

Mikko Poutanen

Chatbot web -sovelluksen suunnittelu ja kehitys

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinööriytyö

14.11.2018

Tekijä Otsikko	Mikko Poutanen Chatbot web -sovelluksen suunnittelu ja kehitys
Sivumäärä Aika	34 sivua 14.11.2018
Tutkinto	insinööri (AMK)
Tutkinto-ohjelma	tieto- ja viestintäteknikka
Ammatillinen pääaine	Smart Systems and Software Engineering
Ohjaajat	Lehtori Sami Sainio
<p>Keskustelevat käyttöliittymät ovat yleistyneet viime vuosina huomattavasti erilaisten viesti- ja verkkopalveluiden kautta toimivien chatbottien muodossa. Chatbotteja voidaan käyttää monissa eri yhteyksissä, ja ne voivat tehostaa yritysten toimintaa niin asiakaspalvelutyössä kuin sisäisissä prosesseissa. Tämän insinööriyön tavoitteena oli selvittää keskustelevan käyttöliittymän vaatimukset ja ominaisuudet alan kirjallisuudesta ja verkkojulkaisuista, vertailla eri palveluntarjoajien pilvipohjaisia chatbot-alustoja ja toteuttaa niistä yhdellä chatbot-verkkopalvelu yrityksen asiakaspalvelutarpeisiin.</p> <p>Toistaiseksi Suomea puhuvat chatbotit ovat olleet melko harvinaisia johtuen puutteellisesta kielituesta yleisimmissä chatbot-alustoissa. Nykyteknologioilla on kuitenkin mahdollista toteuttaa rajallisesti Suomea ymmärtäviä botteja. Tämän mahdollistaa monissa chatbot-alustoissa käytetyt koneoppimisalgoritmit, jotka luokittelevat käyttäjän kirjoittamaa tai puhumaa syötettä aikomusten ja entiteettien perusteella. Näin ollen, kun algoritmille syötetään tarpeeksi suomenkielistä dataa, se voi oppia tunnistamaan näitä ominaisuuksia vapaamuotoisesta tekstistä, ja vastaamaan sitten ennalta määrätyllä tavalla. Projektin tarkoituksena oli botin kehityksen ohella myös kerätä olennaiset havainnot ja haasteet yleisesti chatbot-palvelujen suunnitteluun ja kehitykseen liittyen, sekä arvioida eri alustojen soveltuvuutta ja ominaisuuksia erilaisia suomenkielisiä käyttötarkoituksia varten.</p> <p>Työn tuloksena rakennettiin IBM Cloud -palvelun tarjoamia Watson-rajapintoja käyttäen toimiva chatbot-web-sovellus, joka kykenee vastaamaan suomeksi yrityksen asiakaspalveluun liittyviin kysymyksiin, kerää eri asiakkaiden kanssa käydyistä keskusteluista tilastietokantaa tietokantaan. Sen kykyä vastata kysymyksiin ja käydä keskusteluja voi laajentaa tarpeen myötä ilman suurta vaivaa tai teknistä osaamista. Koneoppimistekniikan kehittyessä kiihtyvää tahtia, yhä suuremman osan tyypillisistä asiakaspalvelutilanteista pystyy teoriassa hoitamaan älykkäät botit, jotka ajan myötä oppivat paremmiksi tehtävissään. Botit voivat myös tarvittaessa ohjata keskustelun ihmiselle, jos se ei kykene ratkaisemaan asiakkaan ongelmaa.</p>	
Avainsanat	Chatbot, pilvipalvelut, Web-kehitys

Author Title	Mikko Poutanen Chatbot web-application design and implementation
Number of Pages Date	34 pages 14 November 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Smart Systems and Software Engineering
Instructors	Sami Sainio, Teacher
<p>The purpose of this thesis was to study the features and requirements of a conversational interface from relevant literature and online publications, to compare different cloud-providers chatbot-platforms, and to develop a chatbot web-application for a company's customer-service use-case using one of the examined cloud platforms.</p> <p>The goal of this project was to collect the relevant details and challenges relating to the development and design of Chatbots in general, and to compare the suitability of different cloud based chatbot-services to a Finnish speaking chatbot.</p> <p>The result of this project was a chatbot web-application, capable of answering customer-service related questions in Finnish. The application was developed using IBM Bluemix -platforms Watson APIs, and it collects statistics from the Assistants with different users, and its ability to answer questions and have Assistants can also be extended when needed without too much effort.</p> <p>The results show that building conversational interfaces using modern tools is quite a straightforward process. And even building a chatbot using an unsupported language can work in some use-cases. It is very likely that as modern deep learning and speech synthesis technologies develop, we can soon build conversational interfaces that can be very hard to distinguish from actual humans.</p>	
Keywords	Chatbot, cloud-services, web-development

Sisällys

Lyhenteet

1	Johdanto	1
2	Keskustelevat käyttöliittymät ja chatbotit	2
2.1	Keskustelevat käyttöliittymät	2
2.2	Chatbot taustaa	2
2.3	Käyttökohteita ja hyötyjä	3
2.4	Keskeiset ominaisuudet chatboteissa	5
2.4.1	Ohjatun ja vapaan keskustelun toteutus	5
2.4.2	Integroitavuus valmiisiin viestikanaaviin	6
2.4.3	Tuki puheentunnistukselle ja puhesynteesille	7
2.5	Tekoälyn ja koneoppimisen rooli chatboteissa	7
3	Chatbot-alustojen nykytilanne ja vertailu	8
3.1	Pilvipohjaiset chatbot-alustat	9
3.1.1	Amazon Lex	9
3.1.2	Microsoft Bot Framework	10
3.1.3	Google Dialogflow	11
3.1.4	IBM Watson Assistant	12
3.2	Tuki suomen kielelle	13
3.3	Vertailun loppupäätelmät	13
4	Chatbotin suunnittelu	14
4.1	Suunnittelumenetelmä ja arkkitehtuuri	15
4.2	Käyttötapaus	18
4.3	Käyttöliittymä	18
4.4	Keskustelun sisältö	20
4.5	Palautetoiminnallisuus	21
4.6	Loki	21
5	Chatbotin kehitys	22
5.1	Watson Assistant -palvelu	22
5.1.1	Työtilat	22
5.1.2	Watson Assistant REST API -rajapinta	24
5.2	Front-End ohjelmointi	25

5.3	Back-End ohjelmointi	25
5.4	Keskustelun rungon kehitys	26
5.5	Palauteikkunan toteutus	28
5.6	Lokin tallennus Cloudant –tietokantaan	29
5.7	Tuki Suomen kielelle	30
6	Yhteenveto	31
	Lähteet	32

Lyhenteet

API	Application Programming Interface. Ohjelmointirajapinta, jonka kautta eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoa keskenään.
AWS	Amazon Web Services. Amazon.com yhtiön pilvilaskentapalvelu.
CLI	Command Line Interface. Komentorivisovellus.
CSS	Cascading Style Sheets. Verkkosivujen tyyliohjeiden määrittelyyn käytetty kieli.
CSV	Comma Separated Values. Taulukkomuotoisen tiedon tallentamiseen käytetty tiedostomuoto.
CRUD	Create, Read, Update, Delete. Tiedon tallentamisen, lukemisen ja päivittämisen yhteydessä käytetyt perustoiminnot.
DOM	Document Object Model. Dokumenttimalli, jolla kuvataan esimerkiksi verkkosivun rakenne puuna.
ES5	EcmaScript 5 standardi. Viides versio JavaScript kielen standardoinnista.
HTML5	Hypertext Markup Language -merkintäkielen viides versio.
IoT	Internet of Things, esineiden internet.
JSON	JavaScript Object Notation. Yksinkertainen tiedostomuoto tiedonvälitykseen.
MVC	Model View Controller. Ohjelmistoarkkitehtuurityyli, jossa käyttöliittymä erotetaan itse sovelluksen tiedoista.
NoSQL	Not Only SQL. Yleisnimitys relaatiomallista poikkeaville tietokannoille.
REST	Representational State Transfer. HTTP-protokollaan perustuva malli rajapintojen toteuttamiseen.

UI User Interface, käyttöliittymä.

URL Uniform resource locator. Verkkosivun tai tiedoston sijainnin osoittamiseen käytettävä merkkijono.

1 Johdanto

Keskustelevat käyttöliittymät ovat yleistyneet viime vuosina huomattavasti erilaisten viesti- ja verkkopalveluiden kautta toimivien chatbottien muodossa. Yhä useammat yritykset ja muut organisaatiot ovat ottaneet käyttöön erityyppisiä chatbotteja etenkin ulkomailla (1; 2). Suomessa esimerkiksi KELA ja vakuutusyhtiö If ovat ottaneet chatbotteja käyttöön (3; 4). Chatbottien ympärille muodostuvalle liiketoiminnalle on ennustettu jopa satojen miljoonien markkinoita. (5; 6.)

Chatbottien mahdolliset käyttötapaukset ovat hyvin laajat ja niiden avulla voidaan tehostaa toimialasta riippumatta yritysten toimintaa niin asiakaspalvelussa kuin sisäisissä prosesseissakin.

Tämän insinööriyön tarkoituksena on suunnitella ja toteuttaa chatbot-sovellus, joka kykenee vastaamaan yrityksen asiakkaiden kysymyksiin suomeksi verkkosivulla toimivan käyttöliittymän kautta. Tavoitteena on myös selvittää, millaisia ominaisuuksia vaaditaan hyvältä keskustelevalta käyttöliittymältä. Lisäksi tavoite on kerätä suomenkielisen chatbot-sovelluksen kehittämiseen liittyviä havaintoja ja haasteita.

Osana projektia pyritään vertailemaan neljän tunnetun pilvipohjaisen chatbot-alustan perusominaisuuksia ja soveltuvuutta suomenkielisen keskustelevan käyttöliittymän kehittämiseen.

Työn tuloksena rakennettiin IBM Cloud -pilvialustan tarjoamia Watson -rajapintoja käyttäen toimiva chatbot-sovellus, joka kykenee vastaamaan suomenkielellä yrityksen asiakaspalveluun liittyviin kysymyksiin. Asiakkaiden kanssa käydyistä keskusteluista pitää pystyä keräämään статистиikkaa tietokantaan palvelun ylläpitoa ja jatkokehitystä varten. Sovellus kerää tietoa keskusteluistaan NoSQL-tietokantaan, josta ne saa helposti analysoitaviksi esimerkiksi Watson Analytics -pilvipalveluun. Botin kykyä vastata kysymyksiin ja käydä keskusteluja pitäisi voida laajentaa tarpeen myötä ilman suurta vaivaa tai teknistä osaamista, mikä onnistuu esimerkiksi jopa Microsoft Excel -ohjelman avulla, sillä chatbotin opettamiseen tarvittavat lause- ja sanaparit voidaan kirjata taulukkoon myös *vähemmän teknisen* henkilön toimesta ja tuoda niistä Watson Assistant -palveluun CSV -muodossa automatisoidusti.

Koneoppimistekniikan kehittyessä kiihtyvää tahtia yhä suuremman osan tyypillisistä asiakaspalvelutilanteista pystyvät ainakin teoriassa hoitamaan älykkäät botit, jotka ajan myötä oppivat jatkuvasti paremmiksi tehtävissään. Keskustelevat käyttöliittymät yleistyvät tulevien vuosien aikana myös yhä useammassa kodinkoneissa ja arkipäiväisissä asioissa esimerkiksi puhuvien jääkaappien ja televisioiden muodossa. Myös esimerkiksi Yhdysvaltojen ja Israelin armeijat ovat kertoneet tutkivansa keskustelevien käyttöliittymien mahdollisuuksia osana omaa toimintaansa. Tekoälyn ja koneoppimisen kehittyessä tietokoneista ja käyttöliittymistä vaikuttaa tulevan jatkuvissa määrin ihmisenkaltaisempia ja voi olla, että joidenkin vuosien päästä on lähes mahdoton tunnistaa, onko keskustelun tai puhelun toinen osapuoli ihminen vai kone.

2 Keskustelevat käyttöliittymät ja chatbotit

2.1 Keskustelevat käyttöliittymät

Keskustelevat käyttöliittymät ovat uudenlainen tapa ihmisen ja koneen väliseen kommunikointiin. Ne mahdollistavat luonnollisen kielen käytön, niin puhuttuna kuin kirjoitettuna, tapana hankkia, luoda ja prosessoida informaatiota. Keskustelevia käyttöliittymiä voi ajatella osana jatkumoa, jossa on siirrytty aluksi komentoriviin pohjautuneista käyttöliittymistä graafisiin käyttöliittymiin, joissa navigointi tapahtuu esimerkiksi hiiren tai kosketuksen avulla. Uudet keskustelevat käyttöliittymät muistuttavat komentorivisovelluksia etenkin tekstipohjaisuudessaan, mutta huomattava ero on, ettei käyttäjän enää tarvitse välttämättä syöttää komentoja tietyssä formaatissa, vaan hän voi esittää kysymyksen tai kuvailla mitä haluaisi tehdä luonnollisella kielellä keskustelunomaisesti. Tämä parantaa sovellusten käyttökokemusta, ja sisältää paljon mahdollisuuksia keskustelevien käyttöliittymien kehittymiseen tulevaisuudessa.

