CC BY NC SA

Voice User Interface On The Web

Human Computer Interaction

Fulvio Corno, Luigi De Russis

Academic Year 2019/2020

How to create a VUI on the Web?

- Three (main) steps, typically:
  - Speech Recognition
  - Text manipulation (e.g., Natural Language Processing)
  - Speech Synthesis
- We are going to start from a simple application to reach a quite complex scenario
  - by using HTML5, JS, and PHP
- Reminder: we are interested in creating an **interactive prototype**, at the end

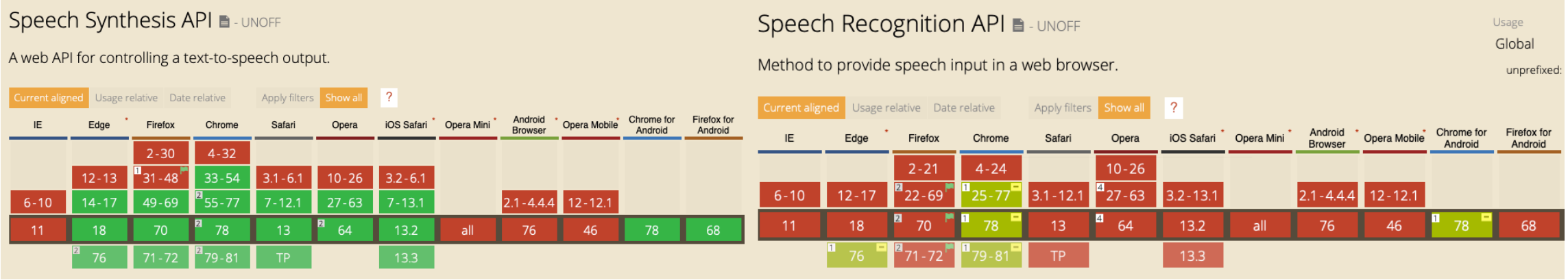
Weather Web App

A VUI for "chatting" about the weather

Base implementation at
https://github.com/polito-hci-2019/vui-example

Speech Recognition and Synthesis

- Web Speech API
  - currently a draft, experimental, unofficial HTML5 API (!)
  - https://wicg.github.io/speech-api/
- Covers both speech recognition and synthesis
  - different degrees of support by browsers



Web Speech API: Speech Recognition

- Accessed via the **SpeechRecognition** interface
  - provides the ability to recognize voice from an audio input
  - normally via the device's default speech recognition service
- Generally, the interface's constructor is used to create a new **SpeechRecognition** object
- The **SpeechGrammar** interface can be used to represent a particular set of grammar that your app should recognize
  - Grammar is defined using JSpeech Grammar Format (JSGF)

Speech Recognition: A Minimal Example

```
const recognition = new window.SpeechRecognition();
recognition.onresult = (event) => {
       const speechToText = event.results[0][0].transcript;
}
recognition.start();
```

- This will ask the user to allow the page to have access to the microphone
- Then, the user can start talking and when she/he stops, the `onresult` event handle will be fired, making the results of the speech capture available as a JavaScript object
- The `onresult` event handler returns a `SpeechRecognitionEvent` with a property results which is a two-dimensional array
- The first object of this matrix is the transcript, i.e., the recognized speech in text format

https://developer.mozilla.org/en-US/docs/Web/API/SpeechRecognition

Web Speech API: Speech Synthesis

- Accessed via the **SpeechSynthesis** interface
  - a text-to-speech component that allows web applications to read out their textual content
  - normally via the device's default speech synthesizer
- Different voice types are represented by **SpeechSynthesisVoice** objects
- Different parts of text to be spoken are represented by

  SpeechSynthesisUtterance objects
  - you can get these utterances spoken with the SpeechSynthesis.**speak()** method

Speech Synthesis: A Minimal Example

```
var synth = window.speechSynthesis;
var utterThis = new SpeechSynthesisUtterance("say this!");
synth.speak(utterThis);
```

- This will create some utterances from a given text
- … and reproduce the text (with a default voice, personalizable) through the device's speakers

https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis

Weather Web App

Let's code: integrating the Web Speech APIs…

Conversational Platforms

- Natural language understanding platforms
  - for developers, mainly
  - typically cloud-based
- To design and integrate voice user interfaces into mobile apps, web applications, devices, …
- Focus on simplicity and abstraction
  - no knowledge of NLP required

Conversational Platforms

- Two main families:
  1. Extension of a product
     - they need an existing product (software and/or hardware) to work
     - e.g., Actions on Google or Skills for Amazon Echo
  2. Standalone services
     - a series of facilities to create a wide range of conversational interfaces in one platform, *typically* integrated in "suites" of cloud services
     - e.g., Dialogflow, IBM Watson, wit.ai, …

wit.ai

- "Natural Language for Developers"
  - https://wit.ai/
- California-based startup, founded in 2013, acquired by Facebook in 2015
- Free to use
- Allow the creation and validations of commands, i.e., "annotated" sentences
- Multiple languages support
  - English, Dutch, Italian, Chinese, …
- Four SDKs
  - Node.js, Python, Ruby, and Go

Snips

snips

- "Create a Private by Design voice assistant that runs on the edge"
  - https://snips.ai
- France-based startup, founded in 2013, acquired by Sonos in 2019
- Run on the edge, not in the cloud
  - Raspbian, Android, iOS, macOS, and most Linux flavors
  - the setup of the NLP component is online
- Free for makers and for building prototypes
- 6 fully supported languages, mostly uses Node.js

