

Jere Ojala

Puhekäyttöliittymien käytettävyyden suunnittelu

Tietotekniikan kandidaatintutkielma

29. huhtikuuta 2016

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Jere Ojala

Yhteystiedot: jejoojal@student.jyu.fi

Työn nimi: Puhekäyttöliittymien käytettävyyden suunnittelu

Title in English: Voice User Interface Usability Design

Työ: Kandidaatintutkielma

Sivumäärä: 24+0

Tiivistelmä: Puhekäyttöliittymät ovat puheella toimivia käyttöliittymiä. Kuten kaikki käyttöliittymät, puhekäyttöliittymien suunnittelu vaatii huolellisuutta, jotta se olisi mahdollisimman helppokäyttöinen. Käytettävyyden voi määritellä opettavuudella, tehokkuudella, muistettavuudella, virheillä ja tyytyväisyydellä. Eri tekniikat puhekäyttöliittymän toteutukselle ovat täsmällisen ja epätäsmällisen komentokehöitteen välimuotoja.

Avainsanat: puhekäyttöliittymä, käytettävyys, käyttöliittymä

Abstract: Speech user interfaces are user interfaces that operates with speech interaction. Just like all user interfaces, design of speech user interfaces requires careful consideration to be as user-friendly as possible. Usability can be defined with learnability, efficiency, memorability, errors and satisfaction. Different techniques for execution of speech user interface are different intermediate forms of the implicit and explicit prompt.

Keywords: speech user interface, voice user interface, usability, user interface

Kuviot

Kuvio 1. Käyttöliittymätyyppien vertailu, Munteanu, Cosmin, Penn, Gerald (2011)	2
Kuvio 2. Jakob Nielsenin mukainen käytettävyyden määritelmä.....	3
Kuvio 3. Jakob Nielsenin käyttäjäkeskitteiset oppimiskäyrät	4

Sisältö

1	JOHDANTO	1
2	KÄYTETTÄVYYDEN SUUNNITTELU	3
2.1	Opittavuus (Learnability)	3
2.2	Tehokkuus (Efficiency)	4
2.3	Muistettavuus (Memorability)	5
2.4	Virheet (Errors)	5
2.5	Tyytyväisyys (Satisfaction).....	7
2.6	10 käytettävyyden heuristiikkaa	7
3	PUHEKÄYTTÖLIITTYMÄN SUUNNITTELUN TEKNIIKAT	10
3.1	Täsmällinen komentokehoite (Explicit Prompt)	10
3.2	Epätäsmällinen komentokehoite (Implicit Prompt).....	11
3.3	Inkrementaalinen ja laajeneva komentokehoite (Incremental and Ex- panded Prompt)	11
3.4	Kapeneva komentokehoite (Tapering Prompt)	12
3.5	Vihjeet (Hints).....	13
3.6	Multimodaalisten systeemien tekniikat	14
4	TEKNIIKOIDEN KÄYTETTÄVYYS	15
5	YHTEENVETO	19
	KIRJALLISUUTTA	20

1 Johdanto

Puhe on ihmisen luonnollisin kommunikointityyli ja puhekäyttöliittymiä on valmistettu pitkän aikaa, mutta siitä huolimatta puhekäyttöliittymät ovat suhteellisen harvaksen käytössä, tai ainakaan yleisessä jokapäiväisessä käytössä. Osasyynä saattaa olla puheentunnistuksen yleistymisen puute, mutta käyttöliittymän käytettävyys on tärkeä ottaa huomioon alan kehittymisen kannalta. Puhekäyttöliittymät ovat käyttöliittymiä, joissa yleisen graafisen näytön ja näppäinten sijaan käytetään joko puhetta tai puheentunnistusta, tai yleensä kumpaakin.




Puhekäyttöliittymät sisältävät muihin käyttöliittymiin verrattuna monia etuja: CPM (Characters per minute) on parempi, laitteisto vaatii ainoastaan mikrofonin, joka on pieni ja vaatii vähemmän tilaa, sekä puhekäyttöliittymien käyttö ei vaadi käsiä eikä silmiä. Puhekäyttöliittymissä on myös kuitenkin omat haittansa muihin nähden, kuten puheentunnistuksen haasteet ja monimutkaisuus, sekä epäkäytännöllisyys meluisalla ja julkisella paikalla Munteanu, Cosmin, Penn, Gerald (2011)

Kuten kaikkien käyttöliittymien, myös puhekäyttöliittymien suunnittelu on haasteellista ja vaatii tarkkaa työtä, eikä niiden suunnittelun ongelmiin ole myöskään yhtä ainoaa ratkaisua. Suunnittelu riippuu erittäin paljon siitä millä laitteella tai alustalla kyseinen käyttöliittymä on, sekä myös mikä on niiden käyttötarkoitus. Osa laitteista pystyy hyödyntämään graafisen ja puhekäyttöliittymien yhdistelmää hyödyntäen kummankin hyviä puolia riippuen siitä, miten sovellus on suunniteltu käytettäväksi.

Yleisin ongelma puhekäyttöliittymän suunnittelussa on käyttäjän rajoittaminen puheentunnistuksen suhteen, mutta kuitenkin antaen käyttäjälle vapauden puheen luonnollisuuteen. Nämä kaksi haastetta ovat usein ristiriidassa keskenään (Hansen, B., Novick, D., and Sutton, S. 1996). Toisaalta käyttäjä haluaisi puhua tavalliseen tapaan käyttöliittymän kanssa miettimättä mitä voi sanoa, mutta mitä enemmän vapauksia käyttöliittymä antaa käyttäjälle, sitä vapaamaksi käyttäjä luulee käyttöliittymän käytettäväksi. Toisaalta täsmällinen käyttöliittymä saattaa käydä turhautta-

vaksi käyttää sekä saattaa osoittautua epäkäytännölliseksi pitemmän päälle, varsinkin jos vaihtoehtojen määrä on suuri.