2.2 Chatbotin taustaa

Chatbotit voidaan nähdä yhtenä keskustelevan käyttöliittymän ilmentymänä. Chatbottien määritelmässä on jonkin verran vaihtelua liittyen niiden eri ominaisuuksiin, mutta yleistä niille kaikille on tietokoneohjelman kyky käydä, tai ainakin simuloida, keskustelua ihmisen kanssa tekstiä tai ääntä käyttäen (7). Chatbottien toteutustavat voivat olla hyvinkin

erilaisia vaihdellen yksinkertaisista sääntöpohjaisista menetelmistä kompleksisempiin tekoälyä ja koneoppimisalgoritmeja hyödyntäviin tekniikoihin.

Chatbottien historian kannalta merkittävässä roolissa on ollut brittiläisen matemaatikon Alan Turingin työ 1950-luvulta alkaen. Hän määritteli vuonna 1950 julkaistussa artikkelissaan nykyään Turingin testiksi kutsutun kokeen, jossa ihminen käy keskustelua kahden osapuolen kanssa, joista toinen on tietokoneohjelma, ja toinen ihminen (8). Turing pohti koetta kehittäessään vastausta kysymykseen siitä, voivatko tietokoneet ajatella. Hänen mielestään konetta voidaan pitää älykkäänä, mikäli koehenkilö ei pysty erottamaan, kumpi keskusteluistaan on koneen ja kumpi ihmisen kanssa.

Ensimmäisenä chatbot-sovelluksena pidetään yleisesti vuonna 1966 kehitettyä ELIZA -nimistä tietokoneohjelmaa. Se oli rakennettu matkimaan psykoterapeutin esittämiä kysymyksiä ja onnistui hämäämään aikanaan useita ihmisiä. Ohjelma ei kuitenkaan käyttänyt monimutkaisia tekoälymenetelmiä, vaan sen toimintaperiaate perustui tiettyjen avainsanojen läsnäoloon käyttäjien syötteissä. Jos esimerkiksi käyttäjän viesti sisälsi sanan "äiti", ohjelma pyysi häntä kertomaan lisää perheestään. (9.)

Esimerkkejä nykyään toimivista boteista on mm. Kelan opiskelijoita opintotukimuutoksessa neuvonut suomenkielinen chatbot ja If-vakuutusyhtiön chatbot.

2.3 Käyttökohteita ja hyötyjä

Chatbottien ja keskustelevien käyttöliittymien mahdolliset käyttökohteet ovat hyvin laajat ja uusia keksitään koko ajan teknologian kehittyessä.

Asiakaspalvelusta löytyy paljon käyttökohteita ja toteutus esimerkkejä (10). Monessa tapauksessa asiakaspalveluun esitetyt kysymykset ovat melko samanlaisia, joten botin voi opettaa vastaamaan näihin yleisiin kysymyksiin ja tarvittaessa ohjaamaan keskustelun oikealle ihmiselle, mikäli botti ei osaa vastata asiakkaan tarpeisiin. Tämän tyylliset käyttötapaukset soveltuvat moniin yksityisen sektorin liiketoimintaa harjoittavien yritysten tarpeisiin etenkin kauppa- ja palvelualalla. (11.)

Puhelinpalvelukeskus on yleinen asiakaspalveluun liittyvä chatbottien käyttökohde. Puhelinpalveluun tulee paljon usein toistuvia kysymyksiä, joihin vastaaminen saadaan automatisoitua pitkälle chatbottien avulla. Säästyy resursseja, kun kone voi hoitaa yksinkertaisiin kysymyksiin vastaamisen, jolloin ihmisten (usein kallis) aika jää haastavampien tilanteiden käsittelyyn.

Palveluun ohjauksessa botin rooli on kysyä asiakkaalta tarpeita ja toiveita, jonka jälkeen se voi tarjota suoran tai viitteen tahoon, joka osaa auttaa tilanteessa. Esimerkkinä terveydenhoitoa tarjoava yritys, jolla on paljon erilaisia yksiköjä ja henkilöitä hoitamaan erilaisia sairauksia. Tutkimusten mukaan myös on havaittu, että ihmiset jakavat esimerkiksi terveyteensä ja elintapoihinsa liittyvää informaatiota mieluummin koneelle kuin tuntemattomalle ihmiselle (12).

Mielenkiintoinen käyttötapaus on virtuaalisen opinto-ohjaajan toteutus chatbotin avulla. Opiskelijoiden tiedontarve ja kysymykset ovat vuodesta toiseen melko samanlaisia opilaitos ja kurssikohtaisestikin tarkasteltuna. Botti voi tarjota apua viikonpäivästä ja vuorokaudenajasta riippumatta väsymättä. Botti myös vastaa usein nopeammin kuin niukasti resursoitu henkilökunta ehtii. (13;14.)

Sisäisten prosessien ja tiedon saatavuuden suoraviivaistaminen on chatbottien yksi tunnistettu käyttötapaus. Jos esimerkiksi suuren yrityksen data on hajautettu monen eri järjestelmän tai osaston kesken, niin chatbot voi tarjota nopean tavan saada tarvittavaa tietoa esittämällä yksinkertaisia kysymyksiä botille sen sijaan, että joutuisi itse etsimään vastauksen mahdollisesti useasta tietolähteestä.

Esineiden Internet, eli Internet of Things (IoT), tarjoaa myös monenlaisia mahdollisia käyttökohteita keskusteleville käyttöliittymille esimerkiksi kotiautomaation suhteen. Hyviä esimerkkejä tästä on esimerkiksi Amazon Alexa, Google Home ja Apple HomePod.

Keskustelevat käyttöliittymät voivat tarjota kiinnostavia mahdollisuuksia tulevaisuudessa myös sotilas- ja armeijakäyttöön esimerkiksi yksittäisen sotilaan apuvälineenä. Taistelutilanteessa ihmisen kognitiiviset kyvyt heikkenevät eikä aikaa haluta käyttää ruutujen tai mittareiden katseluun niin paljoa kuin halutaan pitää huomio tilanteessa ja ympäristössä, joten keskustelevien käyttöliittymien avulla voidaan mahdollisesti säästää kallisarvoista aikaa taistelutilanteissa. (15;16;17.)

Viestipalvelujen kuten Slackin tai Facebookin kautta vuorovaikuttavien chatbottien lisäksi chatbotit voivat olla esimerkiksi luonteva tapa uuden käyttäjän opastamiseen mobiilisovelluksissa. Chatbot voi kysyä käyttäjältä tarvittavat esitiedot sovelluksen käyttöön-ottoon, jolloin käyttäjän ei tarvitse täyttää lomakkeita. Mobiilisovelluksiin voi myös integroida keskustelevan käyttöliittymän, joka voi mahdollistaa sovelluksen käytön esimerkiksi ajaessa, kokatessa tai urheillessa.

2.4 Keskeiset ominaisuudet chatboteissa

Seuraavassa esitetään sellaisia keskeisiä chatbotin ominaisuuksia, joiden avulla käyttäjäkokemus saadaan mahdollisimman hyväksi. Kolmannessa kappaleessa eri bottialustoja vertaillaan keskenään näiden ominaisuuksien suhteen. Kappaleessa neljä tutkitaan, miten nämä ominaisuudet saadaan toteutettua IBM Cloud -alustalla.

Nykyisen teknologian avulla alkeellisen chatbotin toteutus voi olla hyvinkin nopeaa ja helppoa, mutta tuotos voi jäädä melko puutteelliseksi, ellei kiinnitä huomiota seuraaviin seikkoihin. Tärkeää on tarkasti määritellä konteksti, jossa botti toimii. Tähän sisältyy myös keskustelun muoto, eli onko keskustelu ohjattua vai vapaata, kuten myös esimerkiksi botin puhetyyli ja -sävy, näkökulma ja tarkoitus. Alusta, jolla botin toteuttaa, kannattaa myös valita käyttökohteen mukaan, sillä eri alustat tarjoavat toisistaan hieman poikkeavia ominaisuuksia esimerkiksi puhesynteesin osalta.

2.4.1 Ohjatun ja vapaan keskustelun toteutus

Chatbottien kanssa käytävät keskustelut voi karkeasti jakaa kahteen luokkaan; ohjattuun ja vapaaseen. Ohjatussa keskustelussa on aina jokin tietty päämäärä, johon botti pyrkii viemään keskustelua, kuten esimerkiksi pöydän varaaminen ravintolasta, elokuvalipun tilaaminen tai jonkin tuotteen osto. Näitä tilanteita yhdistää botin tarve hankkia käyttäjältä tietty määrä informaatiota keskustelun aikana, ennen kuin toiminto voidaan suorittaa. Keskustelun kulkua voitaisiin esittää päätöspuun tai vuokaavion avulla. Jos esimerkiksi käyttäjä haluaa varata pöydän ravintolasta, botin täytyy tietää esim. mihin aikaan ja kuinka monta henkilöä on tulossa. Ohjattuja chatbot-keskusteluja voidaan ajatella hieman inhimillisempänä vaihtoehtona perinteiselle verkkolomakkeiden täyttämiseksi.

Vapaa keskustelu chatbotin kanssa on yleensä kysymys - vastaus tyylinen vuoropuhelu, jossa ei välttämättä ole varsinaista päämäärää, johon botti aktiivisesti yrittäisi ohjata keskustelua, vaan tavoite on jakaa informaatiota yleisesti sitä tarvitseville. Tällaisessa käytössä usein käyttäjä tarvitsee tietoa isosta tietomäärästä, esim. lakimuutoksen vaikutuksista omaan elämään. Esimerkki tällaisesta chatbotista löytyy Suomesta, sillä KELA otti käyttöön vuoden 2017 aikana chatbotin, joka vastasi suomeksi opiskelijoiden kysymyksiin liittyen tuolloin ajankohtaiseen lakimuutokseen.

Keskustelevan käyttöliittymän voi toteuttaa myös ohjatun ja vapaan mallin hybridinä. Tällöin chatbotin kanssa keskustelu noudattaa vapaata mallia, mutta tiettyjen toimintojen toteuttamiseksi chatbot vaihtaa ohjattuun malliin. Tässä työssä toteutettu chatbot on esimerkki tällaisesta hybridistä. Keskustelen avauksen jälkeen se vastaa vapaan mallin mukaan kysymyksiin esimerkiksi yhteystiedoista ja aukioaloajoista, mutta pöytävarausten ja ruokatilausten kohdalla se noudattaa ohjattua mallia tarvittavien tietojen keräämiseksi.

2.4.2 Integroitavuus valmiisiin viestikanaviin

Erilaiset viestisovellukset ovat yleistyneet viime vuosien aikana paljon ja niillä on laaja käyttäjäkunta. Viestisovellusten ja sosiaalisen median väliset rajat ovat myös tietysmielissä hälventyneet ja viestisovelluksia käytetään nykyään myös muuhunkin kuin vain ystävien kanssa kommunikointiin. Monet yhtiöt ja tuotemerkit ovat nykyään tavoitettavissa esimerkiksi Facebook Messenger -sovelluksen kautta. Palvelut, jotka keskittyivät aiemmin pääasiassa viestien, kuvien ja videoiden jakamiseen ovat kehittyneet omiksi ekosysteemeikseen, joilla on omat kehittäjäyhteisöt, sovellukset ja rajapinnat. (10.)

Viestisovelluksilla on useita erityisiä piirteitä, jotka tekevät niiden käyttäjäkunnasta houkuttelevan kohdeyleisön monille yrityksille ja markkinoijille, kuten käyttäjäkuntien suuri koko, uskollisuus ja sovellusten korkea käyttöaste (10). Integroimalla chatbot tällaiseen kanavaan voi yritys helposti uusia asiakkaita sekä lisää näkyvyyttä.

Chatbot-sovelluksen toteutuksen kannalta merkittävä hyöty on se, että chatbotille ei tarvitse itse rakentaa käyttöliittymää. Tämä vähentää merkittävästi toteutus- ja ylläpitokustannuksia, sillä jo pelkkä käyttöliittymäsuunnittelu voi viedä kymmenistä yli sataan työtuntia puhumattakaan itse toteutuksesta.

2.4.3 Tuki puheentunnistukselle ja puhesynteesille

Kun chatbot kykenee käymään keskustelua myös äänen avulla, kasvavat sen mahdolliset käyttötilanteet, sillä käyttäjän ei tarvitse olla näppäimistön äärellä antaakseen komentoja botille. Puhuminen on myös nopeampaa kuin kirjoittaminen, mikä tekee botin käyttämisestä ylipäätään nopeampaa.

