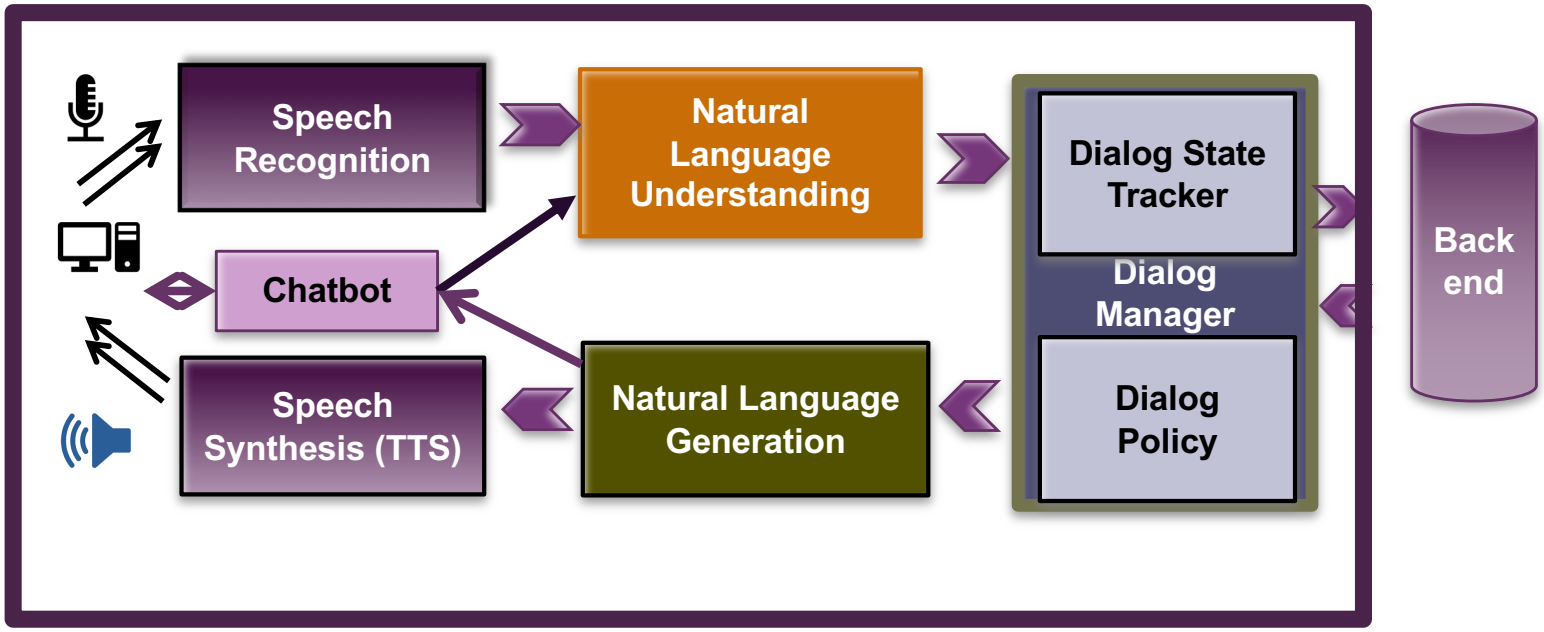
From Speech to Task oriented Dialog

Dialogflow tool

CS136a
Marie Meteer
Oct 26, 2018

+ Dialogue System Architecture



+ NLU Dialog Tools

- Main toolsets
  - Dialogflow
  - Alexa
  - Watson Conversation
  - Microsoft (LUIS + Azure)
- All handle NLU
  - Text → intent + entities
    - Everything not captured in these two is ignored
- Some offer some dialog management
  - “Frame-based dialog management”
    - Will ask for required entities
  - Context
    - Gating what can be the next interpretation
    - Saving some information for future interpretation
- Fulfillment and response built into the intent definition

+ DialogFLow

- Toolset for designing Voice Assistants
- Use Dialogflow to create an “Agent”
- “Fulfillment” refers to the backend application(s) that the user can interact with
- Key problem: How to interpret what the user says in order to know what fulfillment action to take?
- Assumption: The interaction is goal driven and the users know what they want to do

+ Dialogflow: the human and the machine

Welcome

Bill's friend Harry wants to ask him a question. So as not to be rude, Harry says "Hello" to Bill first.

Invocation

In order to start a conversation with an agent, the user needs to invoke the agent. A user does this by asking to speak with the agent in a manner specified by the agent's developer.