DialogFlow

- "Build natural and rich conversational experiences"
  - https://dialogflow.com
- California-based startup, founded in 2010, acquired by Google in 2016
  - previously known as api.ai
- Free to use for simple usage
- One-click integration with several services
  - Telegram, Facebook Messenger, Cortana, Google Assistant, …
- Multiple languages support
  - English, Dutch, Italian, Chinese, …
- REST API and various (official) SDKs
  - Java, C#, Python, PHP, Go, and Node.js

DialogFlow: Definitions

- Each application (an agent) will have different **entities** and **intents**
- Intent
  - a mapping between what a user says and what action should be taken by the agent
- Typically, an intent is composed by:
  - What a user says
  - An action
  - A response
- Different out-of-the-box intents can be enabled on DialogFlow

DialogFlow: Definitions

- Entities
  - represent *concepts*
  - serve for extracting parameter values from natural language inputs
  - should be created only for concepts that require actionable data
- Many pre-existing entities are available on the platform

Weather Web App

Let's code: setting up a DialogFlow agent and integrating it in code…

DialogFlow: Fulfillment

- Often, you need more than "static" responses
  - you can use **fulfillment** to connect an external service to your Dialogflow agent
  - for example, if I want to know the weather in Turin, I need to get information from a real weather forecast service
- *Each* intent has a setting to enable fulfillment
  - if an intent without fulfillment enabled is matched, Dialogflow uses the static response you defined for the intent
  - otherwise, it will call the external service via **Webhook**

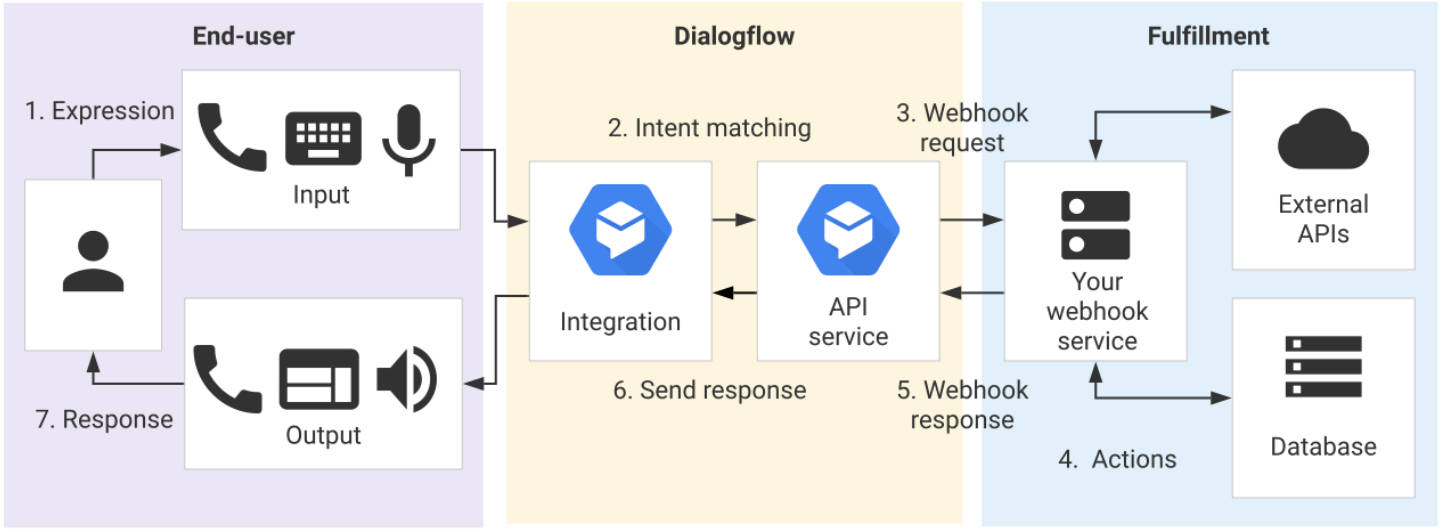
Webhook

- An HTTP callback
- An HTTP POST that occurs when something happens
  - a simple event-notification via HTTP POST
- A web application implementing Webhooks will POST a message to a URL when certain things happen
- They are a way to receive valuable information *when it happens*
  - rather than continually polling for that data and receiving nothing valuable most of the time

Webhook integration in Dialogflow

- A method to pass information
  - from a matched intent
  - into a web service
- and get a result from it

Officially supported via REST APIs, Node.js SDK, and on Firebase



Weather Web App

Last step: adding a Webhook…

References

- MDN web docs - Web Speech API
  - https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
- MDN web docs - Using the Web Speech API
  - https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API
- Introduction to the Web Speech API
  - https://developers.google.com/web/updates/2013/01/Voice-Driven-Web-Apps-Introduction-to-the-Web-Speech-API
- HTML5 Speech Recognition API – Demo (by Google)
  - https://www.google.com/intl/en/chrome/demos/speech.html
- Dialogflow Documentation
  - https://cloud.google.com/dialogflow/docs/

License

- These slides are distributed under a Creative Commons license "**Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)**"

- **You are free to:**
  - **Share** — copy and redistribute the material in any medium or format
  - **Adapt** — remix, transform, and build upon the material
  - The licensor cannot revoke these freedoms as long as you follow the license terms.

- **Under the following terms:**
  - **Attribution** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **NonCommercial** — You may not use the material for commercial purposes.
  - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

- https://creativecommons.org/licenses/by-nc-sa/4.0/