Hyödyntämällä nykyaikaisia chatbot-alustoja, on puhuvan chatbotin toteuttaminen suhteellisen yksinkertaista. Kaikki luvussa 3.1 vertailut palvelut tarjoavat jonkinlaiset *Text to Speech*- ja *Speech to Text* -ratkaisut, joiden avulla tuen puhutuille keskusteluille voi toteuttaa. Niiden avulla voi tehdä samaa keskustelulogiikkaa noudattavan botin tavoitettavaksi sekä tekstin että puheen kautta ilman suurta lisätyömäärää.

Tässä työssä suunniteltu chatbot päätettiin toteuttaa ilman tukea puhutuille keskusteluille, sillä se ei ollut kyseisen käyttötapauksen kannalta merkittävässä roolissa. Lisäksi tuki suomenkieliselle puheentunnistukselle ja puhesynteesille on harvinaista nykyisissä chatbottien rakentamiseen tarkoitetuissa palveluissa.

2.5 Tekoälyn ja koneoppimisen rooli chatboteissa

Toteutuksen kannalta tarkasteltuna chatbotin toiminta voidaan jakaa kahteen vaiheeseen. Ensimmäisessä vaiheessa pitää pystyä tunnistamaan, mitä käyttäjä tarkoittaa kysymyksellään (avoin keskustelu) tai vastauksellaan (ohjattu keskustelu). Toisessa vaiheessa tuotetaan vastaus käyttäjälle.

Monet nykyään käytössä olevat chatbot-sovellukset toteuttavat ensimmäisen vaiheen hyödyntäen koneoppimismenetelmiä (viite). Myös merkkijonojen tai avainsanojen tunnistamiseen perustuvia ratkaisuja on olemassa.

Toisessa vaiheessa vastauksen tuottaminen pitää pystyä tekemään eri tavoilla riippuen siitä, millaisella logiikalla se on tuotettavissa. Nykyiset keskustelevat käyttöliittymät jakaantuvat kahteen ryhmään niiden vastauslogiikan pohjalla olevan mallin perusteella.

Harvinaisemmat generatiiviset mallit tuottavat vastaukset itse. Ne perustuvat yleensä *Recurrent Neural Network* -pohjaiseen sekvenssimalliin ja tuottavat vastauksen perus-

tuen niiden aiemmin näkemiin syöte-vastaus-pareihin. Tämänkaltaiset mallit ovat liiketoimintakäytössä harvinaisia, sillä koska malli tuottaa jokaisen vastauksen ”tyhjästä”, mikä aiheuttaa sen, ettei etukäteen voi varmasti tietää, miten chatbot vastaa tiettyyn syötteeseen. Se voi myös vastata samaan syötteeseen eri tavalla eri kerroilla.

Suurin osa nykyään toiminnassa olevista chatbot-sovelluksista hyödyntää koneoppimismenetelmiä käyttäjän aikeen tai kysymyksen tunnistamiseen. Sen jälkeen vastaus haetaan tai koostetaan valmiiksi olemassa olevasta aineistosta. Vastaukset saattavat olla myös ennalta kirjoitettuja, jolloin tekoälyn rooli liittyy ainoastaan käyttäjän syötteen luokitteluun.

Tekoälyä ja koneoppimista hyödyntävien pilvipalveluiden ja ohjelmointirajapintojen määrän lisääntyessä, sekä niiden laadun ja käytettävyyden parantuessa, älykkäiden bottien rakentamisesta vaikuttaa tulevan jatkuvasti suoraviivaisempaa ja helpompaa.

3 Chatbot-alustojen nykytilanne ja vertailu

Tässä luvussa chatbot-alustoja vertaillaan teoriaosuudessa tunnistettujen keskeisten ominaisuuksien perusteella. Lisäksi vertailukriteereinä on ohjelmointirajapintojen määrä, kattavuus, kompleksisuus, laajuus, eli miten paljon eri toiminnallisuutta on saatavilla, ja miten monen eri rajapinnan takana ne ovat, ja kuinka monipuolisiin käyttötarkoituksiin ne soveltuvat. Käyttäjystävällisyys otettiin myös osaksi vertailua. Tällä tarkoitetaan tässä yhteydessä esimerkiksi alustan ”oppimiskäyrää”, dokumentaatiota ja alustan ”ekosysteemiä” sekä kehittäjäyhteisöä. Laadukas dokumentaatio ja hyvät käyttöesimerkit nopeuttavat huomattavasti uuden alustan/kirjaston oppimista ja käyttöönottoa.

Chatbot-alustan hyödyntäminen osana kehitysprosessia on järkevää resurssien kannalta. Keskustelevan käyttöliittymän alusta asti itse ohjelmoiminen voi olla hidasta, etenkin jos botin käyttötapaus vaatii vapaan luonnollisen kielen käsittelyä, eikä vain esimerkiksi ennalta määrättyjä kysymys-vastus-pareja.

3.1 Pilvipohjaiset chatbot-alustat

Pilvipohjaiset chatbot-alustat ja sovelluskehitystyökalut ovat yleistyneet viime vuosina. Kaikilla suurilla pilvipalvelujen tarjoajilla on valikoimassaan chatbottien kehitykseen suunnattuja tuotteita. Tässä lopputyössä vertailun kohteeksi valittiin Amazon Lex-, Google Dialogflow, IBM Watson Assistant- ja Microsoft Bot Framework -palvelut. On myös olemassa näiden alustojen ylle rakennettuja palveluja ja ohjelmistokirjastoja, jotka pyrkivät yksinkertaistamaan botin ja dialogirakenteen kehitysprosessia. Monet alustat mahdollistavat myös saman chatbotin (keskustelurungon) käyttöönoton samanaikaisesti usealla eri viestikanaavalla.

Puheentunnistus ja luonnollisen kielen ”ymmärtäminen” ovat mm. laskennallisesti erittäin haastavia ongelmia tietokoneelle, ja ne vaativat yleensä hyvin suurella data määrällä koulutettuja syväoppimisalgoritmeja ja paljon laskentatehoa esimerkiksi näytönohjaimista. Näin ollen pilvipohjaisen alustan käyttö huomattavasti nopeuttaa chatbot-sovelluksen kehitysprosessia ja vähentää siihen vaadittavia resursseja.

3.1.1 Amazon Lex

Amazon Lex pohjautuu samaan koneoppimisteknologiaan, jota Amazonin Alexa -virtuaaliassistentti käyttää. Hyviä puolia on tuki sekä tekstilelle että äänelle keskusteluissa, laajat integrointimahdollisuudet muihin AWS-palveluihin Amazon Lambda -funktioiden avulla ja kohtuullinen hinnoittelu, jossa hinta muodostuu vain käytettyjen REST API -kutsujen määrästä.

Lex alusta tukee sekä vapaan että ohjatun keskustelun toteutusta. Tyypilliset vaiheet botin luomisessa Lex alustalla ovat seuraavanlaiset. Ensin luodaan botti käyttäen joko graafista Amazon Lex Console -käyttöliittymää tai AWS CLI -ohjelmaa, jonka jälkeen se konfiguroidaan tunnistamaan yksi tai useampi ”intent” eli aikomus. Seuraavaksi määritellään vastaukset ja mahdolliset lisäkysymykset, joita mahdollisesti tarvitaan käyttäjän toiveen täyttämiseen. Yksi ”intent” voi vaatia nollaa tai useampaa niin kutsuttua ”slot”-parametria. Nämä kuvastavat informaatiota, jonka botti tarvitsee täyttääkseen käyttäjän pyynnön. Jos esimerkiksi käyttäjä haluaisi ostaa elokuvateatterin chatbotilta lipun, botin täytyy tietää, minkä elokuvan mihin näytökseen ja montako lippua käyttäjä haluaa, ennen kuin tilaus voidaan suorittaa. Tämä on hyvä esimerkki ohjatusta keskustelusta. Vapaan

keskustelun toteutus on hyvin samanlaista, mutta siinä ei ole tarvetta käyttää "slot"-parametreja, sillä botti voi suoraan aikomuksen tunnistettuaan vastata siihen sopivalla tavalla.

Suomen kieli ei ole virallisesti tuettuna Amazon Lex -alustassa. Ainoa virallisesti tuettu kieli on englanti. Vertailluista alustoista Lex omaisi suppeimman kielivalikoiman.

Lex alusta tarjoaa valmiiksi melko kattavat integrointimahdollisuudet sen avulla rakennetuille boteille. Kun botin on kerran toteuttanut, se on mahdollista ottaa käyttöön usealla eri kanavalla, kuten Facebook Messenger, Slack, Kik tai Twilio SMS. Lex -alustaan sisäänrakennettu tuki AWS Lambda -palvelulle tarjoaa helpon integraatioväylän moniin muihin AWS -palveluihin.

Lex alusta ei tarjoa itsessään puhesynteesiä, vaan se on osa Amazon Polly -nimistä palvelua. Saman Lex alustalla tehdyn botin voi kuitenkin konfiguroida kommunikoimaan sekä puheen että tekstin avulla.

3.1.2 Microsoft Bot Framework

Microsoftin Azure -alustan kautta tarjoama Bot Framework on kokoelma palveluita chatbottien keskustelurungon rakentamiseen ja testaamiseen. Bot Framework sisältää myös bottien käyttöönottoon ja hallintoihin tarkoitettuja työkaluja selaimen kautta. Se tukee tekstin ja puheen lisäksi niin kutsuttuja vastauskortteja, jotka muistuttavat monivalintalomakkeita. Tämä on melko useissa käyttötapauksissa hyödyllinen ominaisuus, sillä se säästää aikaa esimerkiksi web-käyttöliittymän toteutuksessa.

Azure -palvelun "AI + Cognitive Services" -nimisen kategorian alla on melko laaja joukko eri palveluita, mikä toisaalta on hyvä asia, sillä sovelluskehittäjä saa enemmän mahdollisuuksia ja vaihtoehtoisia toteuttamistapoja tarpeen mukaan. Toisaalta taas vaihtoehtojen runsaus nostaa palvelun oppimiskäyrää ja lisää monimutkaisuutta. Tämä nousee esiin etenkin käyttötapauksissa, joissa tarvitaan useamman eri tekoälyrajapinnan hyödyntämistä, esimerkiksi puheentunnistusta ja synteesiä yhdistettynä erilaisiin luonnollisen kielen käsittelyyn tarkoitettuihin rajapintoihin.

Tuettuna on mahdollisuus niin ohjattuun kuin vapaaseenkin keskusteluun ja tapoja toteuttaa keskustelun runko on monia. Yksi vaihtoehto on Azure Bot Service, joka on graafinen web-käyttöliittymä, jossa valitaan aluksi yksi viidestä rungosta lähtökohdaksi botin toteutukselle. Tarjottuna on myös palvelu, jonka avulla voi luoda yksinkertaisen botin kysymys-vastaus-parien avulla suoraan ilman ohjelmointia.

Bottien käyttöönoton suhteen Bot Framework tarjoaa kaksi palvelimen ylläpito ja laskutusmallia, joista ensimmäisessä botti on normaalisti hinnoiteltu Azure web -sovellus, jolle voi asettaa tavanomaiset kapasiteetti- ja skaalausasetukset. Toisessa laskutus- ja ylläpitomallissa botti ajetaan Azure Functions -palvelussa, joka muistuttaa pitkälti AWS Lambda -ajoympäristöä, ja myös hinta muodostuu käytettyjen API-kutsujen määrästä.

Palvelun integraatiomahdollisuudet olemassa oleviin viestikanaviin ovat melko laajat ja tuettuna on esimerkiksi tekstiviestit, Skype, Slack ja tietyt Office 365 -palvelut. Bottien yhdistäminen olemassa oleviin kanaviin tapahtuu Bot Connector -nimisen palvelun avulla.

Microsoft tarjoaa myös avoimen lähdekoodin ohjelmointirajapinnan nimeltään Bot Builder, joka tarjoaa työkaluja botin ohjelmointiin. Tuettuina kielinä on C# ja JavaScript (Node.js). Verkkoselaimen kautta Bot Service -palvelulla luodut botit käyttävät taustalla tätä samaa rajapintaa.