+ Dialogflow: the human and the machine

Request

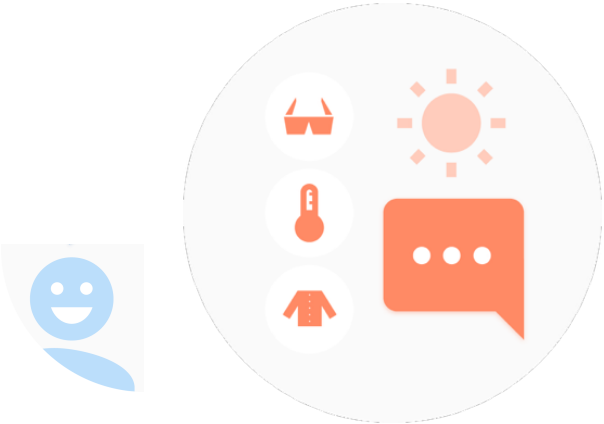
Harry asks Bill "What's the weather supposed to be like in San Francisco tomorrow?" Because Bill is familiar with the city and the concept of weather, he knows what Harry is asking for.

Intent

A user asks the agent "What's the weather supposed to be like in San Francisco tomorrow?" In Dialogflow, an intent houses elements and logic to parse information from the user and answer their requests.

+ Dialogflow: the human and the machine

Language Learning



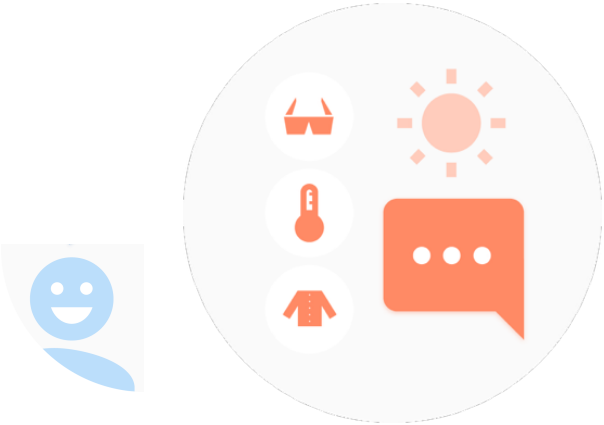
- People learn the meaning of language throughout their lives, but mostly when they're little.

Training Phrases

- For the agent to understand the question, it needs examples of how the same question can be asked in different ways. Developers add these permutations to the Training Phrases section of the intent. The more variations added to the intent, the better the agent will comprehend the user.

+ Dialogflow: the human and the machine

Language Learning



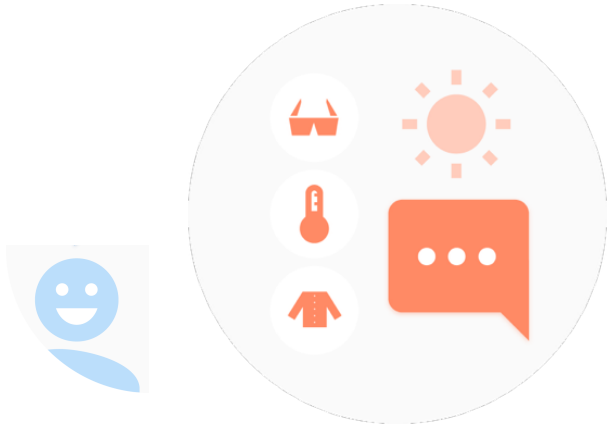
- People learn the meaning of language throughout their lives, but mostly when they're little.

Training Phrases

- The Dialog/ow agent needs to know what information is useful for answering the user's request. These pieces of data are called entities. Entities like time, date, and numbers are covered by system entities. Other entities, like weather conditions or seasonal clothing, need to be defined by the developer so they can be recognized as an important part of the question.

+ Dialogflow: the human and the machine

Fulfillment



- Armed with the information Bill needs, he searches for the answer using his favorite weather provider. He enters the location and time to get the results he needs.

Fulfillment request

- Dialogflow sends this information to your webhook, which subsequently fetches the data needed (per your development). Your webhook parses that data, determines how it would like to respond, and sends it back to Dialogflow

+ Dialogflow: the human and the machine

Response

- After scanning the page for the relevant info, Bill tells Harry "It looks like it's going to be 65 and overcast tomorrow."

Response

- With the formatted reply "in hand", Dialogflow delivers the response to your user. "It looks like it's going to be 65 and overcast tomorrow."

