proceedings

Proceedings

Architecture to Design Booking Appointment Applications for the Smart Personal Assistant Alexa †

Andrés Meza *, Gustavo López, Luis Quesada and Luis A. Guerrero

Programa de Posgrado en Computación e Informática, Ciudad Universitaria Rodrigo Facio, University of Costa Rica, 11501-2060 San José, Costa Rica; gustavo.lopez_h@ucr.ac.cr (G.L.); luis.quesada@ecci.ucr.ac.cr (L.Q.); luis.guerrero@ecci.ucr.ac.cr (L.A.G.)

* Correspondence: andres.meza@ucr.ac.cr (A.M.); Tel.: +506-89710216

† Presented at the 13th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2019, Toledo, Spain, 2–5 December 2019.

Published: 20 November 2019

Abstract: The intelligent smart assistants are becoming more interactive and helpful for everyday tasks. The Amazon Echo has potential for advanced voice interactions and as a tool for conducting complex tasks. The potential of the Amazon Echo in the area of booking appointments is not being fully exploited by developers. A flexible architecture for developing appointment booking applications for the Amazon Echo was proposed. The architecture serves as guide for developers without experience working with Voice User Interfaces and saves development time by abstracting the complexity of voice interactions. A prototype skill was developed following the architecture principles and evaluated by a group of users. The skill successfully defines how an appointment booking skill should be.

Keywords: Alexa; appointments; Amazon Echo; skill; AWS; VUI; architecture

1. Introduction

Intelligent personal assistants are devices that are becoming more interactive and easier to use for everyday tasks, especially the ones controlled by natural language [1]. Most of the applications available in these devices do not provide a mechanism for businesses to engage with their users. There is also a learning curve for developers when it comes to create complex voice experiences. Addressing these issues is the main motivation for this work.

Amazon Echo is a brand of smart speakers developed by Amazon.com. The devices connect via internet to the voice-controlled intelligent personal assistant service called “Alexa Voice Service”, which responds to the name “Alexa” [2]. Amazon Echo applications are called skills [3], the users send commands to the skills using voice only, the skills allow to play music, make to-do lists, set alarms, broadcast podcasts, play audiobooks and get information on weather, traffic, sports, news and other information in real time [4]. The name “Alexa” is often used to identify the physical Amazon Echo devices. Creating voice interaction applications for the Amazon Echo or any other smart assistant requires at least some degree of VUI (Voice User Interfaces) design knowledge, e.g., being familiar with concepts such as utterances, intents, and slots [5,6]. Making the voice agents vast enough to cover almost every usage context of a user story can also be challenging [7].

Despite the fact that amazon provides tools like the Alexa Skill Kit [3] to develop vast experiences, achieving this is usually not a straightforward process for new developers. This leads to the publication of poorly designed VUIs that eventually result in frustration for the final users [8].

The context of this research is limited to appointments in the dentistry field, where an appointment consists of an agreement between a user and a company, so the user can receive a service at a given time and place, optionally an assessor who is the person who provides the service is also considered. Most of the appointment skills available for Alexa at the time of creation of this study cannot be used as channel of interaction between customers and the companies that provide such skills, sometimes they fail to perform the reservation, offer confusing user experiences and have limitations for developers. These drawbacks will be discussed in detail in the background section. It is important to highlight that not all types of appointments are the same, there are variations in their requirements that add even more complexity to this problem. For the reasons exposed, it is necessary to look for an adaptable mechanism that allows developers to be abstracted from the complexity involved in creating booking skills using voice user interfaces, and at the same time allow real interactions between users and businesses.

In this paper, a voice interaction architecture for the Amazon Echo is presented, focused on abstracting Amazon Alexa developers from the complexity of building a VUI for appointment booking in most of its variations. The purpose of the proposed architecture is to serve as template and guide for developers. This approach allows developers to simply fill the gaps in the provided architecture and configure it for their specific needs, Alexa blueprints works in a similar fashion for simpler experiences [9]. This work provides the interaction architecture documentation, the backend code for the prototype skill, and also the code for a business facing back office website that allows one to verify the appointments and configure the prototype. The elements described above are in charge of providing the connection between the final user and the business where he wants to book. Following the guidelines in the proposed architecture, a prototype skill focused on booking appointments for a dental clinic was designed, the usability of the prototype was evaluated by a group of users.

This document is structured as follows: Section 2 describes the background, Section 3 presents the proposed architecture, Section 4 shows the architecture implementation and configurations, Section 5 contains a brief description of the conducted evaluation, finally Section 6 covers the conclusion and future work.

2. Background

During the process of this research, a review of academic literature was conducted, however, the literature does not show much evidence about studies related to the implementation of skills in academic repositories, this is potentially due to the fact that most of the developers are focused on construction, therefore, this section focuses mainly on related works that belong to the industrial field. There have been some recent developments in the area of booking appointments using digital assistants, in the Google I/O conference of 2018 a demo of a system called Duplex AI was presented, it allows one to book appointments and accomplish other tasks by making real phone calls [10]. The technology is still in development and it is not clear if it will be available to the public or developers.