3.1.3 Google Dialogflow

Googlen Dialogflow -palvelu on rakennettu aiemmin API.ai -nimisen chatbot-palvelun päälle, jonka Google osti syyskuussa 2017. Se tarjoaa työkaluja keskustelevien käyttöliittymien kehittämiseen useille eri alustoille. Tuettuna on sekä teksti- että äänipohjaiset botit yli 20 kielellä. Puhesynteesi ja sentimentin analyysi eivät ole saatavilla kaikille kielille eikä Suomi valitettavasti ole tuettuna kielenä. Dialogflow tarjoaa myös valmiiksi koulutettuja botteja sekä "smalltalk" -ominaisuuden, jonka voi halutessaan kytkeä päälle boteille. Se tunnistaa yleisiä kysymyksiä, kuten "mitä kuuluu?" ja "kuka olet?" ja tarjoaa näihin valmiita vastauksia, joita toki voi myös itse muokata. Palvelu hyödyntää taustalla Googlen omia koneoppimisalgoritmeja, joiden avulla se voi tunnistaa yleisiä entiteettejä kuten esimerkiksi päivämäärät, kellonajat ja numerot. Tämä ominaisuus ei kuitenkaan ole tuettuna kaikilla kielillä ja suomenkielisen botin toteutuksessa tästä ei siis ole sinänsä hyötyä.

Tuettuna on myös mahdollisuus rakentaa sekä ohjattuja että vapaita keskusteluja, jotka Dialogflow'n termein ovat lineaarisia tai epälineaarisia. Lineaarisisissa keskusteluissa voidaan keskittyä yhteen tai useampaan "intent" -aikomukseen ja näiden tarkoitus on siis yleensä puuttuvan tiedon kerääminen käyttäjältä jonkin toiminnon, kuten pöytävarauksen toteuttamiseksi. Keskustelujen kontekstit auttavat tiedon jakamisessa luoden luonnollisemman oloisia keskusteluja. Epälineaariset keskustelut voivat haarautua eri aikomusten välillä perustuen käyttäjän vastauksiin.

Dialogflow tarjoaa chatbotin rakentamisen lisäksi käyttöönottoon, ylläpitoon sekä jatkokehitykseen liittyviä työkaluja. Mukana on myös toimintoja, joilla saa kerättyä статистиikkaa botin käymistä keskusteluista käyttäjäkokemuksen parantamista ja mahdollisten ongelmakohtien löytämistä varten. Tarjolla on valmis integraatio 14 eri viestikanavaan, kuten Slackiin ja Facebook Messengeriin.

Valmiin chatbotin integroiminen muihin palveluihin on suhteellisen helppoa, sillä Dialogflow sisältää web-pohjaisen koodieditorin, jonka avulla voi rakentaa serverless-arkkitehtuurin sovelluksia Googlen *Cloud Functions for Firebase* -palvelun avulla. Mukana on myös analytiikka hallintapaneeli, jossa voi tarkastella botin käymien keskustelujen tilastoja mahdollisten ongelmakohtien havaitsemiseksi.

Palvelu on suunniteltu tukemaan tekstipohjaisten bottien lisäksi myös äänen avulla toimivia botteja. Kun käyttäjä on jo rakentanut tekstipohjaisen botin, on sen muuntaminen myös äänellä toimivaksi melko yksinkertaista. Tämä ei kuitenkaan päde suomenkielisen botin kohdalla puuttuvan kielituen vuoksi. Äänibottien toimintatavaksi voi valita reaaliaikaisen tai synkronisen puheentunnistuksen, mikä laajentaa mahdollisten käyttötapauksien määrää ja vähentää tarvittavaa työmäärää. Lisäksi tarjolla on toiminto, jolla botille voi rekisteröidä puhelinnumeron, johon soittaessa pääsee puhumaan suoraan botille. Tämä tuo mieleen vuonna 2018 Googlen julkistaman Duplex-palvelun, joka pystyy esimerkiksi varaamaan käyttäjän puolesta pöydän ravintolasta soittamalla puhelun ihmiselle.

3.1.4 IBM Watson Assistant

Watson Assistant on IBM Cloud -palvelun kautta tarjottu työkalu ja rajapinta keskusteluvien käyttöliittymien rakentamiseen, jonka avulla voi nopeasti kehittää ja testata chatbotteja. Palvelun vahvuuksiin kuuluu hyvä käytettävyys ja monet valmiit integraatiot eri vies-

tikanaviin ja laaja kielituki. Huonona puolena on käytettävyyden hankaloituminen projektin koon kasvaessa, sillä usean henkilön yhteistyö ja versionhallinta web-käyttöliittymän avulla on haasteellista. Lisäksi erittäin monimutkaisten ohjattujen keskustelurunkojen toteutus graafisesti voi mahdollisesti olla vaivalloista, joten on valitettavaa, ettei palvelu tarjoa vaihtoehtoisia keinoja keskustelurunkojen luontiin.

Watson Assistant -palvelu tarjoaa työkaluja keskustelemaan käyttöliittymän nopeaan kehitykseen, testaamiseen ja käyttöönottoon. Tuettuja keskustelukanavia on monia ja yhden botin kanssa voi keskustella usean eri kanavan kautta samaan aikaan esimerkiksi Facebook Messengerin kautta tai vaikka fyysisen robotin välityksellä.

3.2 Tuki suomen kielelle

Chatbotit ovat siis sovellutus keskustelevasta käyttöliittymästä. On selvää, että ihminen pystyy parhaiten keskustelemaan äidinkielellään. Tästä syystä tässä työssä tutkitaan eri alustojen tukea Suomen kielelle, ja kerätään myös kokemuksia sen toteuttamisesta esimerkkibotin avulla.

Suomen kielen tuki pilvipohjaisissa alustoissa on harvinaista, mutta koska kaikki vertailut alustat pohjautuvat malliin, jossa vastaukset on ennalta määrätty, voidaan niillä ainakin teoriassa toteuttaa Suomeksi kommunikoiva chatbot. Tämän mahdollistaa alustojen syväoppimisalgoritmit, jotka käsittelevät kieltä vain merkkijonoina ilman varsinaista ymmärrystä sanojen merkityksestä. Näin ollen, kun mallille syötetään suomenkielistä esimerkkidataa, oppii se tunnistamaan näitä esimerkkejä riippumatta siitä, millä kielellä esimerkit on kirjoitettu.

3.3 Vertailun loppupäätelmät

Työn aikana huomattiin, että vertailtujen alustojen ominaisuudet olivat hyvin pitkälti samanlaisia eri integrointimahdollisuuksien suhteen ja niiden toimintaperiaatteet muistuttivat toisiaan, kuten myös hinnoittelumallit. Monissa toimintaperiaatteena oli niin kutsuttujen intent- eli aikomus-avainsanojen tunnistus botille syötetystä tekstistä tai äänestä. Avainsanojen tunnistus pohjautuu erilaisiin syväoppimisalgoritmeihin alustasta riippuen. Tämä periaate mahdollistaa yksinkertaisten bottien toteutuksen myös suomeksi, vaikkei

kieli olisikaan virallisesti tuettu. IBM Watson Assistant -palvelulla tämä toteutustapa todettiin toimivaksi. Yhteistä kaikille alustoille oli toiminnallisuus, jonka avulla voidaan luoda melko yksinkertaisia keskustelevia käyttöliittymiä verkkoselaimen kautta visuaalisesti ilman koodin kirjoittamista. Nämä ominaisuudet ovat sopivia etenkin vapaan keskustelun toteutukseen tilanteissa, joissa botti lähinnä vastaa kysymyksiin, joiden vastaukset eivät riipu esimerkiksi tietyn laskutoimituksen tai API-pyyynnön tuloksesta.

Serverless-arkkitehtuurin sovellukset ovat yleistyneet viime vuosina melko paljon. Niiden etuna mainitaan usein ajansäästö etenkin siinä mielessä, että sovelluskehittäjän ei tarvitse huolehtia ohjelman ajoympäristön konfiguroinnista tai palvelimien asentamisesta ynnä muusta vastaavasta. Tällöin aikaa jää enemmän käytettäväksi asiakkaalle lisäarvoa tuottavan logiikan ja koodin tuottamiseen. Kaikissa vertailuissa chatbot-alustoissa oli mukana jonkin asteinen suora integraatio tämänkaltaiseen palveluun. Amazonin Lambda, Microsoftin Cloud Functions, Googlen Firebase ja IBM:n OpenWhisk kaikki tarjoavat helpon tavan integroida keskusteleva käyttöliittymä muihin esimerkiksi REST API:n kautta saavutettaviin taustajärjestelmiin.

Työn chatbot-sovellus päätettiin toteuttaa IBM Cloud -alustan kautta saatavilla olevien Watson-rajapintojen avulla. Esimerkiksi Microsoft Bot Frameworkin tarjoamien rajapintojen määrä oli melko suuri ja tarjoaa sisältää paljon toiminnallisuutta, mutta koska tuo toiminnallisuus on usean eri rajapinnan takana, sovelluksen suunnittelusta tulee siten monimutkaisempaa verrattuna esimerkiksi Watson Assistant -palveluun, jossa lähes kaikki keskustelevaan käyttöliittymään tarvittava toiminnallisuus on yhden rajapinnan takana. IBM Watson -rajapinnat tarjoavat lisäksi myös puhesynteesiin, kuvantunnistukseen ja moniin muihin tekoälyn sovelluksiin liittyviä rajapintoja, joiden integrointi osaksi sovellusta onnistuu joustavasti.

4 Chatbotin suunnittelu

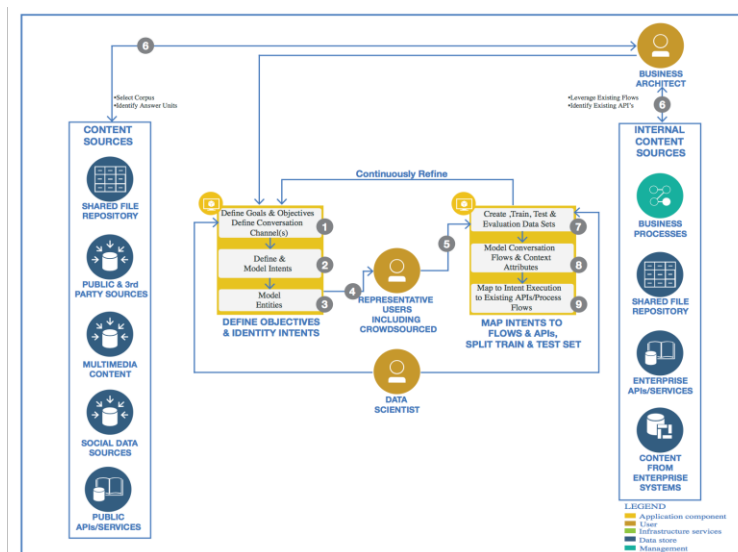
Tässä luvussa kuvataan chatbotin suunnittelusta muutamia keskeisiä vaiheita, jotta lukija saa käsityksen työn laajuudesta ja miten esimerkiksi ohjatun ja avoimen keskustelun käyttö huomioidaan suunnitteluvaiheessa.

Aluksi kuvaillaan menetelmä, sovellusarkkitehtuuri ja sen jälkeen vaatimukset chatbotille. Chatbot toimii kuvitteellisen ravintolan verkkosivulla asiakaspalvelun tukena. Esimerkkikäyttötapaus on laadittu siten, että sen avulla saadaan esille alun teoriaosuudessa esitetyt ominaisuudet.

4.1 Suunnittelumenetelmä ja arkkitehtuuri

Chatbotin suunnittelua ei tarvitse aloittaa tyhjästä, vaan kannattaa käyttää esimerkiksi IBM:n kehittämää menetelmää, joka tarjoaa järjestelmällisen lähestymistavan keskustelelevan palvelun rakentamiseen. (18.)

Alkuvaiheessa on hyvä kartoittaa tiedonlähteet, joista chatbotin kouluttamiseen tarvittava data saadaan. Seuraavaksi on tarpeen määritellä ja mallintaa chatbotin käymien keskustelujen kannalta olennaiset konseptit ja termistö. Jos chatbotin kouluttamista varten käytössä on paljon dataa, on hyödyllistä jakaa tämä data erillisiin osiin, joita voi sitten käyttää erikseen botin kouluttamiseen, arviointiin ja testaamiseen. Kuvan 1 kaavio näyttää suunnitteluprosessin kulun ja vaiheet.