+ Dialogflow: the human and the machine

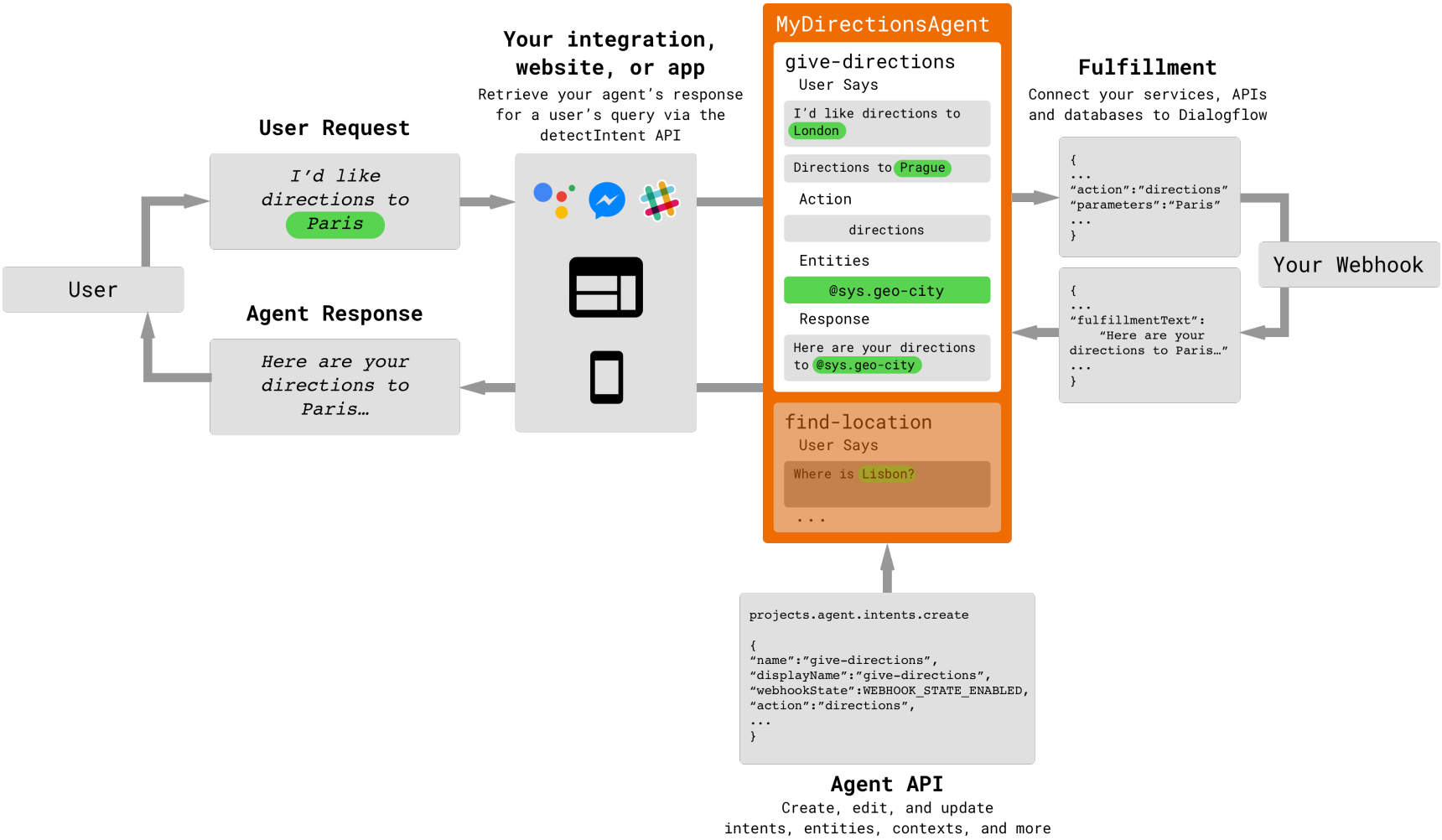
Context

- Now that the conversation is on the topic of weather, Bill won't be thrown off if Harry asks "How about the day after that?" Because Harry had asked about San Francisco, follow up questions will more than likely be about the same city, unless Harry specifies a new one.