At the time of writing this document, May 2018, a search was conducted on the amazon.com website using "Alexa Skills" as a category, this search returned 24 skills using terms such as: "booking", "booking appointments" or "scheduling appointments". The 24 skills found were installed tested and verified manually. Most of them were focused on booking tables in restaurants or booking other services like taxis and hotel rooms. Two skills from recognized travel companies Expedia [11] and Kayak [12] were found. Both aim to rent vehicles, and reserve hotels and flights using Alexa. Skills associated with companies that have booking potential tend to provide contact information or interesting facts rather than providing the booking functionality themselves. Some examples are: the Delta Dental of Kansas [13] skill and the Novant Health [14] skill, both belonging to healthcare institutions.

None of the skills in the category of healthcare offer the appointment booking functionality in their description. Three skills were found which advertise in their description the functionality of allowing to book appointments. The first skill it is called "Make Appointment", it belongs to an ophthalmological center located in Las Vegas called Enhanced Eye Care [15], the skill allows to sign up for being contacted via email to coordinate an appointment. The second skill is called Pingup [16], it belongs to a company of the same name that provides booking services to other companies, Pingup allows to book an appointment on any company registered in their platform. This is achieved by providing the type of business, the service and the name of the business, however, this data must be known prior to its use. The last skill belongs to the Nimblr platform [17] and goes by the same name, it allows to book medical appointments, and is not very adaptable when it comes to another type of appointment or reservation, also it does not allow integration with existing systems. The Nimblr platform has associated costs for its usage.

The booking appointment skills found in the market were enabled and partially tested, the main goals were: testing the booking appointment functionality and explore the skill structure as an application. This quick assessment of the skills revealed the following issues:

- Multiple types of businesses: branding and monetization of the individual companies in the amazon platform is affected as all the businesses are behind a single skill name and brand.
- Lack of help: the user is required to know the type of businesses before using the skill, they do not provide any kind of help mechanisms when it comes to making choices.
- Incomplete experience: the appointment functionality is delegated to email and turned into a manual process, leaving the voice assistant behind.
- Associated fees for businesses: businesses are charged monthly fees for the booking platform service.
- Lack of control: the business is not in control of the Alexa Skill, therefore, there is a limit on the changes that can be made.
- Security risk: having multiple businesses under the same skill means that companies data is combined to certain degree.
- Proprietary voice models: some skills despite working properly belong to businesses and for that reason they do not disclose their voice models to the public.

3. Appointments Architecture

In this section, the main elements of the architecture are described, and a breakdown of the appointment and dialog structures is provided, some of the advantages offered by the architecture in terms of usability and customization are described, finally, the use cases and the reasons behind some of the design choices are discussed.

3.1. Appointment Structure

An appointment is defined as an agreement between two parts at a given time and place, extending this concept further, appointments can have several additional elements that can be optional depending on the business type. This architecture is focused on appointments between a single person and a single business only, no other variations on this area will be supported by this study. Voice assistants like the Amazon Echo are unable to collect big chunks of information in a single voice interaction, this means that several voice interactions are needed in order to exchange enough information to build a useful information set.

3.1.1. Appointment Elements

Thinking in terms of Object-Oriented Programming and with the previous idea in mind, the appointment object was divided in small pieces, these pieces are going to be called appointment elements from now on. Each appointment element must be collected on a single voice interaction following the principle of simplicity for voice designs. Some elements will require the users to choose from long lists of items, only a few of the items in these lists will be read aloud to the user, doing it for all of them might result in a poor voice experience. Voice interactions must be kept short and simple, Section 3.2 describes the proposed solution to this particular issue. Certain elements can be configured for particular needs, this will be explored in detail in the Appointment Configuration Types section. The following is a description of the valid appointment elements in the proposed architecture.

- Establishment (E): This element identifies the company or business where the appointments take place, this is an immutable element and the root of the architecture, an application must have just one establishment. The company name is usually assigned to this element. This element represents the application itself, meaning that each establishment must have its own application. Handling multiple companies in the same application can be problematic as discussed earlier, the proposed solution to this problem will be discussed further in this section.
- Branch (B): This element identifies the physical office or location where the appointment takes place. It is selected by the user from a list of valid branches provided by the application. The application provides the user with a list of the nearest branches obtained based on the location configured in the smart assistant device. This element cannot be configured using the architecture settings. The Branch element is always required to build an appointment object, even if the business has only one physical location.
- Service (S): This element describes the available services the business has for booking. A single service can be selected by the user from a list of valid services provided by the application, services can be global for the skill or programmed based on their availability on each branch, the developer is in charge of making this decision. This element cannot be configured using the architecture settings and is always required to build an appointment object.
- Assessor (A): Some appointments have a person dedicated to providing the desired service, e.g., the doctor in the case of a medical appointment, the assessor element allows to request and store this person information. This element can be fully configured using the architecture settings and its optional when it comes to build an appointment object.
- Date (D): This element indicates the date when the appointment takes place. The user provides his preference of date for the appointment, the input gets validated against the business rules set by the programmer, if the input is valid the interaction finishes. On invalid input the application provides the customer with available date suggestions. Users are allowed to indicate the date in several ways, e.g., by saying tomorrow, next Monday or November 19th. The logic to support this kind of inputs is usually provided by the voice service. This element can be partially configured using the architecture settings and is always required when it comes to build an appointment object.
- Time (T): Same as the date, this element indicates the time when the appointment takes place. The user provides his preference of time for the appointment and then the input gets validated against the business rules, once again set by the programmer, if the provided input is valid the interaction finishes, otherwise, the application provides the customer with available time suggestions. This element can be fully configured using the architecture settings and is optional when it comes to build an appointment object.