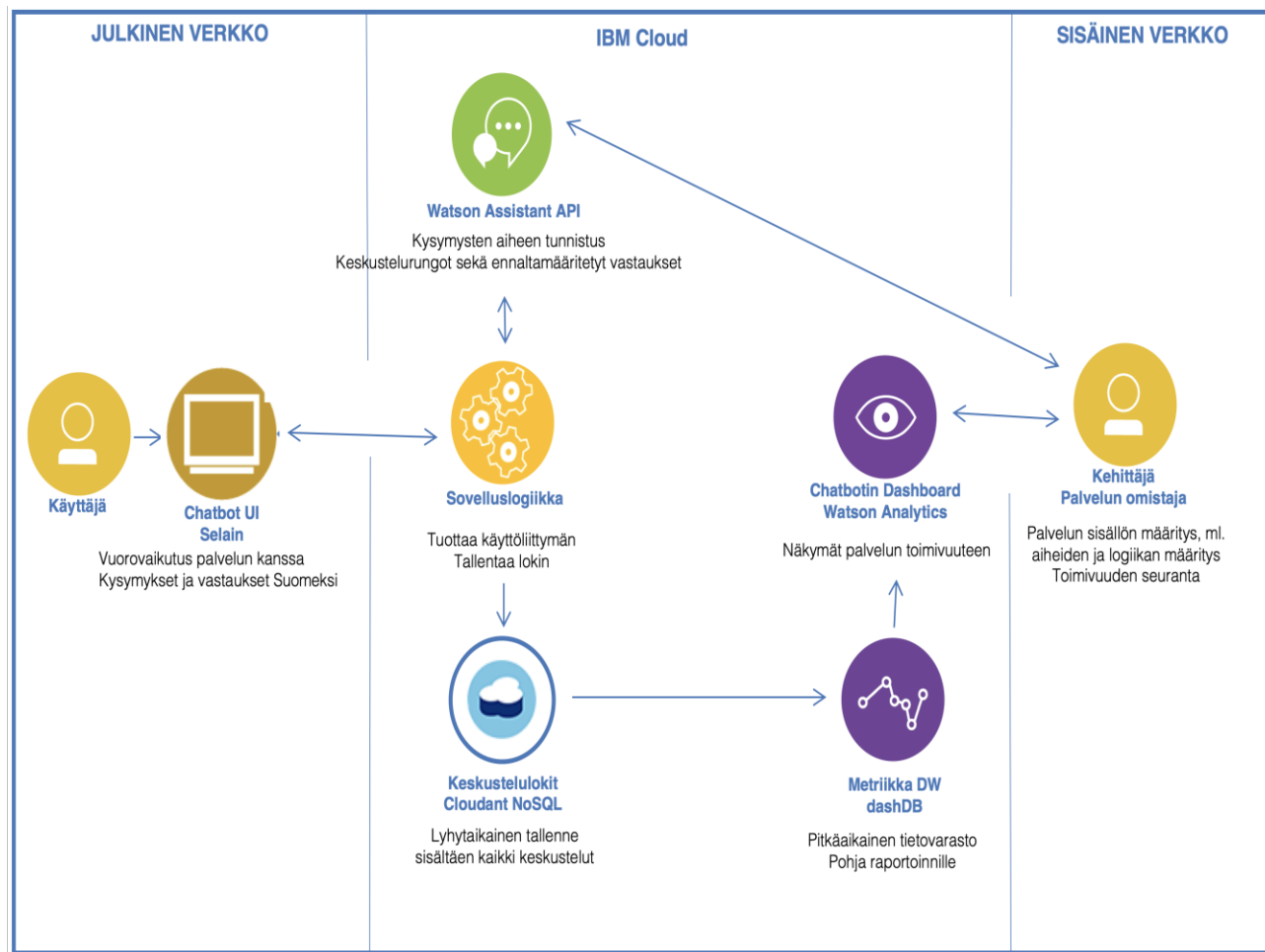
Kuva 1. Prosessikaavio keskustelelevan käyttöliittymän suunnitteluvaiheista (18).

Ensimmäisessä vaiheessa määritellään toiminnot, joita chatbotin halutaan tukevan, ja myös viestikanaavat, joiden kautta chatbotin kanssa on tarkoitus pystyä keskustelemaan. Tämän jälkeen täytyy tunnistaa ja määritellä käyttäjien tarpeet ja aikomukset, jotta niitä voidaan tunnistaa tekstistä. Myös näihin liittyvät keskeiset käsitteet ja termit, joilla käyttäjät voivat tarkentaa aikomuksiaan, on hyödyllistä kartoittaa ja määritellä tässä vaiheessa.

Kun chatbotin toiminnoista ja käyttäjien tarpeista on selkeä kuva, alkaa esimerkkitietojen (kuten kysymysten, käskyjen ja muiden lauseiden) kerääminen, jolla koneoppimiseen perustuva luokittelija-algoritmi koulutetaan. Dataa kerätessä tulee määritellä, mihin chatbotin toimintoon kukin esimerkkilause liittyy. Jos chatbotin on esimerkiksi tarkoitus pystyä antamaan käyttäjälle sääennuste, tulee kehittäjien kerätä mahdollisimman paljon, mieluiten oikeiden käyttäjien kirjoittamia tapoja pyytää sääennustetta tai kysyä säästä. Mikäli dataa saadaan kerättyä paljon, se kannattaa jakaa erillisiin koulutus-, testi- ja arviointiosioihin. Tämän voi toteuttaa esimerkiksi jakamalla data satunnaisesti niin, että 70 % kokonaismäärästä on koulutusta varten, 20 % testaukseen ja 10 % arviointia varten (18).

Seuraavaksi on tarpeen kartoittaa tietolähteet, joita chatbot käyttää vastaustensa muodostamisen tukena, sekä mahdolliset prosessit ja rajapinnat, jotka liittyvät chatbotin toimintaan. Riippuen chatbotin tukemista toiminnoista ja liiketoimintaprosesseista, joiden kanssa se on vuorovaikutuksessa, sen tarvitsemat tietolähteet voivat vaihdella suuresti. Yksinkertaisimmissa käyttötapauksissa (esimerkiksi kyllä- ei-kysymykset) ulkoista tietolähdettä ei sinänsä tarvita, mutta esimerkiksi tuotteen tilaaminen chatbotin kautta on käyttötapaus, jossa sovelluksen täytyy kommunikoida useamman (sisäisen tai ulkoisen) tietokannan tai rajapinnan kanssa ennen kuin se voi antaa käyttäjälle vastauksen.

Teknisen toteutuksen pohjaksi otettiin Watson Assistant -palvelulla kehitettävien chatbottien referenssiarkkitehtuuri, joka on esitetty kuvassa 2. Yleisesti sanottuna kaikki Assistantilla kehitetyt botit noudattavat samaa kaavaa, mikä tehostaa suunnittelua ja toteutusta.



Kuva 2. Korkean tason kaaviokuva Watson chatbot -sovelluksen arkkitehtuurista. Malli on muokattu IBM:n referenssiarkkitehtuurista. (18)

Kuvassa 2 esitetty Chatbot UI on käyttäjän selaimessa toimiva sovellus, joka on vastuussa käyttäjän kirjoittamien viestien ja palautteen lähettämisestä palvelimelle ja chatbotin vastausten esittämisestä käyttäjälle. Chatbot UI kommunikoi IBM Cloud -alustalla sijaitsevan Node.js-prosessin kanssa, joka sisältää kaiken sovelluslogiikan. Tämä palvelin on vastuussa viestien välittämisestä Watson Assistant -palveluun, jossa varsinaisen kielen prosessointi tapahtuu. Palvelin on myös yhteydessä Cloudant NoSQL -tietokantaan, johon se tallentaa lokia käyttäjien keskusteluista. Cloudant-tietokanta on konfiguroitu säilyttämään tämä data vain lyhytaikaisesti, sillä loki replikoidaan tasaisin väliajoin *dashDB*-relaatiotietokantaan. Se on pitkäaikainen tietovarasto, joka toimii pohjana raportoinnille ja analytiikalle. Sieltä data on helposti tuotavissa chatbotin omistavan organisaation työntekijöiden hyödynnettäväksi osana palvelun ylläpito- ja jatkokehitystyötä.

4.2 Käyttötapaus

Toteutettava käyttötapaus on luonteeltaan asiakaspalvelua, jossa sekä asiakas että palvelun tarjoaja hyötyvät chatbotista. Asiakas saa tietoa nopeasti haluamanaan ajankohdasta, eikä hänen tarvitse soittaa ravintolaan eikä etsiä tietoja ravintolan verkkosivuilta. Ravintola puolestaan säästää resursseja, kun henkilöstön tarve vastata puhelinsoittoihin vähenee. Lisäksi hyvin toimiessaan chatbot parantaa asiakaskokemusta.

Ravintolan asiakas saapuu ravintolan verkkosivulle ja hän klikkaa botin kuvaketta, jolloin aukeaa keskustelunäkymä. Näkymä on melko minimalistinen ja sisältää keskusteluruudun lisäksi vain palautepainikkeen. Chatbot tervehtii käyttäjää ja esittelee itsensä ja toimintonsa pikaisesti. Botin on kyettävä vastaamaan yksinkertaisiin ravintolan asiakaspalveluun liittyviin kysymyksiin, kuten

- valikoima ja ruokalistat
- aukioloajat
- pöydän varaus.

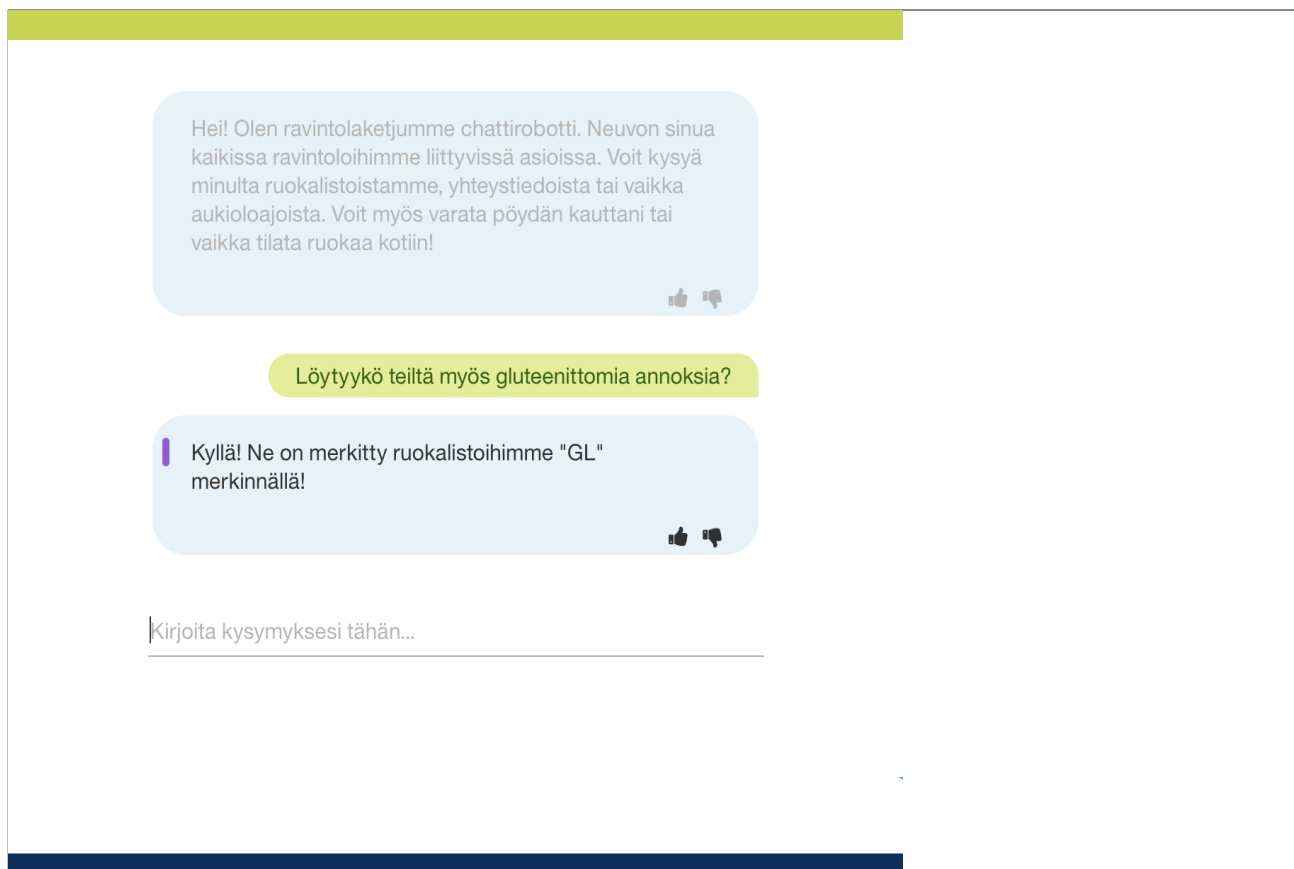
Lisäksi asiakas voi antaa palautetta botin antamasta lisäarvosta ja toimivuudesta palauteikkunan avulla. Käyttäjä voi antaa palautetta chatbotin toiminnasta kokonaisuutena tai yksittäisestä vastauksesta.

4.3 Käyttöliittymä

Sovelluksen käyttöliittymä suunniteltiin melko minimalistiseksi, sillä itse keskustelun rooli chatbot-sovelluksessa on paljon keskeisemmässä roolissa käyttäjäkokemuksen kannalta. Erilaiset visuaaliset käyttöliittymäelementit, kuten valintalistat ja ruudut voivat nopeuttaa tiedon syöttöä, mutta niiden liiallinen käyttö vähentää keskustelun luonnollisuutta.