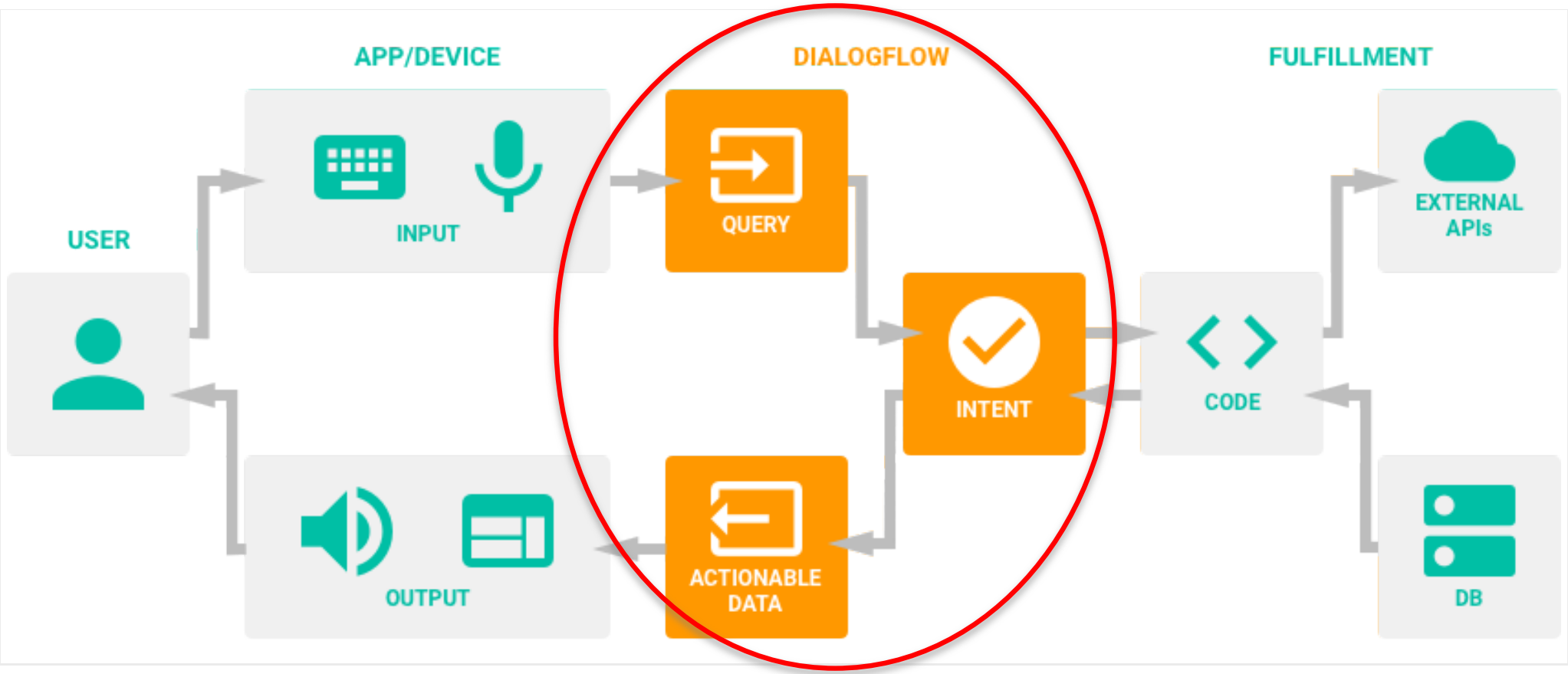
Context

- Similar to Bill's scenario, context can be used to keep parameter values, from one intent to another. Contexts are also used to repair a conversation that has been broken by a user or system error, as well as branch conversations to different intents, depending on the user's response.

Overall System Architecture



+ Agents: NLU modules



+ Top level

- Agent
  - Transform user input into fulfillment requests/actions
  - Select appropriate response
- Custom agents
  - Set of defined intents and entities
- Prebuilt customizable “specialists”
  - Alarm, Date, App management, Navigation
  - Banking, Car, Coffee shop, Flights
- Application
  - Combination of agents

+ Agent

- Embodiment of the application face to the user
- Set the language, time zone
- It provides a project ID, service account
- Machine learning settings
- Import/expo
- Create a group
- Example: MyPizzaPalace

Intents

- Mapping between what a user says and an action by the underlying software
- Parts:
  - User says
  - Action
  - Response
  - Contexts

+ Entities

- Parameter values in natural language inputs
  - The “things” they user referred to
- System
  - Entities with reference values, e.g. dates and currencies
- Developer
  - Entities specific to your application, e.g. pizza toppings
- User
  - Entities specific to a user, such as songs on the playlist

+ Entities

- Entities are used for extracting parameter values from natural language inputs.
- 3 Types
  - System: built in by Dialogflow
  - Developer: added by you
  - User: defined on a session ID level, e.g. for user's playlists
- Mappings
  - Reference values: Today's date would map to 2018-06-14
  - Enum: No reference value (names, phone numbers)
  - Composite: Containing other entities with aliases & returning object type values, e.g. "Clothing" type might have size and color