3.1.2. Appointment Configuration Types

As described above, some elements of the appointment object can be configured using the architecture settings. The purpose of this configuration is to provide a flexible structure that can be easily adjusted to different types of businesses. The following is a description of the architecture settings available.

- Given (G): An element is configured as given when its presence is required in the appointment object, but the user is not allowed to select it from a list. When this configuration is used the application is in charge of assigning a value to the element, either by following a business rule or randomly, this will depend on the application setup. The Assessor, Date and Time elements can be configured to be Given.
- Selectable (S): An element is configured as selectable when its presence is required in the appointment object, and the user is allowed to select it from a list. When this configuration is used the user is in charge of assigning a value to the element. The Assessor, Date and Time elements can be configured to be Selectable.

- Not Required (N): An element is configured as not required when its presence is not required in the appointment object. When this configuration is used the application will skip all the mentions to this element. The Assessor and Time elements can be configured to be Not required.

3.1.3. Configuration Combinations

In order to simplify the understanding of the architecture each appointment element and configuration type has been identified with a letter. Establishment (E), Branch (B) and Service (S) are not configurable, this leaves Assessor (A), Date (D) and Time (T) to make valid combinations with the different configuration types: Given (G), Selectable (S), and Not Required (N). The combinations are made in form of triads e.g., DG, TS, AN, for Date Given, Time Selectable and Assessor Not required. A list of 18 possible triads that stand for valid combinations was devised, see Appendix A for reference.

An application can only use one configuration triad at the time, the developer in charge of building the application must identify the type of appointment the business offers and select the most appropriate triad for it. Some examples of the usage of triads are explained at the end of this section.

3.2. Composition through Dialog

Voice experiences can start in different ways, the user should have the freedom to start the interaction by asking for booking an appointment with a particular assessor, in a particular location, with a specific service, or just an appointment without any other parameter. Having the appointment object divided into smaller elements has the benefit of allowing to obtain a valid dataset regardless of the order in which the elements were collected.

Context is important when it comes to voice interactions, a better experience for the user means providing the user with choices based on the choices he already made. If the application has already a value for the branch element, it should be possible to list only services available in that branch, also if the application already has a value for the service, it should be possible to list only branches that offer that service. The proposed architecture supports this feature by allowing developers to return this data from a data repository based on the known elements.

3.3. Data Storage and Business Dashboard

The appointments data should be easily stored and retrieved, a relational database is the initial recommendation, but this can be also replaced by an API or any kind of storage system, this choice depends on the business needs. However, there are three basic functionalities that the storage system must provide in order to have a functional application: (1) Storage and retrieval of elements and appointments data, (2) Fast response time, and (3) Access from the business dashboard.

The establishment or business must have a dashboard to see and manage the upcoming and existing appointments, having a website for this purpose is recommended, once again this choice is delegated to the developer as these features can also be integrated in an existing application or system. Ideally, this dashboard will extract the appointment information from the storage system.

3.4. Addressing Common Mistakes

As it was mentioned in the background, critical issues in some of the existing skills designed for booking appointments were found, the efforts were focused on devising ways to deal with the most critical ones based on the conducted testing. The following is a list of the issues that the proposed architecture aims to cover.

3.4.1. Multiple Types of Businesses

The proposed architecture focuses on creating an individual application for each single business, some of the existing Alexa skills ask the user for the desired type of business, e.g., "would you like beauty or finance?", this might confuse the user because it makes easier to lose track of the business where the user is trying book. This could potentially lead to booking an appointment in an undesired business, and also prevents the proper marketing and monetization of the skill due the mix of businesses involved.

3.4.2. Multimodal Interaction

As a user it is easy to get lost in applications that rely solely on voice, especially when it comes to make a choice, some of the skills tested ask the user to provide an option but fail on providing the available choices to pick from, users do not know what to say and get frustrated by the lack of help. The proposed architecture makes use of the companion app provided by the Amazon Echo called Alexa App, in this smart phone application the user is presented with the available options for every choice available in form of text cards, this serves as a visual guide to the user and avoids the need of reading all the options aloud, this information is also displayed in Amazon Echo devices that have a screen. In some cases, it is not possible to have access to the companion app, for this reason, the option to have some items in the lists identified as starred was included in the proposed architecture. These items will be voiced to the user if no answer is provided after a short time. The starred items are fully configurable in the storage mechanism, they can be random or based on a business rule. e.g., most common services or most popular branches. It is recommended to have no more than five items as starred in the lists to keep the voice interactions short. Having these helpers in place prevents the users from getting lost and allows them to receive output from two sources.

3.4.3. Confirmation and Changes

The user is requested to provide a prompt confirmation for each piece of information that gets collected on each voice interaction, this is done in order to prevent the condition where a user answers a lot of questions just to realize the device has collected the wrong data, there is also a final confirmation for the appointment summary. If the user indicates an item in the summary is incorrect, the device will ask for the specific element on the appointment to be corrected and redirect the user to that interaction.