Sivun keskelle sijoitettiin itse keskusteluikkuna, joka muistuttaa tavanomaista viestisovellusta. Ylä- ja alareunaan päätettiin sijoittaa yrityksen logon värisävyjä hyödyntävät

palkit, jotta sovelluksen yleisilme muistuttaa yrityksen muita verkkosivuja. Kuvassa 3 on ruutukaappaus sovelluksesta, jossa näkyy lyhyt keskustelu käyttäjän ja chatbotin välillä.



Kuva 3. Chatbot -sovelluksen käyttöliittymä.

Käyttäjän ja chatbotin viestien värit päätettiin eritellä toisistaan selkeyden vuoksi. Lisäksi jokaiseen chatbotin viestiin päätettiin liittää painike vastauskohtaiselle positiiviselle ja negatiiviselle palautteelle. Tämän lisäksi sivun alareunaan sijoitettiin painike, josta avautuu koko sovellusta koskeva palautekysely. Palaute tallennetaan tietokantaan raportoinnin ja analytiikan pohjaksi luvuissa 4.5 ja 5.5 kuvatuilla tavoilla.

Jo suunnitteluvaiheessa käyttöliittymä päätettiin toteuttaa ilman erillistä kirjastoa (kuten React, Angular tai Vue) laajentamalla IBM:n julkaisemaa avoimen lähdekoodin projektia, jossa käytetyt kielet ovat HTML5-, CSS- ja ES5-versio JavaScript-kielestä.

4.4 Keskustelun sisältö

Keskusteleavan käyttöliittymän suunnitteluperiaatteet ovat merkittävästi erilaiset kuin perinteisen graafisen käyttöliittymän kohdalla. Hyvän käyttäjäkokemuksen aikaansaaminen vaatii taitoa suunnitella tarkoituksenmukaisia *keskusteluja*. Tämä on täysin uusi elementti sovelluskehityksessä. Hyvää taustatietoa toimivan keskustelun suunnitteluun saa esimerkiksi IBM-verkkokurssista *Designing Conversational Solutions*.

Chatbotin tarkoituksena on vastata ravintolan asiakaspalveluun liittyviin kysymyksiin luvussa 4.2 esitetyn käyttötapauksen mukaisesti. Chatbotin keskustelun sisältö tulee siis rakentaa tämän kontekstin ympärille.

Keskeisimmät toiminnot tämän chatbotin kannalta ovat kysymyksiin vastaaminen, pöytävarausten tekeminen ja ruokatilausten vastaanottaminen. Näissä yhdistyy vapaa ja ohjattu keskustelu, sillä asiakas voi vapaasti esittää kysymyksiä esimerkiksi ruokalistaan ja yhteystietoihin liittyen. Ohjatut osuudet keskustelusta chatbotin kanssa liittyvät tilauksiin ja varauksiin, sillä näiden toimintojen toteuttamiseksi sovelluksen täytyy kerätä asiakkaalta lisätietoa sekä kommunikoida taustajärjestelmien kanssa.

Luvun 4.1 menetelmän mukaisesti aluksi kartoitettiin chatbotin toimintakontekstiin liittyvää termistöä. Tilaukset, varaukset ja kysymykset todettiin merkittävimmiksi tarpeiksi, joita käyttäjät chatbotille ilmaisevat. Watson Assistant -palvelussa nämä edustavat ”Intent”-aikomuksia. Ruokalistat, osoitteet ja monet ruokien ravintosisältöön liittyvät termit (esim. gluteeniton, laktoositon) ovat myös merkittävässä roolissa sovelluksen kannalta. Nämä puolestaan edustavat ”Entity”-entiteettejä Watson Assistant -palvelussa.

Botin tietopohjan lisäksi on hyvä ohjelmoida botti kykeneväiseksi niin kutsuttuun ”small talkiin”, sillä tämä lisää botin persoonallisuutta, mikä saattaa parantaa käyttäjäkokemusta.

4.5 Palautetoiminnallisuus

Chatbotin käyttäjäkokemuksen parantamisen kannalta on tärkeää, että käyttäjät voivat antaa palautetta palvelun toiminnasta. Toteutettavaan sovellukseen päätettiin tehdä palautteenantotoiminnallisuus, jonka avulla käyttäjät voivat antaa palautetta palvelusta kokonaisuutena, sekä myös yksittäisistä chatbotin antamista vastauksista.

Käyttöliittymän alareunaan sijoitettiin painike laajempaa palautetta varten ja yksittäisten vastausten palautetta varten lisättiin painikkeet sekä positiivista että negatiivista palautetta varten jokaiseen chatbotin lähettämään vastaukseen. Painike avaa käyttäjälle lyhyen kyselyn, jossa pyydetään käyttäjältä perustelu mielipiteelleen (ks. luku 5.5).

Kun käyttäjä antaa vastauskohtaista palautetta, sovellus tallentaa tietokantaan chatbotin vastauksen, käyttäjän palautteen ja keskusteluun liittyvää metadataa, joka auttaa datan jäsentelyä raportointia ja analytiikkaa varten. Myös yleinen palaute tallennetaan samaan tietokantaan. Palautteen kerääminen myös yksittäisistä vastauksista mahdollistaa helpomman laadunvalvonnan ja käyttäjätyytyväisyyden tarkkailun.

4.6 Loki

Loki auttaa keräämään botin jatkokehityksen ja ylläpidon kannalta arvokasta dataa. Sovellus kerää tietoja jokaisesta käydyistä keskustelusta tallentamalla jokaisen käyttäjän kysymyksen sekä botin vastauksen Cloudant-tietokantaan. Tämä luo perustan laadunvarmistukselle sekä teknisessä mielessä että keskustelun sisällön osalta.

Teknisessä mielessä nähdään mahdolliset Watson API sekä sovellustason virheilmoitukset. Lisäksi Assistant API palauttaa, joka kutsun yhteydessä konteksti-tietorakenteen, josta ilmenee luokittelijan tunnistama aikomus sekä varmuuskerroin, esim. 0,91. Tämä tieto on ensiarvoisen tärkeää, koska se kertoo mahdollisesta tarpeista opettaa perustuudet uudestaan. Lisäksi tietorakenne kertoo tunnistetun entiteetin, josta nähdään, mikä tunnistus toivotulla tavalla.

Yleisesti lokista nähdään kysytyjen kysymysten määrä, käyttöajankohdat jne., mikä tarjoaa arvokasta näkyvyyttä siihen, miten bottia käytetään.

Cloudant-lokin voi myös replikoida tietovarastoon, jolloin erilaiset raportointi- ja analytiikkatyökalut tarjoavat palvelun omistajalle arvokasta tietoa.

5 Chatbotin kehitys

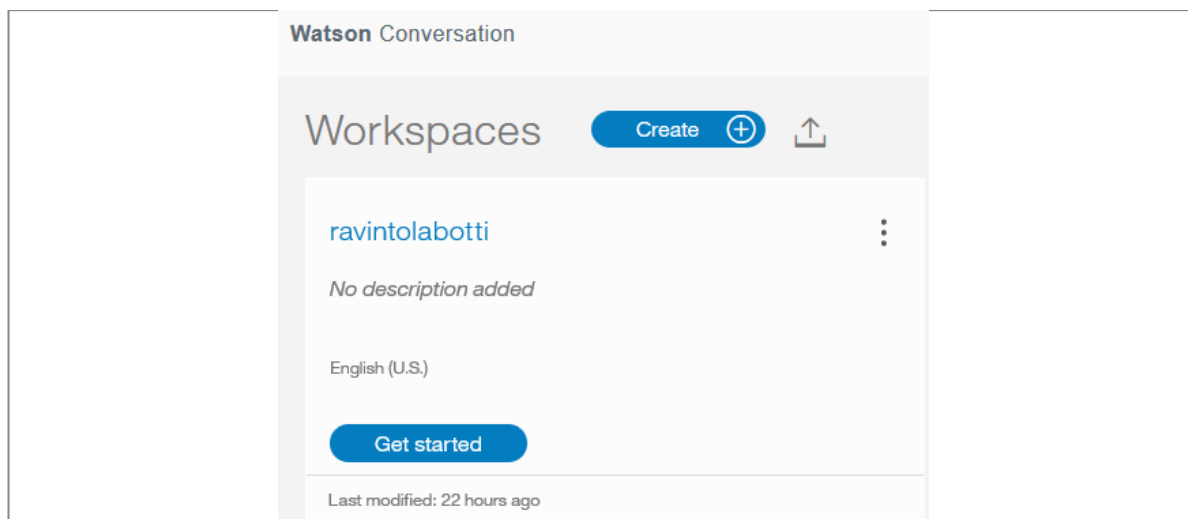
Tässä luvussa käydään läpi botin toteutukseen liittyviä seikkoja sekä erityisesti kokemuksia työn teoriaosuudessa esille nostettujen ominaisuuksien toteuttamisesta. Kehitetty sovellus on IBM Cloud -alustalla oleva Cloud Foundry -sovellus.

5.1 Watson Assistant -palvelu

Kaikki Watson Assistant -palvelussa tapahtuva kielen prosessointi on työtilakohtaista. Watson Assistant -palvelun käyttöönotto IBM Cloud -alustalla on varsin nopea prosessi. Ensin täytyy luoda instanssi Watson Assistant -palvelusta, mikä tapahtuu IBM Cloud sivuston katalogin kautta. Kun instanssi on käynnistynyt, voidaan käynnistää palvelun hallintapaneeli, josta löytyvät kaikki chatbotin rakentamiseen ja hallintaan liittyvät työkalut ja instanssikohtaiset tunnukset.

5.1.1 Työtilat

Ensimmäinen vaihe chatbotin rakentamisessa on työtilan luominen Watson Assistant -palveluinstanssiin. Työtila on kokonaisuus, joka sisältää kaikki chatbotin toiminnan määrittelevät komponentit. Yksi Watson Assistant -instanssi voi sisältää useita työtiloja ja niitä on myös mahdollista ladata ja tuoda sisään JSON -tiedostomuodossa, mikä mahdollistaa esimerkiksi kokonaisten chatbottien jakamisen muiden kehittäjien kanssa ja työtilojen ulkoisen versiohallinnan, sillä palvelu itsessään ei sisällä merkittäviä versionhallinta- tai kollaboraatiotyökaluja. Kuvassa 5. on ruutukaappaus työtilojen hallintapaneelistä, jonka kautta pääsee hallinnoimaan tunnuksia, luomaan uusia työtiloja sekä avaamaan niitä visuaalisella editorilla, jossa itse keskustelulogiikka määritetään.



Kuva 4. Workspaces-näkymä Assistant-palvelun hallintapaneelissa.

Työtila sisältää kolmentyyppisiä komponentteja, jotka ovat nimeltään ”*Intents*”, ”*Entities*” ja ”*Dialog*”.

Intent, eli aikomus kuvastaa käyttäjän kirjoittaman syötteen päämäärää tai tavoitetta, kuten esimerkiksi kysymystä yhteystiedoista. Jokaiselle chatbotin toiminnolle, jota halutaan tukea, tulee määritellä ja nimetä oma *Intent*. Käytännössä tämä tarkoittaa useiden esimerkkilauseiden- ja kysymysten sekä niiden edustamien aikomusten muodostamien parien syöttämistä palveluun, jotta luokittelija-algoritmia voidaan kouluttaa. Työkalussa aikomusten nimet alkavat aina #-merkillä.

Entity, eli entiteetti edustaa keskustelun kontekstin kannalta merkittävää olentoa, termiä tai muuta informaatiota. Jos aikomukset kuvastavat verbejä (chatbotin toimintoja), niin entiteetit kuvaavat substantiiveja (toimintojen kohteita tai kontekstia). Jos esimerkiksi käyttäjän aikomus on sääennusteen kysyminen, niin relevantteja entiteettejä olisivat ajankohta sekä sijainti, jotta sovellus voi antaa relevantin vastauksen. Entiteettien nimien tulee aina alkaa @-merkillä.

Dialog, eli keskustelu on haarautuva kaavio, joka määrittää, miten chatbot vastaa käyttäjälle perustuen luokittelija-algoritmin tunnistamiin aikomuksiin ja entiteetteihin. Työkalu sisältää visuaalisen editorin, jossa keskustelun runko ja logiikka rakennetaan määrittämällä, miten chatbot vastaa tunnistamiinsa aikomuksiin ja entiteetteihin.

Watson Assistant -palvelu käyttää näitä uniikkeja komponentteja työtilakohtaisen koneoppimismallin rakentamiseen. Malli oppii sille syötetyn esimerkkidatan perusteella tunnistamaan käyttäjän syötteistä niiden edustamia aikomuksia ja entiteettejä. Joka kerta kun esimerkkidataa päivitetään, käynnistyy mallin opetusprosessi uudestaan, jotta malli pysyy jatkuvasti ajan tasalla.

