieved with the predefined rules stored by the users themselves. The system can be enhanced to generate these rules on its own. Inductive Learning Algorithm can be implemented as it is an iterative and inductive machine learning algorithm. This algorithm is predominantly used for generating classification rules which are of the “IF-THEN” form. The Inductive Learning Algorithm overcomes the shortcomings of the rules generated by the Decision Tree

consisted of two Amazon Echo Dot devices, two personalized Google calendars, and two personalized Amazon Alexa applications on smartphones of two users namely User-A and User-B, respectively. The echo dot was configured and connected to the Alexa application for each user. Set up was completed by enabling the developed calendar skill on the Alexa application. Once the skill was enabled, the calendars of the users were linked via the Google OAuth login service. Upon completion, the users could converse with the calendar skill using their echo devices and voice commands.

User-A was able to successfully create calendar events, delete calendar events and update the details of certain existing events. User-A also stored the following rule through their device: “Always accept review meetings”. User-B queried to find the common free slots for both User-A and themselves. Then User-B used their device to send a calendar invite to User-A in the free timeslot and titled it as “Review”. The daemon process was able to detect this invitation, and with the help of the pre-defined rules set by User-A, the decision to accept the invite from User-B was taken by the skill on behalf of the user.

Learning Algorithm as they lack generalization of rules, are difficult to analyze and have slow performance. This can be achieved by tracking the user decisions and storing them. The details of every meeting which is accepted or declined by the user can be stored and this data can be used to further generate more rules for more precise decisions to be taken by the system in the future.

The current system makes use of contemporary technology to solve a problem that occurs in most workplaces. Meeting scheduling is indispensable in any organization and manual solutions are still being used. To overcome the challenges of

the ongoing methods, the system leverages the Voice User Interface of Alexa and the extensively available application package interfaces of G-Suite calendars to model an intelligent meeting scheduler.

6. REFERENCES

- [1] A. Ashir, K. Hyoun, J. Tetsuo and K. N. Shiratori, "Multi-Agent Based Decision Mechanism for Distributed Meeting Scheduling System". IEEE, 1997. Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.
- [2] R. Kozierok, P. Maes "A Learning Interface Agent for Scheduling Meetings". ACM, 1992.
- [3] Saruladha K., Priya C. and Rohini S., "Distributed Meeting Scheduler – A Knowledge Based Approach to Schedule Meetings". IEEE, 2007.
- [4] E. M. Shakshuki, S. M. M. Hossain, "A personal meeting scheduling agent". Springer, 2013.
- [5] Stefan Stieglitz, Florian Brachten, Tobias Kissmer, "Defining Bots in an Enterprise Context", Thirty Ninth International Conference on Information Systems, 2018.
- [6] Ryen W. White, "Skill Discovery in Virtual Assistants", ACM, 2018.
- [7] Irene Lopatovska, Alice London Griffin, Kelsey Gallagher, Caitlin Ballingall, Clair Rock, Mildred Velazquez, "User recommendations for intelligent personal assistants", Journal of Librarianship and Information Science, 2019.
- [8] I. Refanidis and N. Yorke-Smith, "On Scheduling Events and Tasks by an Intelligent Calendar Assistant". Association for the Advancement of Artificial Intelligence, 2009.
- [9] S. Sen and E. H. Durfee, "A Formal Study of Distributed Meeting Scheduling: preliminary results.". ACM, 1991.
- [10] S. Sen, T. Haynes and N. Arora "Satisfying User Preference While Negotiating Meetings". International Journal of Human Computer Studies, 1996.
- [11] J. Cranshaw, E. Elwany, T. Newman, R. Kocielnik, B. Yu, S. Soni, J. Teevan, A. Monroy-Hernández, "Calendar.help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop". ACM, 2017.
- [12] Leonardo Garrido, Katia Sycara, "Multi-Agent Meeting Scheduling: Preliminary Experimental Results". Association for the Advancement of Artificial Intelligence. 1996.
- [13] M. Sugumaran, K. S. Easwarakumar, P. Narayanasamy, "A New Approach for Meeting Scheduler using A*-Algorithm", IEEE, 2003.
- [14] Selwyn Piramuthu, Narayan Raman, Michael J. Shaw, Sang Chan Park, "Integration of Simulation Modeling and Inductive Learning in an adaptive decision support system", - Decision Support Systems - Elsevier Science Publishers B.V, 1993
- [15] Stephen Muggleton, Luc de Raedt, "Inductive Logic Programming: Theory and Methods", The Journal of Logic Programming, 1994.
- [16] Gustavo López, Luis Quesada, Luis A. Geurrero, "Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces", International Conference on Applied Human Factors and Ergonomics, 2017.
- [17] Matthew B. Hoy, "Alexa, Siri, Cortana and More: An Introduction to Voice Assistants", Medical Reference Services Quarterly, 2018.
- [18] Io, H. N., and Lee, C. B., "Chatbots and Conversational Agents: A Bibliometric Analysis," IEEE, 2017.
- [19] Alhadlaq A, Tang J, Almaymoni M, Korolova A, "Privacy in the Amazon Alexa skills ecosystem", 17th Privacy enhancing technologies symposium, 2017.