3.5. Customization Requirements

This model structure aims to be simple and easily customizable for the developers in charge of the application implementation, aside from the basic skill requirements, the following items are the minimum conditions needed to start the development process: (1) A valid configuration triad, (2) An establishment or company name, (3) A list of branches, services, and assessors, each one with their respective starred items, (4) A storage mechanism, and (5) A business dashboard. Branches, Services, and Assessors can go by different names, it is recommended to include synonyms in the voice model, this provides more flexibility in terms of user input.

3.6. Triads Use Cases

Table 1 describes some common use cases for different types of triads. There are more use cases depending on the business type, the architecture aims to be flexible enough to cover most of them. See Appendix A for a full list of triads.

Table 1. Common Use Cases for Triads.

Triad	Use Case
DS, TS, AS	Private dental office where users can book an appointment choosing the date, the time and the dentist.
DS, TS, AG	Car dealership where users can book an appointment for a test drive, choosing the date and time but are given an associative in charge of demoing the vehicle.
DG, TG, AG	Public hospital where users can book an appointment for a particular service but do not get to choose the date, time or Doctor.
DS, TG, AN	Government office where users can book an appointment for a selected date, but the time is given to them and no assessor is required.

D: Date, T: Time, A: Assessor, S: Selectable, G: Given, N: Not Required.

3.7. Features

Appendix B describes a list of the application features desired by the architecture. Most of these features aim to enhance the user experience. The book appointment intent is the most critical piece of functionality for this architecture as it encompasses the logic early described. Figure 1 describes the appointment intent flow and the structure of the appointment object that is being built.

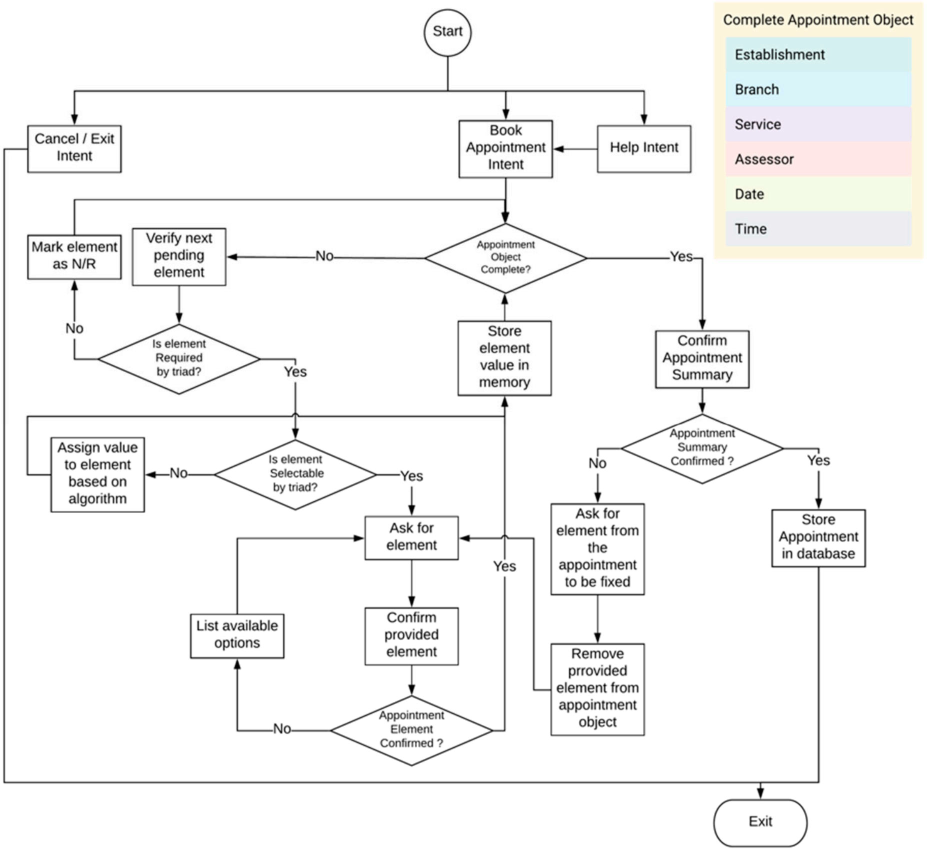
Start
Complete Appointment Object
Establishment
Branch
Service
Assessor
Date
Time
Cancel / Exit Intent
Book Appointment Intent
Help Intent
Mark element as N/R
Verify next pending element
No
Appointment Object Complete?
Yes
No
Is element Required by triad?
Yes
Store element value in memory
Confirm Appointment Summary
Assign value to element based on algorithm
No
Is element Selectable by triad?
Yes
Appointment Summary Confirmed ?
No
Yes
Ask for element
Yes
Ask for element from the appointment to be fixed
Store Appointment in database
List available options
Confirm provided element
Remove prrovided element from appointment object
No
Appointment Element Confirmed ?
Exit

Figure 1. Appointments Architecture Main Structure and Flow.

In the next section, the implementation of the testing prototype and its parts will be described.

4. Architecture Implementation and Configurations

Based on the architecture described in the previous section, an Amazon Echo skill called dental clinic was implemented with the purpose of evaluating its usability. The development of the skill has reached an initial evaluation phase; however, it is worth to mention that this is still a work in progress, for that reason not all the desired features in the architecture were implemented, and more improvements will be added based on the evaluation results.

