Enhance customer experience with Conversational Interfaces

Tara E. Walker
Sr. Technical Evangelist
Amazon Web Services
@taraw
Agenda

✓ The What & Why of Conversational Interfaces
✓ Ins and Outs of Amazon Lex
✓ Putting Conversational Interfaces to Work
What & Why of Conversational Interfaces
“What we need now is to be able to simply talk with our devices. That's why I believe it's finally time for the conversational user interface, or "CUI."

This is the interface of the future, made even more necessary as computing propagates beyond laptops, tablets and smartphones to cars, thermostats, home appliances, and now even watches … and glasses.”

~ Ron Kaplan (via WIRED magazine), Lead-Nuance Communications' NLU R&D Lab, Professor of Linguistics at Stanford University, former CTO of Powerset
Why Conversational Interface Access

- Natural
- On-demand
- Accessible
- Efficient
Developer challenges

Conversational interfaces need to combine a large number of sophisticated algorithms and technologies.
Considerations when building Conversation Interfaces

• Understand the conversation flow
• What information do you need from the conversation
• What is the intent/goal of the conversation
• Validate your input
• Handle errors gracefully
• Add clarification prompts if required
• Test, test and more testing
• Understand Metrics after your bot or conversational interface is deployed in production
“Alexa, What Is Amazon Lex?”
Amazon Lex - Overview

Text and speech language understanding: powered by the same technology as Alexa

Build once and deploy to multiple platforms

Designed for builders: efficient and intuitive tools to build conversations; Scales automatically

Enterprise Ready: connect to enterprise systems via SaaS connectors; Versioning and alias support

Continuous Learning: monitor and improve your bot
Hello Amazon Lex

Built-In Integration with Slack, Messenger, & Twilio
Text and speech language understanding

Speech recognition

Natural language understanding

Powered by the same deep learning technology as Alexa
Amazon Lex – Multi-platform

Mobile
SDKs: iOS & Android Mobile Hub

Messaging Platforms
Facebook, Twilio SMS and Slack

Web
SDKs: Java, JavaScript, Python, CLI, .NET, Ruby on Rails, PHP, Go

IoT
Integrated with AWS IoT via AWS Lambda

Build once and deploy to multiple platforms
Amazon Lex – Technology

AWS Services
- Amazon Cognito
- CloudTrail
- CloudWatch

Input: Speech or Text
- Multi-Platform Clients: Mobile, IoT, Web, Chat
- SDK
- Console
- API

End Users

Output: Speech (via Amazon Polly TTS) or Text

Authentication & Monitoring

Action
- AWS Lambda

Fulfillment

Amazon Lex
- Automatic Speech Recognition (ASR)
- Natural Language Understanding (NLU)

Same technology that powers Alexa

Intents, Slots, Prompts, Utterances

Speech API

Language API

Multi-Platform Clients: Mobile, IoT, Web, Chat

SDK

End Users

Developers

Console

API

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Amazon Lex – Key Concepts

**Intents**
An intent performs an action in response to natural language user input.

**Utterances**
Spoken or typed phrases that invoke your intent.

**Slots**
Slots are input data required to fulfill the intent.

**Fulfillment**
Fulfillment mechanism for your intent.

---

**BookHotel**

- I’d like to book a hotel
- Sure, which city?
- New York City
- What date do you check in?
- November 30th

- Are you sure you want to book the hotel in New York City?
- Yes.
- Thank you. The reservation went through successfully.
Amazon Lex – Example: “Book a hotel”

“Book a Hotel in Stockholm”

Automatic Speech Recognition

Hotel Booking
- City: Stockholm
- Check In: May 14th
- Check Out: May 16th

Natural Language Understanding

Intent/Slot Model

“Can I go ahead with the booking?"

Confirmation: “Your hotel is booked for May 14th”

Amazon Polly

“Your hotel is booked for May 14th”
Amazon Lex – Slot elicitation

I would like to book a hotel

Sure, which city will you be traveling to?

Stockholm

What date do you want to check in?

May 14th

City Stockholm

Check In 5/14/2018
Amazon Lex – Dialog management

I would like to book a hotel

Sure, which city will you be traveling to?
Stockholm

What date do you want to check in?
Tomorrow

... And for how many nights is this for?

Simple Declarative Model

<table>
<thead>
<tr>
<th>Slots</th>
<th>Prompts</th>
</tr>
</thead>
<tbody>
<tr>
<td>City</td>
<td>Which city will you be traveling to?</td>
</tr>
<tr>
<td>Check-In Date</td>
<td>What date do you want to check in?</td>
</tr>
<tr>
<td>Check-Out Date</td>
<td>How many nights is this for?</td>
</tr>
</tbody>
</table>

Build Multi-turn Conversations

Easy Setup in Console

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
I would like to book a hotel

Would you prefer to stay in Downtown this time as well?

Yes

What date do you want to check in?

May 14th

Sorry no availability. Would a different location work?

Personalize conversation based on user preferences

Validate user input and re-prompt as necessary
Amazon Lex – Error handling

- I want to .... {garbled} .....  
  - Sorry can you please repeat that?

- I am having trouble understanding  
  Can you please say that again?

- Sorry I am not able to assist you at this time

- Clarify by requesting user to repeat
- Uses a different prompt every time
- Hang up phrase to end the conversation
Lex maintains context by storing data throughout the conversation.
Amazon Lex – Dynamic conversation flow

**First Intent**
- **Anything else today?**
- **Book a Car**

**Second Intent**
- **Dine In or Take out?**
- **Dine In**
- **Takeout**

**Switch Intents**
- **Session Attributes**

**Chain Intents**
- **Session Attributes**
Amazon Lex – Versioning and Alias support

**Versioning**
- Supported for intents, slots, and bots
- Enables multideveloper environment
- Rollback to previous versions

**Alias**
- Deploy different aliases to different platforms
- Run different stacks for dev, stage and prod environments
- Target different user groups with different aliases
Amazon Lex – Fulfillment & Response

Intents and slots passed to **AWS Lambda** function for business logic implementation.

User input parsed to derive intents and slot values. Output returned to client for further processing.

**AWS Lambda integration**

**Return to Client**
Amazon Lex – Deployment Cycle

Save

Saving your bot preserves the current state on the server

Build

Building your bot creates versions that you can test

Publish

Publishing your app will create a version of your bot and provide an alias to your clients

Test

Test your bot in a chat window on the console
Amazon Lex – Export to Alexa Skill

Amazon Lex

Export to Alexa Skills Format

Chatbot schema (JSON)

Import into Alexa Skill Builder

Alexa Skill
Amazon Lex

Demo
Thank You!

Tara E. Walker
Sr. Technical Evangelist
Amazon Web Services
@taraw