

Een samenvatting van de presentatie die Robby Wiegmans gaf op de Polteq Conferentie 2019.

Voice User Interface is ouder dan je denkt

De introductie van Voice User Interface (VUI) kwam voor het grote publiek in 2011 met Siri op de iPhone 4s. Je kon toen bijvoorbeeld met je stem het weerbericht raadplegen. Mensen vonden dit best indrukwekkend. Toch is VUI al heel wat jaren ouder. In 1952 introduceerde Bell 'Audrey': een spraakherkenner die 0 tot 9 verstond en deze op beeld kon weergeven. In 1961 kwam IBM met de 'shoebox': een spraak gestuurde rekenmachine.

Zeg pizza en het is bezorgd

Nu zijn we een paar jaar later en de meesten kennen Alexa van Amazon of Google Home. Voor de niet-wetende: dit zijn virtuele assistenten, al dan niet ingebouwd in een speaker. Elk met hun eigen App Store (Amazons Skills en Google's Actions), waarmee je zelf Skills kan toevoegen. Een voorbeeld van zo'n Skill is Domino's, waarmee je jouw favoriete pizza bestelt door te zeggen: "Alexa, ask Domino's to order my favorite pizza." Met deze actiewoorden: 'Order' en 'Favorite pizza', wordt jouw favoriete pizza direct besteld, betaald en bezorgd op jouw adres. Simpel en gebruiksvriendelijk: oftewel een 100% user experience gevoel.

Zij verstaat mij niet

Voor de Polteq conferentie wilde ik ook zo'n Skill ontwikkelen: de gebruiker vraagt naar een presentatie en Alexa antwoordt in welke ruimte dit moet plaatsvinden. Het ontwikkelen ging redelijk simpel met het volgen van tutorials en guidelines; een halve dag en ik had mijn eerste Alexa Skill klaar. Getest (uiteraard!) en wel, introduceerde ik enthousiast de Polteq Conferentie Skill bij mijn vrouw. En toen kwam het probleem: ze wist niet welke actiewoorden ze moest gebruiken om te werken met de Skill. Na wat hulp had ze niet alleen nog maar de oppervlakte geschraapt van wat de Skill kon, maar omdat ze een andere zinsopbouw gebruikt dan ik, verstond Alexa ook niet alle spraakcommando's en wisten zowel Alexa als mijn vrouw niet wat ze moesten doen. Weg user experience!

Nog meer zinnen toevoegen

Terug naar de tekentafel, waar ik de oplossing vond in het toevoegen van vele 'utterances' en 'fallback intents'. Kortgezegd zijn 'utterances' verschillende soorten zinnen om hetzelfde commando te activeren. En 'fallback intents' zijn een if, then, else, zodat een gebruiker wordt geholpen om bij de juiste actie uit te komen. Maar als ik, naast de tientallen commando's, ook nog tientallen verschillende zinnen mondeling moet testen, dan zijn dat er enkele honderden. Geen vrolijke gedachten.

De ene testtool is de andere niet

Testen kan je met echte devices of simulators als EchoSimIO, maar daar moet je alsnog tegenaan praten. De testtool van Amazon biedt de mogelijkheid om JSON-bestanden met uitgeschreven zinnen te importeren. Voordeel is dat je regressietests betrouwbaar zijn en tests in enkele secondes klaar zijn. Nadeel is het gebrek aan paralleltests, CI/CD omgeving, en testrapportage. Wil je dit wel, dan is de commerciële tool 'Botium' interessant, met een prijskaartje van 999,- dollar per maand.

Maar brengt het toevoegen van honderden zinnen echt gebruiksvriendelijkheid in je Skill?

Nee, ik heb geleerd om eindgebruikers vooraf al te betrekken hoe het gedrag van de Skill moet zijn. Daarnaast biedt het developersdashboard van Amazon inzicht in het gebruikersgedrag: hoeveel 'utterances' elk commando heeft en hoe vaak 'utterances' slagen of falen. Pas daarna komt het toevoegen van verschillende zinnen.

Conclusie

- Vergroot verschillende manieren waarop eindgebruikers verzoeken kunnen verwoorden;
- Evalueer hoe gemakkelijk de spraakherkenning is;
- Verbeter het taalbegrip;
- Zorg dat gebruikers natuurlijk en spontaan kunnen praten met jouw Skill;
- Zorg dat de Skill de meeste verzoeken begrijpt: binnen de context van de functionaliteiten van de Skill;
- Zorg dat de Skill op de juiste manier reageert op de verzoeken van gebruikers, door ze uit te voeren of uit te leggen waarom ze dat niet kan;

[Download hier de presentatie van Robby Wiegmans over het testen van Alexa.](#)