4.1. Setup

The implemented skill attempts to provide a booking service for a dental clinic using Alexa as a communication channel. This case was selected as it can be easily adjusted to represent different types of appointments during the evaluation process of the skill. The dental clinic has offices in 6 different hospitals, these hospitals served as branches for the skill, a list of 31 services provided by the clinic were included in the skill, also a list of 6 doctors were included to serve as assessors. The lists of branches, services and assessors were included in the voice model and also in the database, also synonyms for all the items were included in the voice model to improve the user experience (see supplementary materials). The branches were set to be open to receive appointments from Monday to Friday from 8 a.m. to 6 p.m. Each appointment was set to have a fixed timespan of 30 min.

4.2. Architecture Overview

The dental clinic skill was developed using the Alexa Skills Kit SDK for Node.js version 2 [18] being this the latest available SDK for Node.js at the time of writing this paper. Node.js version 6.11.0 [19] was selected as development framework mainly due its event-driven, non-blocking I/O model, well-suited for Alexa Skills.

An AWS lambda function is a serverless platform provided by Amazon [20], for this particular skill an AWS lambda function in charge of running backend code in response to events. The code was created using typescript as programming language [21] and deployed to the lambda function. In this particular implementation MySQL is being used as storage mechanism, this database is hosted in an independent server. Like it was mentioned before this implementation can be easily replaced with another database or API.

4.3. Interaction Flow

The user starts the skill by calling one of the invocation phrases supported by Alexa including the skill name and an optional intent [22] e.g., Alexa ask dental clinic to book an appointment.

Voice model utterances are mapped to intent methods. Each time the user speaks, the natural language input is translated to text via the Alexa Skills Kit, if the resulting text matches a valid utterance, the Alexa Skills Kit calls the mapped method for that utterance in the lambda function using an intent request. In the lambda function code, intent methods are defined as middleware functions in the application controllers, there are also services in charge of providing the controllers with external data and helper functions.

Each time an intent method is executed, the control is returned to the Alexa Skills Kit and potentially to the user. The lambda function returns information and behavior commands to the Alexa Skill Kit using the SSML format [23]. The lambda function also communicates with the database server to store the appointments data and retrieve the lists of appointment elements. This database also serves as storage for the appointments dashboard web site. Figure 2 describes the main components of the prototype skill.

At this point of the development process, not all the model features were included in the prototype. Appendix B describes the features included and excluded from the prototype skill.

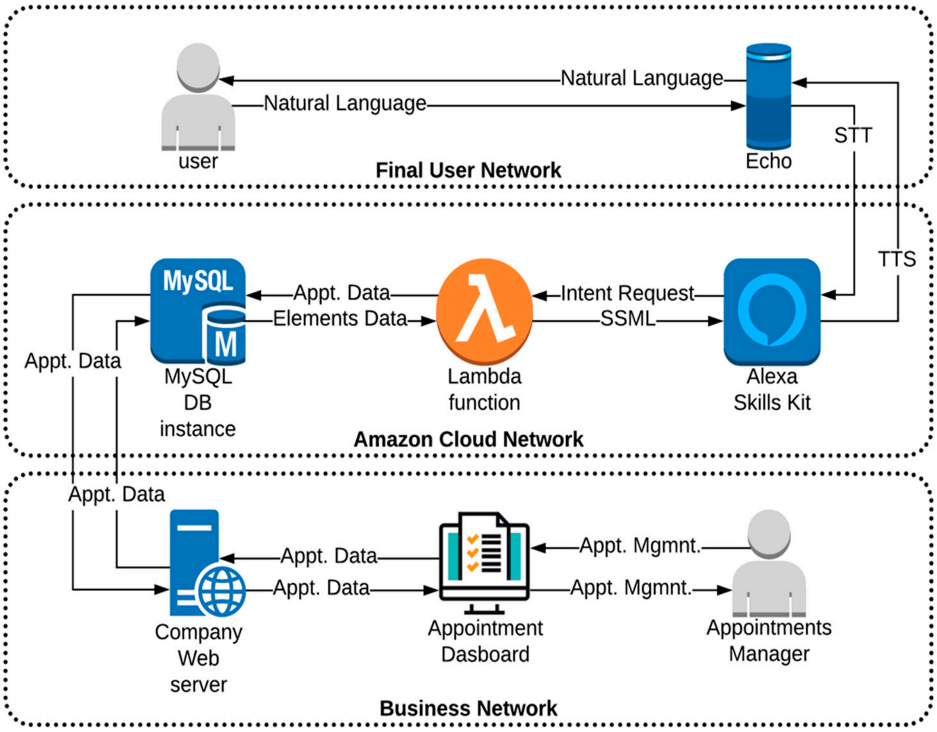

Figure 2. Main components of the prototype skill. TTS: Text-To-Speech, STT: Speech-To-Text.

5. Evaluation Procedure

This section describes the methodology and instrument used to evaluate the prototype skill. An evaluation instrument was created in order to measure the usability of the elaborated prototype when its being used to conduct booking appointment tasks. The instrument uses the SUS system (System Usability Scale) as base adapted for the evaluation of user interfaces by voice, as this system has been proved to be appropriate for evaluating VUIs, offering both convergent and concurrent validity [24]. In addition to the SUS system, open questions were included in the instrument to collect the opinion of the user according to the following elements: positive and negative aspects of the skill, characterization, recommendation of use, emotion, innovation, motivation, level of attractiveness, and global evaluation. The instrument recorded the success or failure in the execution of the tasks performed by the users, the configuration used by each user and the causes of failed tasks, if any.