5.1.2 Watson Assistant REST API -rajapinta

Kommunikointi sovelluspalvelimen ja Watson Assistant -palvelun välillä tapahtuu Assistant REST API -rajapinnan kautta. Rajapinta tarjoaa CRUD-operaatiota työtiloille, aikomuksille, entiteeteille, esimerkkidatalle ja keskustelulogiikalle. Se tarjoaa siis ohjelmallisesti saataville samat toiminnot kuin selainpohjaisessa visuaalisessa editorissa. Kommunikointi rajapinnan kanssa tapahtuu lähettämällä autentikointi-kredentiaalit sisältäviä HTTP-kutsuja palvelun URL-osoitteeseen.

Keskeinen metodi rajapinnassa on nimeltään Message. Se ottaa parametrina käyttäjän syötteen ja työtilan tunnisteiden ja käyttää kyseisen työtilan luokittelija-algoritmia tunnistukseen siitä aikomukset ja entiteetit. Kuvassa 6 on esitettyä rajapinnan palauttamien vastausten tyypillinen rakenne.



Kuva 5. Watson Assistant -palvelun palauttamien vastausten JSON-rakenne. Tämä vastaus sisältää sekä kuvan että tekstiä.

Koska REST-rajapinnat ovat tilattomia, sovelluksen ja rajapinnan täytyy käyttää keskusteluun liittyvän tilatiedon välittämiseen muuttujaa. Watson Assistant -rajapinta käyttää tähän tarkoitukseen Context-muuttujaa, joka palautetaan tarvittaessa vastausten mukana. Sen avulla chatbot voi esimerkiksi muistaa moniportaisen kysymysketjun aiemmat vastaukset.

5.2 Front-End -ohjelmointi

Watson Assistant ei sisällä mitään käyttöliittymän kehittämiseen liittyvää toiminnallisuutta, vaan se on aina kehittäjän tehtävä muilla välineillä. Vaihtoehtona on hyödyntää valmiista viestisovellusta, kuten Slackia tai Facebook Messengeriä.

Käyttöliittymän rakentamisessa hyödynnettiin kappaleessa 4 mainitun avoimen lähdekoodin projektin sisältämiä ES5-moduuleita. Nämä moduulit tarjoavat yleishyödyllisiä funktioita esimerkiksi DOM-elementtien manipulointiin sekä selaimen ja palvelimen väliseen kommunikaatioon.

5.3 Back-End -ohjelmointi

Sovelluksen palvelin on Express-kirjastoa hyödyntäen rakennettu Node.js-prosessi, joka tarjoaa käyttöliittymälle yksinkertaisen REST API:n, jonka kautta viestien välitys Watson Assistant -palveluun tapahtuu. Palvelin ylläpitää myös sovellustason lokia (kuvattu kappaleessa 4.6) ja tallentaa käyttäjien antaman palautteen Cloudant-tietokantaan.

IBM on julkaissut Watson Assistant SDK-nimisen kirjaston, joka sisältää Assistant REST API:n metodeja vastaavat funktiot, mikä tekee sovelluskehityksestä suoraviivaisempaa. Kun tarvittavat kredentiaalit on lisätty, Assistant-rajapinnan kanssa kommunikointi tapahtuu kutsumalla sen metodeja vastaavia funktioita. Ohessa on yksinkertainen esimerkki, joka näyttää, kuinka kirjastoa käytetään.

```
var watson = require('watson-developer-cloud');

var assistant = new watson.AssistantV1({
  username: '{username}',
  password: '{password}',
  version: '2018-09-20'
});
```



```

assistant.message({
  workspace_id: '{workspace_id}',
  input: {'text': 'Löytyykö teiltä myös gluteenittomia annoksia?'},
}, function(err, response) {
  if (err)
    console.log('error:', err);
  else
    console.log(JSON.stringify(response, null, 2));
});

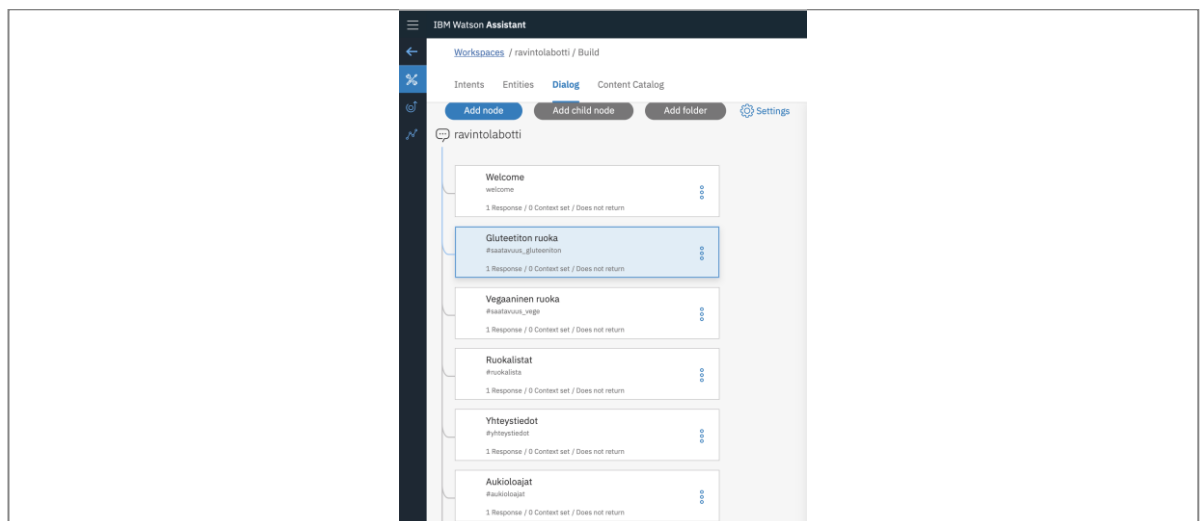
```

Esimerkkikoodi 1. Node.js-ohjelma, joka lähettää viestin Watson Assistant -palveluun

5.4 Keskustelun rungon kehitys

Keskustelun rungon kehitys tehtiin selainpohjaisella visuaalisella editorilla, joka on esitettyä kuvassa 7. Se on helppokäyttöinen, mutta keskustelun monimutkaistuessa hallintaa saattaa muuttua melko työlääksi ja monen henkilön samanaikainen, saman botin, kehitystyön hallinnointi voi olla hankalaa.

Keskustelun rungon kehitys tapahtuu käytännössä *Dialog Node* -solmujen avulla, muodostamalla keskustelun kulkua edustava vuokaaviota muistuttava puurakenne. Yksinkertaisimmillaan nämä solmut koostuvat ehdosta ja vastauksesta. Yksittäistä solmua klikkaamalla editorissa pääsee muokkaamaan kyseisen solmun ominaisuuksia. Tämä näkymä on esitettyä kuvassa 8.



Kuva 6. Ohjatun keskustelun runko Watson Assistant -palvelun Dialog-näkymässä

Ehto määrittää, mitä informaatiota käyttäjän syötteessä täytyy olla, jotta kyseinen solmu aktivoituu ja palauttaa vastauksen. Yleensä tämä informaatio on tietty *Intent* – aikomus,

mutta se voi myös olla jonkin entiteetin tyyppi tai arvo. Ehdot voivat liittyä myös luvussa 5.1.2 kuvatun kontekstimuuttujan arvoihin. Palvelu tarjoaa myös muutamia yleishyödyllisiä ehtoja, kuten ”welcome”, joka täyttyy vain ensimmäisellä keskustelukierroksella. Ehdossa on mahdollista käyttää *Regex*-lausekkeita arvojen tarkistamiseen. Ehtojen rakentamiseen voi myös käyttää *Spring Expression Language* -kieltä.

Gluteetiton ruoka Customize ✕

If bot recognizes:
#saatavuus_gluteeniton ⊖ ⊕

Then respond with: ⋮

Text Move: ^ v 🗑️

Kyllä! Ne on merkitty ruokalistoihimme "GL" merkinnällä! ⊖

Jep! Tunnistat gluteenittomat vaihtoehdot "GL" merkinnästä! ⊖

Enter response variation

Response variations are set to **sequential**. Set to [random](#) | [multiline](#) ⓘ

⊕ Add response type

And finally

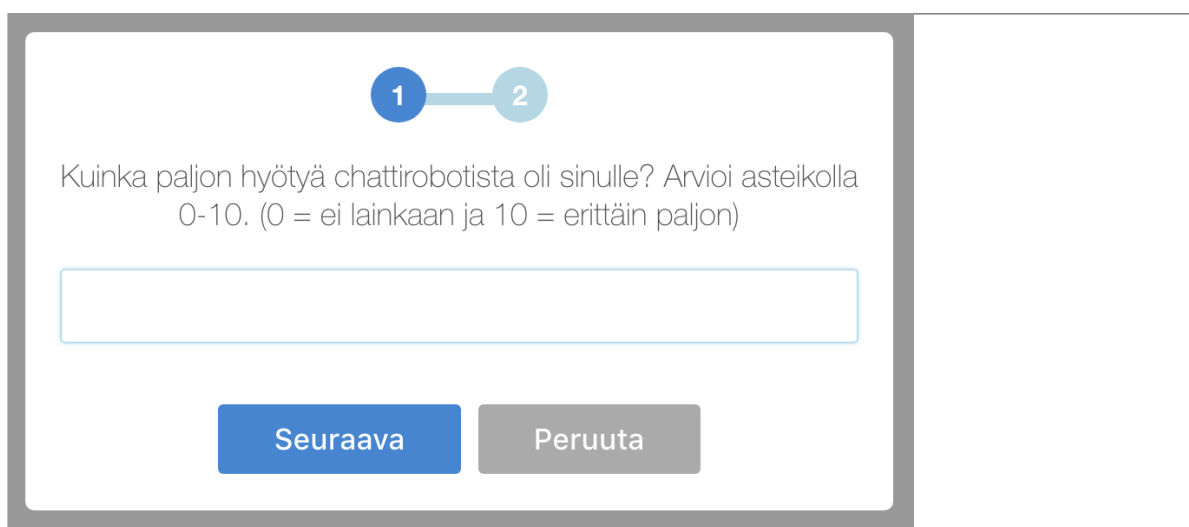
Wait for user input ▾

Kuva 7. Vastauslogiikan määrittäminen Dialog-näkymässä.

Vastaus on yksinkertaisimmillaan teksti, jonka solmu palauttaa ehtonsa täytyessä. Vastaukset voivat kuitenkin sisältää myös kuvia, listoja vaihtoehdoista tai ne voivat kutsua ulkoista koodia. Yksi solmu voi myös sisältää useita vastauksia, joista se arpoo tai valitsee yhden kerrallaan. Tätä ominaisuutta hyödyntämällä voi lisätä helposti chatbotin vastauksen monipuolisuutta, sillä se voi ilmaista saman asian useammalla tavalla.

5.5 Palauteikkunan toteutus

Chatbot-sovellukseen toteutettiin myös mekanismi, jonka avulla käyttäjät voivat antaa palautetta sekä sovelluksesta kokonaisuutena että yksittäisistä chatbotin vastauksista. Palautteenanto tapahtuu samasta näkymästä, jossa itse keskustelu käydään. Sivun alareunassa on painike, josta avautuu kuvassa 8 esitetty yleisen palautteen ikkuna. Vastauskohtaisen palautteen antamiseksi jokaiseen chatbotin viestiin liitetään käyttöliittymässä painike sekä positiiviselle, että negatiiviselle palautteelle (kuva 3).



Kuva 8. Sovelluksen palauteikkuna.

Palauteikkunan toteutuksessa päätettiin käyttää SweetAlert2-kirjastoa, jonka avulla monivaiheisen kyselyn rakentaminen on hyvin suoraviivaista. Ohessa on esimerkki, kuinka kuvan 9 ikkuna rakennetaan tätä kirjastoa käyttäen.

```
function displayServiceReview() {
  swal.defaults({
    input: 'text',
    confirmButtonText: 'Seuraava;',
    showCancelButton: true,
    cancelButtonText: 'Peruuta',
    animation: false,
    progressSteps: ['1', '2']
  });
  var steps = [
    {
      text: 'Kuinka paljon hyötyä chattirobotista oli sinulle? Arvioi asteikolla 0-10. (0 = ei lainkaan ja 10 = erittäin paljon)',
      confirmButtonText: 'Seuraava'
    },
    {
```

```

        text: 'Perustelisitko vastaustasi? Miten voisimme parantaa palvelua?
    ',
    confirmButtonText: 'Tallenna'
  }
];
swal.queue(steps).then(function (result) {
  swal.resetDefaults();
  swal({
    title: 'Kiitos!',
    confirmButtonText: 'Sulje',
    showCancelButton: false,
    type: 'success',
  })
  saveServiceReview(result);
}, function () {
  swal.resetDefaults();
})
}

```

Esimerkkikoodi 2. Palauteikkunan luonti SweetAlert2 -kirjaston avulla

Esimerkkikoodissa 2 kuvattu funktio näyttää käyttäjälle palauteikkunan ja lopuksi kutsuu funktiota, joka ottaa parametrina käyttäjän palautteen ja tallentaa sen tietokantaan.