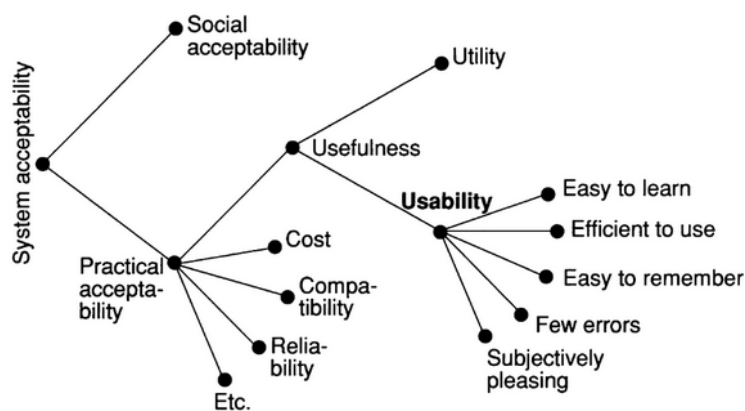
Tässä kirjallisuuskatsauksessa tarkoituksena miettiä, mistä käytännöllisyys koostuu, sekä eri suunnittelumalleja puhekäyttöliittymien suunnitteluun. Luvussa kaksi katsotaan käytettävyyttä, kolmannessa eri tekniikoita, ja neljännessä luvussa pohditaan näiden tekniikoiden käytettävyyttä toisen luvun kriteereillä.

Mode	CPM	Reliability	Devices	Practice	Other tasks
Handwriting 	200-500	recognition errors	tabloid, scanner BIG	No (requires literacy)	hands and eyes busy
Typing 	200-1000	~ 100% (typos)	keyboard BIG	yes, if high bdwidth	hands and eyes busy
Speech 	1000-4000	recognition errors	micro SMALL	no	hands and eyes free

Kuvio 1. Käyttöliittymätyyppien vertailu, Munteanu, Cosmin, Penn, Gerald (2011)

2 Käytettävyyden Suunnittelu

Käytettävyys on laaja käsite, joka koostuu monista vaikuttavista tekijöistä. Liittyen eri systeemien käytettävyydelle, mukaan lukien käyttöliittymöiden, on olemassa viisi ominaisuutta, jotka ovat opittavuus, tehokkuus, muistettavuus, virheet, tyytyväisyys (Nielsen, Jakob 1993, ss. 26–37). Näiden ominaisuuksien lisäksi on vielä 10 käyttöliittymän suunnitteluun tarkoitettua heuristiikkaa (Nielsen, Jakob 1995).

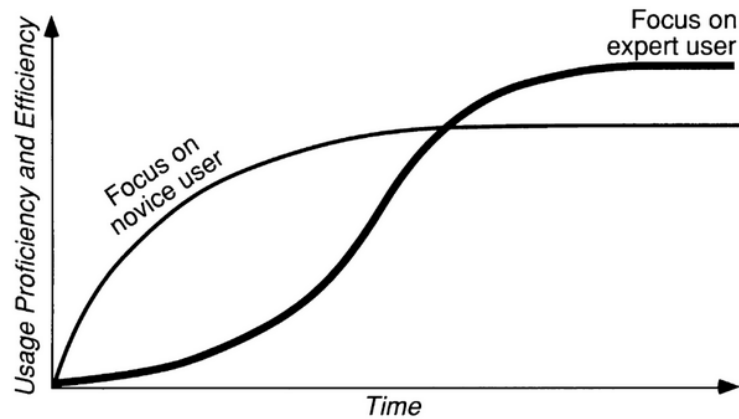


Kuvio 2. Jakob Nielsenin mukainen käytettävyyden määritelmä

2.1 Opittavuus (Learnability)

Opittavuudella tarkoitetaan kuinka helposti käyttöliittymää oppii käyttämään. Mitä nopeammin liittymää oppii käyttämään, sitä mielekkäämpää sen käyttäminen on. Kun käyttöliittymä on käyttäjätavallisesti opittava, sen täydellisen osaamisen oppiminen nopeutuu, jolloin tuottavuus ja tehokkuus on korkeimmillaan.

Kuvasta 3 näemme kuinka aloittelijan käyttöön keskittynyt käyttöliittymän tehokkuuden kasvu kasvaa erittäin nopeasti kohti maksimitehokkuutta ja tasoittuu maksimitehokkuutta lähestyessä, kun taas ammattitaitoiseen käyttöön tarkoitettu kasvaa eksponentiaalisesti. Kuitenkin ammattitaitoiseen käyttöön tehdyllä käyttöliittymällä on hieman korkeampi maksimitehokkuus (Nielsen, Jakob 1993, ss. 26).



Kuvio 3. Jakob Nielsenin käyttäjäkeskitteiset oppimiskäyrät

2.2 Tehokkuus (Efficiency)

Tehokkuudella tarkoitetaan käyttöliittymän tasaista tuottavuutta. Tällä tarkoitetaan kuvan 3 käyrän kohtaa, jolloin käyrä on mahdollisimman tasoittunut. Kuitenkin tulisi ottaa huomioon se, että tasaiseen tuottavuuden tilaan pääsemiseen saattaa kulua aikaa pitempiäkin aikoja monitasoisemmissa systeemeissä. Joskus käyttäjät myös saattavat opetella vain muutaman korkeamman käyttötason ominaisuuden ja tyytyvät niihin, mikä saattaa osoittautua vähemmän tehokkaammaksi ajankäytön suhteen kuin enempien ominaisuuksien opettelu.

Tehokkuuden mittaaminen vaatii kokeneita käyttäjiä. Tosin mikä lasketaan koke-museksi saattaa olla hiukan epämääräisesti arvioitu, esimerkiksi kuinka kauan käyt-täjä on työskennellyt systeemin parissa. Uudelle systeemille, jolle ei löydy pitkäai-kaisia käyttäjiä, kokeneiden käyttäjien saanti voi vaatia jonkin sortin testikäytön, jossa testikäyttäjät käyttävät systeemiä tietyn ajan, jonka jälkeen tehokkuus mita-taan. Toinen keino mitata uuden systeemin tehokkuutta on itse oppimiskäyriä tut-kimalla. Tehokkuutta tutkitaan esimerkiksi yhden operaation suoritukseen, ja kun suoritus ei enää nopeudu tarpeeksi, tästä mitataan tehokkuus. Tehokkuuden mit-taus näin ollen vaatii jonkin sortin määrtielmän asiantuntemukselle, josta päätellään tehokkuus (Nielsen, Jakob 1993, ss. 30–31).