The prototype skill was evaluated by a group of 24 users selected using the following criteria: being over 18 years old, being of any gender, located in Costa Rica, and having an intermediate or advanced English level. At the time of conducting the evaluation Alexa was not available in Spanish, the language should not be a challenge for the normal use of the skill. The evaluation consists of each user executing two booking tasks using the dental office skill. The first task requested the users to book an appointment using the preferences of their choice, in this case, the users were asked to confirm the appointment reservation in the last step. The second task requested the users to book an appointment using the preferences of their choice, however, in this case, the users were asked to reject the appointment reservation in the last step and then make one or more changes in their appointment, once the changes are made, the users were asked to accept the appointment reservation in the last step. The evaluation was conducted by a single evaluator on private sessions, an Amazon Echo device with a screen was used. Prior to the evaluation, the users were provided with basic instructions on how to use an Alexa device, the purpose of the skill and the two tasks. For each user, the configuration of triads in the skill was changed randomly, this in order to have at least one evaluation for all the reserved booking scenarios supported by the architecture. Once both tasks were completed, the instrument was provided to each user in order to collect their input. The configuration used for each user and the success or failure of each task were also documented in the instrument by the evaluator, an exclusive section in the instrument was devised for this purpose.

6. Results

This section presents the results of the evaluation performed. The results include responses to the System Usability Scale questionnaire and the responses to the open questions. The users age was located between 20 and 40 years old, being 37.5% of them female and 62.5% male.

Most of the users were able to complete both tasks successfully, 83.3% of the users were able to complete the first task, and 75% of the users were able to complete the second task. In terms of failure 75% and 33% of the failures for the first and the second task respectively were caused by an incorrect pronunciation of the voice commands by the users, also 25% and 16.7% of the failures in the first and second task respectively were caused by users providing invalid path commands to the skill, an invalid path command takes Alexa out of the program.

The users were asked what they like the most and the least from the skill, the users liked the most the easiness and speed of the skill, and also not needing to use a laptop or interact with a real person to book the appointment. e.g., *"How easy to learn and simple it was"*, *"How easy and fast it is to use it"*, *"Not having to make a call or turn on the laptop"*. Alexa failing to understand their voice commands and the complexity of providing a date using their voice only were the things users liked the least. e.g., *"It did not catch my accent"*, *"I was confused by the date format expected by Alexa"*. The users were asked to characterize the skill using three words, Figure 3 presents a cloud of the words used by the users. The most relevant words found were easy, useful and fast.

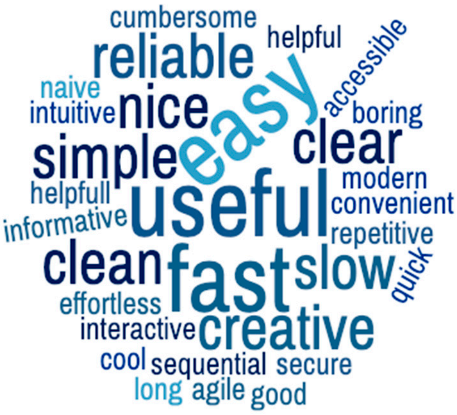

Figure 3. User characterization word cloud.

The responses to the System Usability Scale questionnaire were overall positive with an average SUS score of 80. Users strongly agree on how quickly the usage of the skill can be learned, most of the users agree on the proper integration of the of the skill functions.

Most of the users also think that the skill was easy to use and believe they will use it with frequency. Despite having an agree value of only 33.33% the complexity of some areas remains to be the main issue found by the users. Figure 4 shows the complete results of this survey.

In addition to the SUS questions the users were asked about their perception of the skill in terms of attractiveness, motivation, innovation and excitement.

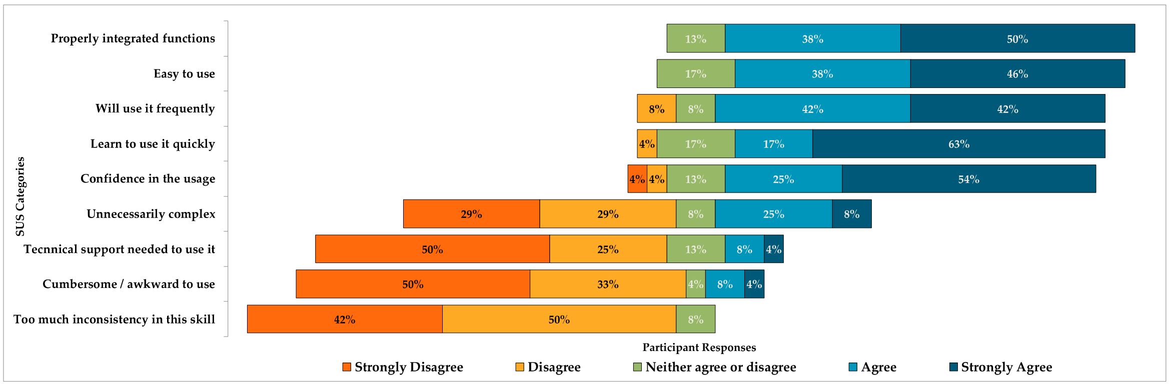

Figure 4. Skill prototype SUS Evaluation Results.