Other features of entities

- ”is list”
  - For when there can be multiple values of that type in a list, e.g. Toppings.
  - This gets automatically checked if you give an example with a list
    - Order a ”pizza with mushrooms and peppers
- Automated expansion
  - Will allow entities values to be found that are not predefined
  - For example if you say “put **avocados** on my shopping list” and **avocados** is not defined as an ”item” value, it will be automatically added
  - Be careful with this! If the sentence is ambiguous, you may get additions you don’t want
- Uploading sets of entities
  - Json or csv format
  - csv: “reference value”, ”item”, ”synonym1”,”synonym2”, …
    - "New York City", "New York City", "NYC", "New York City, USA”
- Sharing between agents
  - Can copy or move entities between agents

+ Composite entities

- Connecting pieces of information
  - 2 pizzas
  - Small red t-shirt
  - Under $100,000
- Define basic types
  - @item: pizza, soda
  - @size; small, medium
- Define an entity type “order-item” @type:value
  - Rather than words as values, use types

    @sys.number:number @item:item

    A @item:item

    @sys.number”number @size:size @item:item

+ Annotating Intents and Entities

- Create Intents
  - What actions/questions do you want your assistant to be able to do
- Create entities types
  - What parameters do these actions take?
- Creating data
  - In the Intents interface type in a sentence and highlight each parameter value and assign it to an entity type

How does it work?

- Machine learning
  - Both from your examples and
  - from language models in the system
  - Continually update from logs
- “Match Mode”
  - Hybrid: Rule based on ML when there is a small number of intents, lots of templates or composite entities
  - ML only: Agents with large number of examples and lots of @sys parameters.
- Training tool, lets you annotate conversations with the agents

Key is examples!

- Design
  - Personas → Task space → Scenarios → Examples → Entities & Intents

- Some one-shot pizza orders

  Hi I'd like a large pepperoni pizza (Amy_1)
  Hi I'd like a large pizza with pepperoni (Amy_2)
  Hi I want a small pizza with sausage and tomatoes (Chi_Zhang_007)
  May I have a large Barbeque chicken pizza with pineapples please (Chi_Zhang_008)
  Give me a large double cheese pizza with mushroom please (Chi_Zhang_010)
  I'd like to order a large cheese pizza (Dewar_001)
  Hello I would like to order a large pepperoni and olive pizza (Dewar_002)
  Hi Please give me a large spinach and feta pizza (Dewar_003)
  Can I get a small pizza with sausage and onions for takeout (Dewar_004)
  Could I get a medium Hawaiian pizza (Dewar_005)
  I would like a hot sauce and ham pizza for delivery (Dewar_006)
  Can I get a Memphis BBQ chicken pizza (Dewar_007)
  I want a medium pizza with Banana Peppers and spinach (Dewar_008)

+ Contexts

- Allows inputs to be interpreted differently based on context, such as user preferences, location, topic, …
  - “Turn off” will default to something that is currently playing or has been recently turned on
  - Note this does not appear to include things that were just said
    - ??Can it do “What’s the weather in Boston?” “How about LA?”

+ Topics

- Dialog state
- Understanding the complexities in your dialogs
- Competitive analysis
- From ideas to wireframes
  - Follow the axioms

Where is the “state”

- Fluent conversations require a representation of the context
- 3 ways to capture context
  - Within dialogflow itself through the context variable
    - This is passed to the server each time and can hold previously acquired information
  - In the server.js program.
    - Once you start it up, global variables stay set until it is shut down
    - Note when there are multiple users, this needs to be a complex data structure, such as a hash indexed by the session ID
  - Externally on a server
    - In a json file that is read in, adjusted, and saved out each turn.
    - In a database

What is in the state?

- Differentiate long term user model with current state of the dialog
- User model
  - Log in credentials are long term
  - User profile information
- Current state:
  - Whether the user is logged in is in the current state
  - What has been mentioned so far

+ Dialog Sequencing

- Linear: Collect info to complete a required action (aka Frame)
  - Example: Book a hotel
  - Multiple parameters need to be collected in order to successfully execute.
  - Doesn't always come in the first request
  - Action has multiple parameters. Define prompts to get necessary info
- Non-linear: Several branches depending on users answer
  - Example: Survey which determines follow up questions based on answers

From understanding to action

- Intents and entities are about the words
  - How actions are expressed, how things in the world are referred to
- Actions & parameters are about communicating with the application and response generator
  - Training phrases has an action & parameter table for each example
  - Action & parameter table defines the parameters for each action globally

But for now, let’s just chat!

- Integrations: Choose Slack