2.3 Muistettavuus (Memorability)

Muistettavuus on tärkeää satunnaisille käyttäjille, jotka ovat kolmas ryhmä aloittelevien ja asiantuntijoiden lisäksi. Muistettavuus tarkoittaa sitä, että käyttäjä pystyy jatkamaan yhtä tehokasta tuottamista mihin jäi pitempää taukoa ennen. Satunnaisten käyttäjien ja aloittelevien käyttäjien suurin ero on siis satunnaiset käyttäjät ovat jo käyttäneet liittymää pitemmänkin aikaa. Satunnaisten käyttäjien lisäksi muistettavuus on tärkeää muillekin käyttäjille, jotka saattavat pitää pitemmän tauon käytöstä syystä tai toisesta.

Pitää myös muistaa, että opittavuus ja muistettavuus eivät ole sama asia eivätkä myöskään välttämättä korreloi, vaikka näin voisi luulla. Asiat voivat olla vaikeasti opittavia, mutta silti muistettavia. Jakob Nielsen (1993) antaa kirjassaan esimerkiksi tästä Washington DC:n kyltin, jonka merkitys ei ole ilmiselvä ensimmäistä kertaa sen näkeväälle, mutta muistaa sen helposti tästä eteenpäin (Nielsen, Jakob 1993, ss. 31).

2.4 Virheet (Errors)

Käyttöliittymän tulisi aiheuttaa mahdollisimman vähän haitallisia virheitä, eli taphtumia, joissa haluttua tavoitetta ei saavutetaankaan. Osa virheistä ei kuitenkaan välttämättä lasketa tähän virheeksi vaan tehokkuuden laskuksi, jos virhe ainoastaan hidastaa käyttäjää, eikä erikoisemmin estä toiminnallisuuksia.

Muunlaiset virheet ovat taas vahingollisia luonteeltaan. Tämän tyyllisiä virheitä on kahdenlaisia: Virheet, joita käyttäjät eivät pysty itse huomaamaan ja aiheuttaa näin virheellistä tulosta, ja sitten virheet, jotka tuhoavat käyttäjän työn, ja mahdollisesti tehden sen korjaamattomaksi. Virheet tulisi luokitella siis erikseen pieniin haitallisiin virheisiin ja isoihin vahingollisiin virheisiin (Nielsen, Jakob 1993, ss. 31).

Systeemin virheiden lisäksi on myös käyttäjistä johtuvat virheet. Käyttöliittymää suunniteltaessa nämä tulisi huomioida ja välttää systeemin taipuvaisuutta näille virheille, eikä syyttää käyttäjää osaamattomuudesta. Nämä virheet ovat suunnittelijan

vastuulla.

Käyttäjän virheet jaetaan kahteen luokkaan: lipsahdukset ja erehdykset. *Lipsahdukset* ovat virheitä, joissa käyttäjällä on tarkoitus suorittaa muu toiminta kuin mitä oikeasti tarkoitus suorittaa. Näppäimistöllä esimerkiksi viereisen näppäimen painallus lasketaan tällaiseksi. Yleensä lipsahdukset tulee, kun käyttäjä tekee asioita, jotka eivät vaadi tältä keskittymistä. Lipsahduksia voidaan välttää parilla keinolla. Ensimmäinen keino on asettaa käyttäjälle järkeviä rajoituksia, jotta käyttäjät eivät voi tehdä mitään, mikä ei ole mahdollista. Esimerkiksi ohjelma antaa tilata lentokoneelle menopäiväksi tulopäivän jälkeinen päivä, mikä ei tietenkään ole mahdollista todellisessa maailmassa. Toinen keino on antaa käyttäjälle ehdotuksia, aivan kuin nykyisin Google-hakukoneessa. Kolmanneksi systeemissä tulisi olla hyvät oletusvaihtoehdot. Kun käyttäjällä on toistuvia tai tarkkuutta vaativia toimintoja, on hyvä jos systeemin oletusvaihtoehdot ovat loogisia. Esimerkiksi uunin ajastimen oletusvaihtoehto on tunti, sen sijaan että käyttäjä naputtelee manuaalisesti huomisen päivän. Lopuksi systeemin formaattien tulisi olla joustavampia. Esimerkiksi puhelinnumeron asettamiseksi ei tarvitsisi laittaa sitä tietyssä muodossa, vaan käyttäjä voisi sen sijaan kirjoittaa sen vapaassa, helposti tarkistettavassa muodossa (Laubheimer, Page 2015).

Erehdykset tapahtuvat, kun käyttäjän yritykset saavuttaa tavoitteensa väärillä keinoilla. Tämä tarkoittaa sitä, että käyttäjä onnistuu siinä, mitä yrittää tehdä, mutta tämä ei tuota haluttua tulosta, koska toiminto itsessään on väärä (Laubheimer, Page 2015). Tällaiset erehdykset tapahtuu yleensä, koska suunnittelijalla on erilaiset mielikuvat tai mentaaliset mallit siitä, kuinka systeemin tulisi toimia. Suunnittelijan tulisi tietää, että käyttäjä ei todennäköisesti ikinä tule oppimaan suunnittelijan omaa mentaalista mallia. Suunnittelijan tulisi osata kerätä itse tietoa, millaisia mentaalisia malleja käyttäjillä on. Systeemiä suunnitellessa kannattaa myös muistaa käyttää yleisiä käytänteitä, sillä käyttäjät aina käyttävät enemmän muita systeemejä. Toinen tärkeä keino välttää erehdyksiä on tehdä toiminnoista itseään ilmaisevia, eli toiminnon tulisi kommunikoida kuinka sitä käytetään. Esimerkiksi painettaviin painikkeisiin yleensä lisätään varjo, jotta se näyttäisi kohoavan ruudulta, mikä näyttää

että sitä voisi painaa alas. Kolmas keino on näyttää käyttäjälle ennakkoon, mitä toiminto tulee tekemään. Joskus käyttäjä ei tiedä, että heidän tekemänsä toiminto tulee vaikuttamaan suuremmalla asteikolla systeemiin. Tällöin olisi hyvä näyttää, minkälainen muutos tulee systeemiin ennen kuin he hyväksyvät sen. Tämä on erityisesti tärkeää kuvanmuokkausohjelmissa, joissa muokkauksen renderöinti saattaa kestää tovin (Laubheimer, Page 2015).