Most of the users have a good perception of the skill, in particular, the skill was found by the users innovative and attractive, Figure 5 shows the collected results. The users were asked to evaluate the skill in general terms using a scale from 1 to 10, being 10 the best possible score 83.3% of the users ranked the skill between 8 and 10. Finally, the users were consulted if they would recommend the skill to others, 91.7% of the users answered yes on this question.

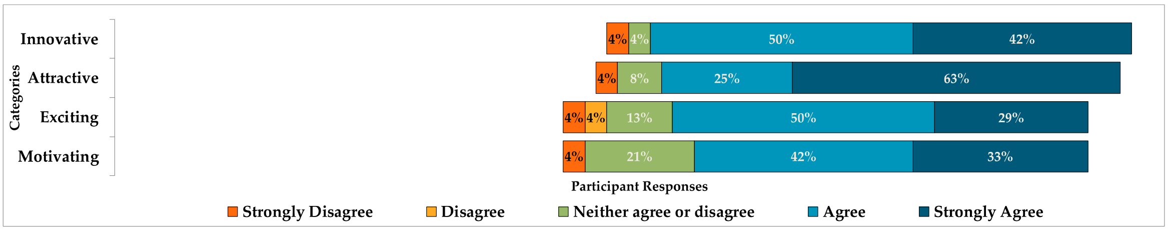

Figure 5. Skill prototype User Perception Results.

7. Conclusions and Future Work

Few Alexa skills provide the booking appointment functionality, the ones that do rely on a third party to achieve it, creating such skills from scratch pose a challenge for new developers. We created an architecture that serves as base to create booking appointment skills in several business scenarios. A prototype skill was created following the architecture principles and eventually evaluated by a group of people using an instrument based on the System Usability Scale.

The characterization and perception results found allowed us to conclude that the system was positively received by the users and most of the users found it easy, useful and fast. Users believe the application can be easily part of their daily lives and could potentially help them to save time.

The SUS results revealed a high score in terms of properly integrated functions, being the most liked one the fact that appointments get properly booked and no further action is required from the user's part. The user confidence and easiness of use got also a high score in the SUS evaluation, we associate this result to the constant guidance included all over the application, users rarely do not know what to say. Despite that most users find the skill easy to use, 33.33% agree the application is unnecessary complex, further exploration on the user's comments shed some light in this result. A group of users experienced a barrier trying to indicate the appointment's date, no help was provided on how the format for the expected date should be provided. Moreover, the date field is interpreted by Amazon and there is little to no control on its interpretation, we associate the complexity score with the complains about the date input.

The results of the first task are a clear indicator of how easy was for users to pick up the skill for the first time with no previous experience.

The results of the second task indicate that most of the users were able to sort the roadblock of making changes in the appointment data. The fact that the second task was also the second time the users hear the instructions, serves as an example of the skill intuitiveness.

Innovation and attractiveness got the highest score in terms of perception, according to the comments people see the smart assistants as a novelty and this creates an interest in having more useful solutions available in such devices.

The skill got high scores in terms of usage frequency (83.34%), recommendation (91.7%) and overall score (8.20/10), we conclude the architecture serves as good base for creating usable booking skills that can be adapted to different business requirements.

It is the expectation of this work to encourage developers into creating not only better voice experiences for the Amazon Echo, but also applications that help connecting businesses with their customers in a more natural way.

There is plenty of work ahead in the area of smart assistants, future work for this project involves an evaluation of the business experience during the process of booking appointments using the prototype skill and improving the VUI dialogs and responses. An additional evaluation of the architecture with a group of developers could provide some insight on how much time can be saved by using it. Furthermore, at this point, the architecture is generic enough to be implemented in other VUIs that follow the concept of utterances, intents, and sessions. The architecture could be adapted to be used in other smart assistants and also modified to be used in different languages for the Amazon Echo. Finally, the model could be extended to work for other business models.

Supplementary Materials: The following are available online: Voice Model: https://s3.amazonaws.com/cdn.andresmeza.com/Alexa/AppointmentsModel/en-US.json, Boilerplate Code: https://github.com/ameza/alexa-appointments-blueprint.

Author Contributions: Conceptualization, A.M. and G.L.; Methodology, G.L.; Software, A.M.; Validation, L.Q., and L.A.G.; Data Curation, A.M.; Writing-Original Draft Preparation, A.M.; Writing-Review & Editing, G.L. and L.Q.; Supervision, G.L. and L.A.G.

Funding: This work was funded by Centro de Investigaciones en Tecnologías de la Información y Comunicación (CITIC), Escuela de Ciencias de la Computación e Informática (ECCI) both at Universidad de Costa Rica. Grant No. 834-B6-178. And the APC received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Valid Triads.

Triad
DS, TS, AS
DS, TS, AG
DS, TS, AN
DS, TG, AS
DS, TG, AG
DS, TG, AN
DS, TN, AS
DS, TN, AG
DS, TN, AN
DG, TS, AS
DG, TS, AG
DG, TS, AN
DG, TG, AS
DG, TG, AG
DG, TG, AN
DG, TN, AS
DG, TN, AG
DG, TN, AN

D: Date, T: Time, A: Assessor, S: Selectable, G: Given, N: Not Required.