5.6 Lokin tallennus Cloudant-tietokantaan

Cloudant on IBM Cloud -alustalla toimiva hajautettu NoSQL-tietokanta-palvelu. Sen pysyttäminen ja käyttöönotto on yksinkertaista ja helppoa. IBM Cloud -palvelun katalogin kautta voi käynnistää instanssin tietokannasta, jolloin saa käyttöön tarvittavat kredentiaalit, joiden avulla sovellus voi kommunikoida tietokannan kanssa.

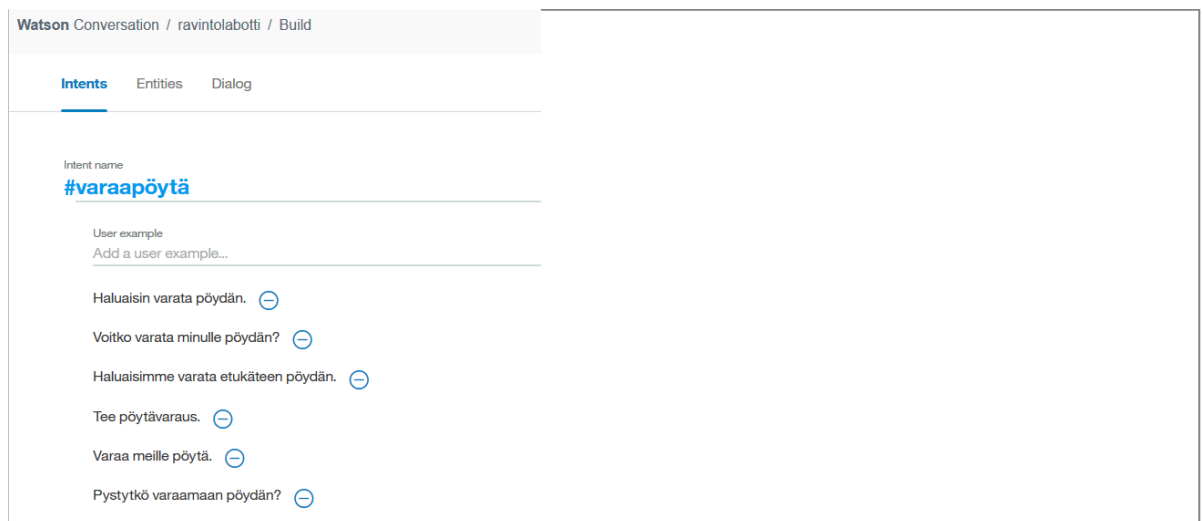
Lokien kirjoittaminen tietokantaan on varsin suoraviivaista Nano-kirjaston avulla. Aluksi tarvitsee alustaa instanssi kirjaston pääluokan oliosta antamalla parametrina tarvittavat URL-osoitteet ja kredentiaalit. Tämän jälkeen tietokantaan kirjoittaminen onnistuu kutsuamalla tämän luokan metodia, joka ottaa parametrina JavaScript-olion ja tallentaa tämän tietokantaan.

Kyselyiden ja hakujen teko NoSQL-tietokannassa on hankalampaa kuin relaatiotietokannoissa, joten keskustelulokit päätettiin replikoida *dashDB*-relaatiotietokantaan helpomman raportoinnin ja analytiikan mahdollistamiseksi. IBM Cloud tarjoaa toiminnon, jonka avulla tämän voi toteuttaa automatisoidusti. Kehittäjän tarvitsee vain ottaa tämä toiminnallisuus käyttöön Cloudant-hallintapaneelin kautta.

5.7 Tuki Suomen kielelle

Koska kielenprosessointi Watson Assistant -palvelussa perustuu syväoppimiseen, se toimii käytännössä millä kielellä tahansa, sillä se käsittelee lauseita ja sanoja ikään kuin symbolisesti. Se assosioi lauseita aikomuksiin algoritmisesti ja toimii niiden perusteella ennalta määrätyllä tavalla.

Näin ollen, kun tälle neuroverkolle syötetään tarpeeksi suomen tai minkä tahansa kielisiä esimerkkipareja lauseista ja aikomuksista/pyrkimyksistä tai entiteetistä ja synonyymeistä, niin se oppii tunnistamaan niitä uudesta tekstistä. Kuinka hyvin se oppii, riippuu pitkälti esimerkki datan määrästä ja monipuolisuudesta, sekä käytetystä laskentatehosta. Tämän ollessa pilvipalvelun tarjoama rajapinta, joten laskentateho ei sinänsä ole ongelma.



Kuva 9. Ruutukaappaus Intents -näkökulmasta, jossa voi määrittellä esimerkkilauseita.

Työn aikana todettiin, että nopein tapa toteuttaa Intent-malleja Watson Assistantiin on tuoda esimerkit CSV-muodossa sisään palveluun. Näin tehdessä voi käyttää mieleistään tekstinkäsittelyohjelmaa, ja tämän esimerkkien kasaamisen voi suorittaa myös henkilö ilman ohjelmointitaitoa. CSV-tiedostojen muodostaminen ohjelmallisesti on myös melko suoraviivaista.

6 Yhteenveto

Insinööriyössä selvitettiin laadukkaan keskustelevan käyttöliittymän ominaisuuksia, sekä kartoitettiin toimivan suomeksi keskustelevan chatbotin suunnitteluun ja toteutukseen tarvittavia vaiheita. Lisäksi tavoitteena oli selvittää, miten nykyisiä pilvipalveluita hyödyntäen pystyy rakentamaan suomea ymmärtäviä chatbotteja. Työn tavoitteena oli myös tutkia, kuinka nämä palvelut soveltuvat chatbottien luomiseen kielellä, joka ei ole virallisesti tuettuna. Watson Assistant -palvelun avulla tämä menetelmä todettiin toimivaksi.

Työn tuloksena suunniteltiin ja toteutettiin onnistuneesti kuvitteellisen ravintolan asiakaspalvelutehtävissä avustava chatbot-sovellus. Se kykenee tunnistamaan käyttäjien kysymyksistä välittyvät aikomukset yllättävän hyvin ottaen huomioon, ettei suomi ollut valitulla Watson Assistant -palvelulla tuettuna kielenä. Monimutkaisten keskustelurunkojen rakentaminen on tosin melko työlästä Watson Assistant -palvelun selainpohjaisella editorilla.

Työn tulosten perusteella voidaan todeta, että luokittelija-algoritmeihin perustuvat koneoppimismenetelmät -ja palvelut ovat hyödyllisiä keskustelevien käyttöliittymien toteutukseen myös kielillä, jotka eivät välttämättä ole virallisesti tuettuna näissä palveluissa.

Työn aikana todettuja haasteita suomenkielisen keskustelevan käyttöliittymän kehitysprosessiin liittyen ovat laadukkaan opetusdatan kerääminen ja puutteellinen kielituki nykyisissä alustoissa. Tämä tulee etenkin esiin puhetunnistukseen ja synteisiin liittyvissä seikoissa. Yleisesti ottaen näiden alustojen käyttö on suhteellisen yksinkertaista, ja niiden ominaisuudet muistuttavat toisiaan hyvin pitkälti.

Mahdollisia jatkotutkimus- ja kehityskohteita on esimerkiksi muiden työssä vertailtujen alustojen soveltuvuus suomenkielisiin chatbotteihin, puhetunnistus ja synteesi suomeksi sekä opetusdatan keräämiseen liittyvän metodologian kehittäminen ja optimointi. Myös paremman käyttöliittymän suunnittelu, keskustelulogiikan- ja runkojen rakentamiselle, olisi hyvä jatkokehitysmahdollisuus.

Lähteet

- 1 Speeding Customer Service by 99% with Watson. 2018. Verkkoaineisto. IBM. [<https://www.ibm.com/watson/autodesk/>]. Luettu 10.11.2018.
- 2 UBank. 2017. Verkkoartikkeli. IBM. [<https://www.ibm.com/case-studies/ubank>]. Luettu 10.11.2018.
- 3 Chatbot Emma eli IFin asiakaspalvelurobotti. 2018. Verkkoartikkeli. If Oyj. [<https://www.if.fi/henkiloasiakkaat/asiakaspalvelu/verkkopalvelut/chatbot-emma>]. Luettu 10.11.2018.
- 4 Kela tarjoaa uudenlaisen palvelun opiskelijoille. 2017. Verkkoartikkeli. Kansaneläkelaitos. [https://www.kela.fi/ajankohtaista-opiskelijat/-/asset_publisher/I7X3vuEkReGH/content/kela-tarjoaa-uudenlaisen-palvelun-opiskelijoille]. Luettu 10.11.2018.
- 5 Chatbot Market to grow at 31 Percent CAGR from 2018 to 2024. 2018. Verkkoartikkeli. SDCExec. [<https://www.sdcexec.com/software-technology/news/21011880/chatbot-market-to-grow-at-31-percent-cagr-from-2018-to-2024>]. Luettu 10.11.2018.
- 6 Maida, Jesse. 2018. Global Chatbot Market 2018-2022 to Post a CAGR of Over 24%. Verkkoartikkeli. Technavio Research. [<https://www.businesswire.com/news/home/20180615005428/en/Global-Chatbot-Market-2018-2022-Post-CAGR-24>]. Luettu 10.11.2018.
- 7 What is a Chatbot. 2018. Verkkoaineisto. Investopedia. [<https://www.investopedia.com/terms/c/chatbot.asp>]. Luettu 10.11.2018.
- 8 Turing, Alan. 1950. COMPUTING MACHINERY AND INTELLIGENCE. Artikkel. Mind. [<https://www.csee.umbc.edu/courses/471/papers/turing.pdf>].
- 9 Wardip-Fruin, Noah. 1976. Expressive Processing: Digital Fictions, Computer Games, and Software Studies. Cambridge: The MIT Press.
- 10 Messaging apps are now bigger than social networks. 2016. Verkkoartikkeli. Insider Inc. [<http://www.businessinsider.com/the-messaging-app-report-2015-11?IR=T>]. Luettu 10.11.2018.
- 11 Dale, Robert. 2016. The return of the chatbots. Natural Language Engineering, Cambridge University Press. [<https://www.cambridge.org/core/journals/natural-language-engineering/article/return-of-the-chatbots/0ACB73CB66134BFCA8C1D55D20BE6392/core-reader>].

- 12 Sparks, Sarah D. 2017. How 'Intelligent' Tutors Could Transform Teaching. Verkkoartikkeli. [<https://www.edweek.org/ew/articles/2017/09/27/how-intelligent-tutors-could-transform-teaching.html>]. Luettu 10.11.2018.
- 13 IBM Watson Education and Pearson to Drive Cognitive Learning Experiences for College Students. 2016. Verkkoartikkeli. Pearson. [<https://www.pearson.com/corporate/news/media/news-announcements/2016/10/ibm-watson-education-and-pearson-to-drive-cognitive-learning-exp.html>]. Luettu 10.11.2018.
- 14 Strauss, Mark. 2014. The U.S. Army Is Using a Chatbot to Recruit New Soldiers. Verkkoartikkeli. [<https://io9.gizmodo.com/the-u-s-army-is-using-a-chatbot-to-recruit-new-soldier-1564813737>]. Luettu 10.11.2018.
- 15 Solomon, Shoshanna. 2017. Techie soldiers program chatbots for the military in IDF hackathon. Verkkoartikkeli. [<https://www.timesofisrael.com/techie-soldiers-program-chatbots-for-the-military-in-idf-hackathon/>]. Luettu 10.11.2018.
- 16 Button, Robert W. 2017. Artificial Intelligence and the Military. Verkkoartikkeli. [<https://www.rand.org/blog/2017/09/artificial-intelligence-and-the-military.html>]. Luettu 10.11.2018.
- 17 Maass, David. 2014. Everything We Know About the Army's Uncanny Chatbots. Verkkoartikkeli. [<https://gizmodo.com/everything-we-know-about-the-armys-uncanny-chatbots-1564766836>]. Luettu 10.11.2018.
- 18 Cognitive conversation for creating conversations between apps and users. 2018. Verkkoaineisto. IBM. [<https://www.ibm.com/cloud/garage/architectures/cognitiveConversationDomain/cognitive-planning>]. Luettu 10.11.2018.