2.5 Tyytyväisyys (Satisfaction)

Tyytyväisyys on tarkoittaa systeemin käytettävyyden mukavuutta. Tyytyväisyys on kuitenkin subjektiivinen ominaisuus. Yleisesti tyytyväisyys on tärkeä enimmäkseen järjestelmille, jotka ovat luotu viihdetarkoitukseen työtarkoituksen sijaan. Tällaisille systeemeille tyytyväisyyteen panostaminen on usein tärkeämpää kuin nopeuteen. Systeemin tyytyväisyys yleensä mitataan käyttäjäkyselyillä.

2.6 10 käytettävyyden heuristiikkaa

Jakob Nielsen loi vielä 10 heuristiikkaa liittyen erityisesti käyttöliittymien suunnitteluun. Nämä ovat erittäin laajoja käsitteitä (Nielsen, Jakob 1995). Osa kyseisistä heuristiikoista ovat laadittu graafinen käyttöliittymä mielessä, joten tässä alaluvussa myös mietitään, miten näitä heuristiikkoja voisi toteuttaa puhekkäyttöliittymässä.

Statuksen näkyvyys (Visibility of system status): Käyttäjän tulisi olla tietoinen koko ajan käyttöliittymän nykyisestä tilasta. Käyttöliittymän täytyy antaa käyttäjälle tarvittava palautetta mitä on tapahtumassa tai mitä tulee tapahtumaan. Puhekkäyttöliittymässä statusta ei näy, ellei puhekkäyttöliittymä ole luotu näytölliselle laitteelle, joten tilan tietoisuus saattaa olla hankala lisätä, ei kuitenkaan mahdotonta.

Systeemin ja todellisen maailman yhteys (Match between system and the real world): Systeemin täytyy osata puhua käyttäjän kieltä, sanoilla tai ymmärrettäville käsitteillä, sen sijaan että käyttäisi omia termejään ja kieltä. Systeemin ymmärrettävyys tulisi olla luonnollinen käyttäjälle. Vaikka tässä tarkoitetaan kielellä sen kokonaisval-

taisempaa merkitystä, eri maiden kielten osaaminen on harkittava asia käyttöliittymää suunnitellessa. Puhekäyttöliittymä on erittäin rajoittunut, jos se tunnistaa vain paria kieltä. Vaikka käyttäjä osaisikin eri kieltä, puheentunnistus ei välttämättä ymmärrä aksentin takia. Jopa saman kielen aksentit saattavat koitua ongelmalliseksi puheentunnistuksen kannalta (Price, David 2013).

Käyttäjän hallinta ja vapaus (User control and freedom): Käyttäjät tekevät virheitä ja heillä tulisi olla selvä keino peruuttaa virheensä. Esimerkkinä ovat "tee uudestaan" ja "peruuta"-toiminnot. Riippuen toiminnosta, käyttäjän hallinta puhekäyttöliittymään voidaan helposti lisätä.

Yhdenmukaisuus ja normit (Consistency and standards): Kaikkien sanojen ja toimintojen tulisi olla yhdenmukaisia, ettei muodostu sekaannusta sanoissa ja toiminnoissa. Toinen tärkeä asia on ottaa huomioon myös yleiset tavat. Käyttöliittymän tulisi käyttää yleiseen käytäntöön tulleita konsepteja. Esimerkkinä yleisestä käytännöstä on tallennusikonina perinteinen levyke, joka pitkän käytön ajan on tullut kaikille tutuksi tässä yhteydessä. Tällaiset yleiset tavat ovat usein tärkeämpiä kuin uudet kehitelmät (Nielsen, Jakob 1995). Puhekäyttöliittymissäkään ei saisi olla eri muotoja vaativia kysymyksiä, jos käyttöliittymä vaatii täsmällistä syötettä. Esimerkiksi ei saisi etu- ja sukunimen pyytämässä niiden järjestys vaihtua.

Ongelmien ehkäisy (Error prevention): Systemin virheet pitäisi ennaltaehkäistä suunnitteluvaiheessa. Pelkkien virheilmoitusten lisääminen systemiin ei ole ihanteellista ennaltaehkäisyyn verrattuna. Virheitä käytiin jo läpi luvussa 2.4.

Tunnistaminen muistamisen sijaan (Recognition rather than recall): Kaikki tehdyt toiminnot tulisi olla huomattavia, sen sijaan että käyttäjä joutuisi itse muistamaan mitä tuli tehtyä. Systemin käyttömanuaalikin tulisi olla saatavilla kaikissa tilanteissa. Tunnistaminen on sinällään vaikeaa puhekäyttöliittymissä, joissa ei ole graafista näyttöä olemassa. Toisaalta osassa tekniikassa käyttöliittymä osaa neuvoa käyttäjää, jos tämä pysyy hiljaa.

Joustavuus ja käytön tehokkuus (Flexibility and efficiency of use): Käyttöliittymän pitäisi olla muokattavissa kokeneelle käyttäjälle sopivaksi ja käytännölliseksi, ilman et-

tä se haittaa vähemmän kokemattomia käyttäjiä. Varsinkin kaikkein käytetyimmät toiminnot tulisi olla helposti muokattavissa. Henkilökohtaiseen käyttöön tarkoitettuissa puhekäyttöliittymissä pitäisi olla mahdollisuus muokata sen dialogia.

Estetiikka ja minimalistinen suunnittelu (Aesthetic and minimalist design): Dialogi tulisi olla mahdollisimman suoranaista eikä sisältää liikaa turhaa ja tarpeetonta tietoa. Dialogilla tarkoitetaan alunperin mitä tahansa vuorovaikutusta, mutta tämä on erittäin olennaista puhekäyttöliittymässä.

Auta käyttäjää tunnistamaan, diagnosoimaan ja korjaamaan virheet (Help users recognize, diagnose, and recover from errors): Virheilmoitukset pitää olla ymmärrettäviä, tarkkoja ja ymmärrettävällä kielellä, jonka aloittelevatkin käyttäjät voivat ymmärtää. Puhekäyttöliittymien tulisi osata korjata käyttäjää sanomaan ymmärrettävät dialogivaihtoehdot, ja antamaan palautetta hiljaisuuden sijaan.