Appendix B

Model Desired Features

Features included in the prototype skill

- Booking of appointments selecting location, service, assessor, date and time
- Configurable skill to support all the triads described in the model
- Promptly confirmation of elements on each voice interaction
- Functionality to make changes on individual elements before confirming the appointment summary
- Recommendation of starred items on user silences
- Displaying of available options in the Alexa companion App
- Elements data pulled from a MySQL database or alternative storage mechanism
- Synonyms on the voice models for the assessor names, and locations
- Support for different types of date and time input, (incomplete formats are not supported yet. e.g., next week, afternoon)
- Dashboard website to manage the appointments
- Requesting the list of available options at any given time
- Notification via email with summary of appointment.

Features not included in the prototype skill

- Requesting the nearest available date for an appointment
- Requesting the nearest available time in a given day
- Requesting the most popular office for a given service
- Requesting next available date for a given service
- Appointment cancelation using voice
- Appointment conflicts in terms of date and time
- Valid recommendations in terms of date and time
- Requesting help based on context at any given time
- Using the device location for branches

References

1. Wobcke, W.; Nguyen, A.; Ho, V.; Krzywicki, A. The Smart Personal Assistant : An Overview. *Scenario* **2007**, 2005–2006.
2. Amazon Team Amazon.com Help: Set Up Your Echo. Available online: https://www.amazon.com/gp/help/customer/display.html?nodeId=201601770 (accessed on 22 May 2018).
3. Alexa Skills Kit—Build for Voice with Amazon. Available online: https://developer.amazon.com/alexa-skills-kit (accessed on 22 May 2018).
4. Alexa Voice Service Overview (v20160207) | Alexa Voice Service. Available online: https://developer.amazon.com/docs/alexa-voice-service/api-overview.html (accessed on 16 December 2018).
5. Amazon.de Amazon Alexa Voice Design Guide. Available online: https://developer.amazon.com/designing-for-voice/design-process/ (accessed on 14 May 2018).
6. Create Intents, Utterances, and Slots | Custom Skills. Available online: https://developer.amazon.com/docs/custom-skills/create-intents-utterances-and-slots.html (accessed on 22 May 2018).
7. Ghosh, S.; Pherwani, J. Designing of a natural voice assistants for mobile through user centered design approach. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer, Cham, 2015; Volume 9169, pp. 320–331.
8. Abbott, K. VUI Design Principles and Techniques. In *Voice Enabling Web Applications: VoiceXML and Beyond*; Apress: Berkeley, CA, USA, 2002; pp. 87–103, ISBN 978-1-4302-0850-1.
9. Alexa Blueprints. Available online: https://blueprints.amazon.com/help (accessed on 14 May 2018).
10. Google AI Blog: Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. Available online: https://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html (accessed on 23 May 2018).
11. Amazon.com: Expedia: Alexa Skills. Available online: https://www.amazon.com/Expedia-Inc/dp/B01M7TT1O8 (accessed on 23 May 2018).
12. Book Hotels, Track Flights, Explore &amp; Search with Amazon Alexa + KAYAK. Available online: https://www.kayak.com/alexa (accessed on 22 May 2018).
13. Amazon.com: Delta Dental of Kansas: Alexa Skills. Available online: https://www.amazon.com/Delta-Dental-of-Kansas/dp/B07BDRKK5K/ (accessed on 23 May 2018).
14. Amazon.com: Novant Health: Alexa Skills. Available online: https://www.amazon.com/Novant-Health/dp/B0773HPGMX (accessed on 23 May 2018).
15. Amazon.com: Make Appointment: Alexa Skills. Available online: https://www.amazon.com/TsaTsaTzu-Make-Appointment/dp/B073F1541N/ref=sr_1_1?s=digital-skills&ie=UTF8&qid=1527066184&sr=1-1&keywords=book+appointment (accessed on 23 May 2018).
16. Pingup | Be Bookable. Anytime. Everywhere. Available online: https://pingup.com/ (accessed on 23 May 2018).
17. Nimblr | Much more than appointments, reminders and scheduling. Available online: https://nimblr.ai/home (accessed on 27 November 2018).
18. Alexa Skills Kit SDK for Node.js | SMAPI. Available online: https://developer.amazon.com/docs/sdk/alexa-skills-kit-sdk-for-nodejs.html (accessed on 22 May 2018).
19. About | Node.js. Available online: https://nodejs.org/en/about/ (accessed on 25 November 2017).
20. AWS Lambda—Serverless Compute—Amazon Web Services. Available online: https://aws.amazon.com/lambda/ (accessed on 22 May 2018).
21. TypeScript—JavaScript that scales. Available online: https://www.typescriptlang.org/ (accessed on May 22, 2018).
22. Understanding How Users Invoke Custom Skills | Custom Skills. Available online: https://developer.amazon.com/docs/custom-skills/understanding-how-users-invoke-custom-skills.html (accessed on 22 May 2018).

23. W3C Speech Synthesis Markup Language (SSML) Version 1.1. Available online: https://www.w3.org/TR/speech-synthesis11/ (accessed on 22 May 2018).
24. Ghosh, D.; Foong, P. S.; Zhang, S.; Zhao, S. Assessing the Utility of the System Usability Scale for Evaluating Voice-based User Interfaces. In Proceedings of the Sixth International Symposium of Chinese CHI on—ChineseCHI '18, Montreal, QC, Canada, 21–22 April 2018; ACM Press: New York, New York, USA, 2018; pp. 11–15.