Apu ja dokumentointi (Help and documentation): Tarvittaessa systeemin dokumentointi ja lisäavustus pitäisi olla saatavilla käyttäjälle. Kuitenkin systeemi tulisi luoda niin, ettei dokumentointia tarvita käyttämiseen. Dokumentointi olisi hyvä olla ymmärrettävää ja mahdollisimman vähäsanaista uhraamatta tarvittavaa tietoa. Puhekäyttöliittymiinkin voi saada luotua dokumentoinnin saataville puheella, jolloin vähäsanaisuus ja ymmärrettävyys on erittäin tärkeää.

3 Puhekäyttöliittymän suunnittelun tekniikat

Kaksi yleisintä tekniikkaa ovat täsmällinen (explicit) ja epätäsmällinen (implicit) komentokehote, minkä variaatioita ja yhdistelmiä monet loput tekniikat ovat. Tässä luvussa käsitellyt tekniikat ovat esiteltty Nicole Yankelovichin kirjassa *How Do Users Know What To Say* (Yankelovich, Nicole 1996).

3.1 Täsmällinen komentokehoite (Explicit Prompt)

Täsmällisessä komentokehoteessa käyttöliittymä antaa suoraan kaikki vaihtoehdot, mitä käyttäjä voi ehdottaa sille. Tällöin ei jää paljon varaa käyttää termejä, joita puheentunnistus ei ymmärrä, mikä vähentää huomattavasti virheiden määrää ja tekee tästä luotettavan. Käyttäjälle ei myöskään tule ongelmaa, mitä ei saa sanoa puhekäyttöliittymälle.

Kuitenkin täsmällisen tekniikan ongelma on sen epäkäytännöllisyys. Vaihtoehtojen kuunteleminen vie aikaa, varsinkin jos niiden määrä on suuri. Määrän suuruuden takia käyttäjä saattaa unohtaa helposti, mitä vaihtoehtoja oli mennyt. Tämän takia täsmällinen komentokehoite toimii parhaiten erittäin lyhyissä vaihtoehtolistoissa.

Esimerkki täsmällisestä komentokehoteesta (Yankelovich, Nicole 1996, ss. 37):

Tietokone: Valitse haluamanne artistin nimi?

Käyttäjä: The Beatles

Tietokone: Beatlesiltä löytyy monta albumia. Sano haluamanne albumi. Esimerkiksi voitte sanoa "Magical Mystery Tour" tai "Abbey Road". Jos haluatte koko listan Beatlesin albumeita, sano lista.

Käyttäjä: Lista.

Tietokone: Tässä on kaikki mitä Beatlesiltä löytyy. Sano, kun oikea tulee kohdalle. "With the Beatles." "Beatles For Sale." "Revolver." "The White Album." "Please Please Me.."

Käyttäjä(keskeyttää): "The White Album".

Tietokone: Kuulin, että sanoit "The White Album". Onko tämä totta?

Käyttäjä: Kyllä

Tietokone: Selvä. Otetaan "The White Album"...

3.2 Epätäsmällinen komentokehoite (Implicit Prompt)

Epätäsmällinen komentokehote on taas päinvastainen täsmälliseen verrattuna. Käyttäjä voi aloittaa suoraan antamalla vapaasti komentoja, ja käyttöliittymä osaa tulkita ne. Jos käyttäjä ei anna kaikkea tietoa, käyttöliittymä pyytää täydentämään puuttuvat tiedot. Muita pakollisia vaiheita ei tule.

Luonnollisen käyttämisen ja normaalin ihmisen kaltaisen vuorovaikutuksen takia epätäsmällinen käyttöliittymämalli on helpohko ja mukava käyttöinen. Tosin tämän tekniikan suurin ongelma on sen taipuvuus virheille, sillä käyttäjä saattaa luulla käyttöliittymän ymmärtävän paremmin mitä se oikeasti ymmärtää.

Esimerkki epätäsmällisestä komentokehotteesta (Yankelovich, Nicole 1996, ss. 38):

Käyttäjä: Haluaisin lentää Bostonista Lontooseen.

Tietokone: Minä päivänä haluaisit matkustaa?

Käyttäjä: Ensi perjantaina.

Tietokone: Tässä on lennot Bostonista Lontooseen perjantai, Toukokuun 31....

3.3 Inkrementaalinen ja laajeneva komentokehoite (Incremental and Expanded Prompt)

Inkrementaalinen komentokehote on muunnos epätäsmällisestä komentokehotteesta. Inkrementaalisella tekniikalla toteuttu käyttöliittymä aloittaa dialogin pyytämällä käyttäjältä syötettä, mutta liittymän kysymys on muodostettu vapaaseen ja epä-

täsmälliseen tyyliin. Esimerkiksi käyttöliittymä voisi kysyä: ”Mitä haluat tehdä?”, mikä on erittäin monitahoinen kysymys, johon ei ole täsmällistä vastausta. Kuitenkin, jos käyttäjä ei tiedä mitä voisi sanoa liittymälle ja pysyy hiljaa, inkrementaalinen käyttöliittymä alkaa täsmentämään kysymystä luettelemalla sallitut dialogivaihtoehdot.

Jos sallittuja vaihtoehtoja on liikaa liittymällä, käyttöliittymä antaa käyttäjälle vain oleellisimmat, tai suuntaa antavat vaihtoehdot. Esimerkiksi jos käyttöliittymällä on puhelinluettelollinen lista valittavia nimiä, inkrementaalinen käyttöliittymä pyytää käyttäjää sanomaan, missä muodossa käyttäjän tulisi lausua haluttu nimi sen sijaan, että listaisi kaikki nimet.

Esimerkki inkrementaalisesta komentokehotteesta (Yankelovich, Nicole 1996, ss. 40):

Tietokone: Tervetuloa ABC-pankkiin! Mitä haluaisit tehdä?

Käyttäjä: (hiljaisuus)

Tietokone: Voit tarkistaa tilin, tehdä tilisiirron, tai maksaa laskun.

Käyttäjä: (hiljaisuus)

Tietokone: Sano joku seuraavista: tarkista tili, tee tilisiirto, tai maksa lasku.

3.4 Kapeneva komentokehoite (Tapering Prompt)

Kapeneva komentokehotteen tarkoitus on vähentää turhaa vuorovaikutusta käyttäjän kanssa, mitä enemmän käyttäjä tottuu käyttöliittymään. Tämä tapahtuu aloittamalla täsmällisellä dialogilla, mutta jos täsmällinen kysymys on jo käytön aikana kysytty ja se esitetään uudelleen, se kysytään epätäsmälliseen tyyliin ensimmäisen kerran jälkeen. Tällä tekniikalla toteutetut käyttöliittymät onnistuvat karsimaan pitkään jatkuvista sessioista paljon turhaa aikaa ja toistavat kysymykset pois.

Esimerkki kapenevasta komentokehotteesta (Yankelovich, Nicole 1996, ss. 40):

Tietokone: Jättäkää viesti vastaajaan äänimerkin jälkeen. Pitäkää parin sekunnin tauko, kun

olette valmis.

Jos saman session aikana tulisi toisen kerran äänitettävä viesti, komentokehote lyhenisi näin:

Tietokone: Jätä viesti ja pidä tauko.

3.5 Vihjeet (Hints)

Vihjeitä antava komentokehote on variaatio kapenevasta kehotteesta. Tällaisella tekniikalla toteutettu puhekäyttöliittymä antaa puhuttelevan, epätasällisen kysymyksen, johon lisätään perään vielä täsmällinen vihje. Täsmällisen ja vihjeitä antavan erona on vihjeitä antavan tekniikan tapa listata ainoastaan hyödyllisimmät vaihtoehdot, toisin kuin täsmällinen, joka luettelee kaikki mahdolliset. Vihjeet voidaan myös ottaa pois käytöstä, kun niitä ei enää käyttäjä tarvitse. Käyttöliittymä voidaan suunnitella niin, että kun sama kysymys vihjeineen on esitetty tarpeeksi monta kertaa ja käyttäjä on vastannut hyväksyttävästi, vihjeet poistetaan kysymyksestä.

Esimerkki vihjeitä antavasta komentokehotteesta (Yankelovich, Nicole 1996, ss. 41):

Tietokone: Terve. Tässä Tony Lowellin toimisto. Jos olette hän, paina tähtimerkkiä. Muuten sanokaa koko nimenne.

*Käyttäjät: (Painaa *)*

Tietokone: Mikä on salasana? Sanokaa numerot yksi kerrallaan, tai käytä painikkeita painamalla lopuksi risuaitaa.

Käyttäjät: 4 5 4 5 1 2.

Tietokone: Terve! Teillä on kaksi uutta viestiä. Ensimmäinen on Rich Minerilta. Suosittelen sanomaan Mitä-siinä-sanotaan, Kuvaile-sitä, tai Seuraava- viesti.

Käyttäjät: Mitä siinä sanotaan?

Tietokone: (Toistaa viestin) Suosittelen Soita-heille, Lähetä-vastaus, tai Poista- viesti.

Käyttäjä: Soita heille.

Tietokone: Rich Minerin työpuhelimeen? Kyllä vai ei?

Käyttäjä: Kyllä

Tietokone: Soitetaan... Paina 44 kun haluat lopettaa puhelun.

3.6 Multimodaalisten systeemien tekniikat

Multimodaalisten systeemien tekniikoilla eroavat edellisistä siinä, että se viittaa käyttöliittymiin, jotka koostuvat puheen lisäksi myös muiden käyttöliittymätyyppien elementeistä, kuten graafisen näytön. Esimerkiksi tällainen toteutus voisi antaa näytölle listan dialogeja, joita käyttäjällä on oikeus sanoa liittymälle. Nämä listat voivat olla joko vihjaavia tai erittäin täsmällisempiä, riippuen laitteiston näytön koosta.

4 Tekniikoiden käytettävyys

Kuten aiemmin mainittuna, kaksi "päätekniikkaa"puhekäyttöliittymää suunniteltaessa ovat täsmällinen ja epätäsmällinen komentokehotemalli. Nämä kaksi ovat tavallaan toistensa vastakohdat toteutuksiltaan (Hansen, B., Novick, D., and Sutton, S. 1996).

Täsmällinen tekniikka panostaa virheettömättömään käyttöön. Tekniikka pyrkii ohjaamaan käyttäjää niin, ettei tämä sano mitään mitä käyttöliittymä ei ymmärrä. Tämä tekniikka ei kuitenkaan välttämättä estä käyttäjiä tekemästä virheitä. Varsinkin jos lista dialogivaihtoehdoista on pitkä, käyttäjällä saattaa olla hankaluuksia muistaa kaikkia vaihtoehtoja ja lipsahduksia saattaa tapahtua. Täsmällisellä tekniikalla kannattaa siis käyttää vain, jos dialogivaihtoluettelo on sopivan lyhyt (Yankelovich, Nicole 1996, ss. 37-38). Toinen asia missä täsmällinen tekniikka on käytettävä, on sen opittavuus ja muistettavuus. Koska täsmällinen käyttöliittymämalli kertoo aina mitä käyttäjä voi sanoa, käyttäjän pitäisi nopeasti ymmärtää miten käyttöliittymää käytetään (Yankelovich, Nicole 1996, ss. 38). Sama pätee muistettavuuteen. Täsmällisestä käyttöliittymästä ei saa muokattua dialogivaihtoehtoja, joten se toimii joka kerralla, kuten ensimmäisellä kerralla. Kuitenkin tällainen muokattavuuden puute on vastoin Jakob Nielsenin (1995) heuristiikkoja kohdassa *Joustavuus ja käytön tehokkuus* 2.6. Tehokkuus riippuu käyttöliittymän koosta. Jos vaihteita on mahdollisimman vähän, ja dialogivaihtoehtoja on vähän, puhekäyttöliittymä olisi virheetön ja tehokas. Subjektiivisen tyytyväisyyden suhteen on hankala sanoa, sillä tyytyväisyys on subjektiivinen kokemus, tosin ottaen huomioon rajatut vaihtoehdot, täsmällisellä käyttöliittymällä ei ole paljon varaa käyttää luovasti. Ottaen kaiken huomioon täsmällinen käyttöliittymämalli saattaisi toimia parhaiten lyhyen toiminnan käyttöliittymissä, joita käyttäjä ei edes pystyisi muokkaamaan mieleisekseen, kuten esimerkiksi julkisen yhtiön tai laitoksen automaattisissa puhelinvastaajissa.

Epätäsmällinen tekniikan tarkoitus on luoda mahdollisimman luonnolliselta tuntuva puhekäyttöliittymä. Opittavuuden suhteen on vaikea sanoa, onko epätäsmällinen käyttöliittymä kuinka helposti opittava se on kullekin henkilölle. Joillekin saat-

taa olla helppoa oppia käyttämään liittymää, johon voi vain puhua luonnollisesti, joillekin taas saattaa olla hankalaa (Price, David 2013). Epätäsmällisen käyttöliittymän muistettavuuskin todennäköisesti riippuu käyttäjän omista puhetaidoista. Koska epätäsmällisessä käyttöliittymässä on vähiten rajoituksia, se on nopea ja tehokkuuden kannalta ideaalinen, jos sitä osaa käyttää oikein (Hansen, B., Novick, D., and Sutton, S. 1996). Tämä kuitenkin saattaa aiheuttaa toistuvia lipsahduksia ja erehdyksiä, jos käyttäjä ei osaa käyttää oikein käyttöliittymää. Erehdykset ovatkin täsmällisen käyttöliittymän suurin haittapuoli (Yankelovich, Nicole 1996, ss. 34). Luonnollinen puheen tavoittelemisen epätäsmällisessä puhekäyttöliittymässä yrittää maksimoida käytön tyytyväisyyttä (Hansen, B., Novick, D., and Sutton, S. 1996). Riippuen puhekäyttöliittymän tarkoituksesta, käyttäjä saattaa haluta kokeilla epätäsmällisellä käyttöliittymällä mahdollisia dialogivaihtoehtoja, nähdäkseen ymmärtääkö nämä niitä. Esimerkiksi Applen SIRI:llä on kattava määrä erilaisia viihdetarkeitukseen luotuja dialogivaihtoehtoja, kuten haikurunon generoiminen ja vitsien kyseleminen (Apple 2016). Epätäsmällinen käyttöliittymä toimii parhaiten sovelluksissa, joissa dialogin määrä on kattava ja puheentunnistus on kehittynyt, kuten monikäyttöiset sovellukset ja henkilökohtaiset avustajat.

Inkrementaalinen ja laajeneva komentokehote on epätäsmällisen komentokehoteen muunnelmä. Ero täsmälliseen komentokehoteeseen verrattuna inkrementaalinen komentokehote antaa tarvittaessa käyttäjälle hieman täsmällisempiä vaihtoehtoja, jos käyttäjä ei vastaa epätäsmälliseen dialogiin. Koska käyttöliittymä neuvoo käyttäjää sanomaan oikeat asiat, voi inkrementaalista keinoa luokitella käyttäjää opettavaksi käyttöliittymäksi. Myöskään jos käyttäjä unohtaa, mitä sanoa käyttöliittymälle, käyttöliittymä voi muistuttaa tästä. Tämä myös auttaa jonkin verran korjaamaan epätäsmällisen käyttöliittymän taipuvaisuutta virheille. Toisaalta riippuen kuinka inkrementaalinen käyttöliittymä on toteutettu erehdyksien sattuminen on silti huomioonotettava riski. Tehokkuuden osalta inkrementaalinen komentokehote toimii samalla tavoin, kuin epätäsmällinen komentokehote. Jos käyttäjä tietää, mitä pitäisi sanoa, käyttöliittymä toimii samalla tavalla kuin epätäsmällinen komentokehote. Inkrementaalinen käyttöliittymä toimii lähes samalla tavalla kuin epätäsmällinen käyttöliittymä, joten riippuen tarkoituksesta inkrementaaliseen käyttöliittymän voi

epätäsmällisen tavoin lisätä viihdetarkoituksellisia ominaisuuksia.

Kapeneva komentokehote on variaatio täsmällisestä komentokehoteesta. Kapeneva komentokehote antaa käyttäjälle täsmällisiä kysymyksiä, mutta eroaa täsmällisestä siinä, että jos samat kysymykset esitetään uudelleen, ne lyhennetään epätäsmällisiksi. Kapeneva komentokehote antaa käyttäjälle neuvoja komentojen suhteen, jotka auttavat käyttäjää sanomaan oikeat sanat. Kuitenkin koska kapeneva käyttöliittymä lyhentää dialogia, vähemmän aikaa joutuu käyttäjä käyttämään liittymää. Vähemmän dialogia nopeuttaa dialogia ja tekee liittymästä tehokkaamman (Hansen, B., Novick, D., and Sutton, S. 1996). Myöskin käyttäjän ohjaus oikeisiin sanoihin täsmälliseen tyyliin on tarkoitettu vähentämään virheiden määrää. Kapenevan käyttöliittymän käytön miellyttävyyden on suurin piirtein samaa tasoa kuin täsmällisen käyttöliittymän miellyttävyyden. Epätäsmällinen ja kapeneva käyttöliittymä ei käyttötarkoitukseltaan sovi viihdekäyttöön. Kapeneva käyttöliittymä sopii epätäsmällisen käyttöliittymän tavoin sopii lyhyisiin dialogien käyttöliittymöihin, mutta kapenevan käyttöliittymän kapenevuuden takia sillä tekemään pitempiä ja itseään toistavia dialogeja paremmin.

Vihjeitä antava komentokehote on vielä variaatio kapenevaan komentokehoteeseen. Kapenevan käyttöliittymän tapaan vihjeitä antava käyttöliittymä perustuu täsmälliseen käyttöliittymään. Vihjeitä antava käyttöliittymä antaa kuitenkin epätäsmällisiä dialogeja, joiden perään annetaan vielä täsmälliseen suuntaan johtava vihje siitä, mitä pitäisi sanoa. Vihjeiden idea on ohjata käyttäjä sanomaan sallittuja dialogeja, mikä helpottaa uusia käyttäjiä. Tällöin myös aloittelijan virheiltä vältetään, erityisesti lipsahduksilta. Kun käyttäjä on tottunut käyttöliittymään, vihjeet voi ottaa pois. Vaikka tämä tekeekin käyttöliittymästä tehokkaamman, satunnainen käyttäjä saattaa poistaa vihjeet ja sitten unohtaa kuinka käyttöliittymää käytetään. Tällaisen varalle olisi hyvä, jos vihjeet saataisiin takaisin päälle mahdollisimman helposti. Vihjeitä antavan käyttöliittymän täsmällinen tyyli ja rajoitettu puhe saattavat haitata käytön miellyttävyyttä, mutta vihjeitä antavan käyttöliittymän vahvempia puolia on sen muokattavuus täsmällisestä epätäsmällisenpää jäljittelevämpään. Kuitenkin pitää ottaa huomioon, että jos käyttöliittymä on liian hankala ja epämiellyttävä käyt-

töinen aloittavalle, käyttäjä ei välttämättä jatka sen käyttöä. Vihjeitä antava käyttöliittymämallia kannattaa käyttää, kun kyseessä on henkilökohtaisesti muokattava käyttöliittymä, jossa on niin harvinaisia dialogeja, sekä myös usein toistuvia dialogeja (Yankelovich, Nicole 1996, ss. 42).

Multimodaalisia systeemejä ei tässä nyt käydä, koska eri elementtejen yhdistelmiä on paljon. Loppujen lopuksi epätasällistä ja tasällistä tekniikat painottuvat eri asioihin. Epätasälliset käyttöliittymämallit toimivat parhaiten henkilökohtaisissa sovelluksissa, joilla myös on kattava määrä toimintoja, kuten henkilökohtaisina avustajina. Tasälliset taas sopivat hyvin yksinkertaisiin sovelluksiin, sekä julkisesti käytettäviin.

5 Yhteenveto

Tässä kirjallisuuskatsauksessa kävin läpi, mitä puhekäyttöliittymät ovat, mitä käytettävyys on, ja puhekäyttöliittymien toteutuksen tekniikat ja niiden käytettävyys. Puhekäyttöliittymä on käyttöliittymä, joka käyttää vuorovaikutuksessa puhetta ja puheentunnistusta. Puhekäyttöliittymillä on paljon hyötyjä, kuten nopeampi CPM (Characters per minute), pienempi laitteisto, kuten pieni mikrofoni. Puhekäyttöliittymän käyttäminen ei myöskään vaadi käsiä tai silmiä. Haittoina tosin puhekäyttöliittymissä on puheentunnistuksen haasteet ja rajoittunut käyttö julkisella paikalla.

Käytettävyys määritellään viidellä ominaisuudella. Nämä ominaisuudet ovat opittavuus, tehokkuus, muistettavuus, virheiden määrä ja tyytyväisyys. Toisen luvun lopussa vielä käytiin läpi Jakob Nielsenin (1995) kymmenen käyttöliittymän heuristiikkaa.

Puhekäyttöliittymien suunnitteluun löytyy erilaisia tekniikoita. Nicole Yankelovich (1996) luettelee kirjassaan *How do users know what to say?* puhekäyttöliittymän toteutukseen tarkoitettuja tekniikoita. Näistä kaksi tärkeintä ovat täsmällinen ja epätäsmällinen komentokehote. Täsmällinen komentokehote (eng. explicit prompt) listaa kaikki sallitut syötteet, jotka käyttäjä voi sanoa käyttöliittymälle. Epätäsmällinen komentokehote (eng. implicit prompt) antaa käyttäjän sanoa vapaasti, mitä haluaa, olettaen että käyttöliittymä ymmärtää tämän. Loput tekniikat ovat inkrementaalinen komentokehote, kapeneva komentokehote, vihjeitä antava komentokehote sekä multimodaaliset tekniikat, joita ovat muiden käyttöliittymän osia hyödyntävät puhekäyttöliittymät. Edellä mainitut tekniikat ovat täsmällisen ja epätäsmällisen väli-
muotoja.

Luvussa 4 vielä pohdiskelin, kuinka edellisen luvun tekniikat toteuttavat käytettävyyden ominaisuudet. Tekniikoiden toiminnallisuudet eroavat toisistaan, joten eri tekniikat sopivat paremmin eri systeemeille mitä toiset.

Kirjallisuutta

Yankelovich, Nicole, 1996. *How Do Users Know What To Say*. Saturn.

Nielsen, Jakob, 1993. *Usability Engineering*. Academic Press Inc.

Munteanu, Cosmin, Penn, Gerald, 2011. *Speech-based interaction*. Conference Proceedings - I/ITSEC Conference 2011, 2011

Hansen, Brian, Novick, David, and Sutton, Stephen, 1996. *Systematic Design of Spoken Prompts*. Conference on Human Factors in Computing Systems (CHI'96)

Nielsen, J. 1995. *10 Usability Heuristics for User Interface Design*. Saatavilla WWW-muodossa <URL: <https://www.nngroup.com/articles/ten-usability-heuristics/>> viitattu 16.4.2016

Nielsen, J. 1999. *Do Interface Standards Stifle Design Creativity?*. Saatavilla WWW-muodossa <URL: <https://www.nngroup.com/articles/do-interface-standards-stifle-design-creativity/>> viitattu 19.4.2016

Laubheimer, P. 2015. *Preventing User Errors: Avoiding Unconscious Slips*. Saatavilla WWW-muodossa <URL: <https://www.nngroup.com/articles/slips/>> viitattu 16.4.2016

Laubheimer, P. 2015. *Preventing User Errors: Avoiding Conscious Mistakes*. Saatavilla WWW-muodossa <URL: <https://www.nngroup.com/articles/user-mistakes/>> viitattu 22.4.2016

Price, D. 2013. *Siri troubleshooting guide: The 10 worst Siri annoyances, and how to fix them*. Saatavilla WWW-muodossa <URL: <http://www.macworld.co.uk/news/iphone/siri-troubleshooting-guide-10-annoyances-and-how-to-fix-them>> viitattu 19.4.2016

Apple, 2016. *iOS - Siri - Apple*. Saatavilla WWW-muodossa <URL: <http://www.apple.com/ios/siri/>> viitattu 22.4.2016