

## **Puhekäyttöliittymän opasteiden suunnittelu**

Susanna Helin

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Pro gradu -tutkielma  
Marraskuu 2002

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Susanna Helin: Puhekäyttöliittymän opasteiden suunnittelu  
Pro gradu -tutkielma, 62 sivua

Marraskuu 2002

---

Puheteknologiaa on kehitetty vuosikymmeniä ja puhekäyttöliittymien tutkimus edistyy jatkuvasti. Puhekäyttöliittymien opasteiden suunnitteluohjeita ei ole kuitenkaan tehty paljon. Tutkin tässä työssä puhekäyttöliittymän opasteiden suunnittelua. Esittelen puhesynteesin ja puheentunnistuksen periaatteita puhekäyttöliittymän opasteiden suunnittelun näkökulmasta ja tutkin olemassa olevia suunnitteluohjeita soveltaen niitä puhekäyttöliittymän opasteiden suunnitteluun. Lopuksi suunnittelen opasteet erääseen puhekäyttöliittymään tässä työssä esittelemieni suunnitteluohjeiden perusteella.

Avainsanat ja -sanonnat: puhekäyttöliittymä, puhesynteesi, puheentunnistus, on-line-opasteet, käyttöliittymän suunnittelu.

## Sisällys

1.	Johdanto .....	1
2.	Puhekäyttöliittymät ja opasteet .....	3
2.1.	On-line-opasteen ja -opastejärjestelmän määrittelmä .....	6
2.2.	Puheen ominaisuudet käyttöliittymän ja opasteiden kannalta .....	7
2.3.	Kuunteleva ja puhuva tietokone .....	10
2.3.1.	Luonnollinen vuoropuhelu ja palaute .....	11
2.3.2.	Puhesynteesi .....	15
2.3.3.	Puhesynteesi käyttöliittymän tulosteena ja opasteena .....	18
2.3.4.	Puheentunnistus .....	20
2.3.5.	Puheentunnistus käyttöliittymässä ja opasteiden käytössä .....	23
2.3.6.	Virheisiin varautuminen ja varmentaminen .....	25
2.4.	Puhekäyttöliittymien tulevaisuus .....	28
3.	Puhekäyttöliittymän opasteiden suunnittelu .....	29
3.1.	Postimies .....	29
3.2.	Miksi ja milloin käyttäjä tarvitse opasteita? .....	32
3.3.	Puhekäyttöliittymän ja opasteiden suunnitteluohjeet ja soveltaminen .....	34
3.3.1.	Käyttäjän tarpeet .....	35
3.3.2.	Opastejärjestelmän käynnistäminen .....	37
3.3.3.	Opasteen valitseminen .....	39
3.3.4.	Opasteiden esitystavan suunnitseminen .....	41
3.3.5.	Opasteiden suunnitseminen ja toteuttaminen .....	43
3.3.6.	Navigointimallin toteuttaminen .....	51
3.3.7.	Opastejärjestelmästä poistuminen .....	52
3.3.8.	Opasteiden soveltaminen käytännössä .....	53
4.	Yhteenvedo ja ajatuksia .....	54
	Viiteluettelo .....	56

## 1. Johdanto

Puheella ohjattavia tietokoneohjelmia on rakennettu vuosikymmeniä. Vaikka puhekäyttöliittymät eivät ole levinneet tavallisimpien hiirellä ja näppäimistöllä ohjattavien tietokonesovellusten lailla suurten käyttäjäjoukkojen suosioon, ala edistyy jatkuvasti. Tekniikan kehittyessä puheentunnistus ja puhesynteesi paranevat laadultaan ja tehokkuudeltaan koko ajan ja myös puhekäyttöliittymät kehittyvät edelleen. Puhekäyttöliittymien kohderyhmät ovat usein pieniä, koska puheohjaus soveltuu erityisen hyvin erikoistilanteisiin. Tällaisissa tilanteissa saattaa esimerkiksi käsien ja silmien käyttö sovelluksen ohjaamiseen olla rajoitettua toisen tehtävän tai liikuntarajoitteen takia. Pienten kohderyhmien vuoksi puhekäyttöliittymät eivät ole toistaiseksi olleet tietotekniikan suosituimpien alueiden joukossa.

Puhelin on lähes maailmanlaajuisesti käytössä oleva laite, mikä teoriassa mahdollistaa yhtä laajat markkinat kaikille puhelimitse käytettäville tietokoneohjelmille. Vaikka puheenkäsittelyä on tutkittu ja kehitetty jo vuosikymmeniä, vasta viime vuosien teknologian kehitys - tietokoneiden prosessointikyvyn tehostuminen, tietokoneen muistin halpeneminen ja puheen prosessoinnissa käytettävän algoritmiikan kehitys - on tuonut puheenkäsittelyä hyödyntäviä sovelluksia markkinoille yhä enemmän. Sovelluksia on suunniteltu sekä tavallisille kuluttajille että erikoisryhmille. Teknologian edistymisen myötä jatkuva puheentunnistus suuria sanakirjastoja käyttäen jopa ominaisuuksiltaan vaativassa lankapuhelinverkossa on mahdollista [Mané et al., 1996]. Viimeisten parinkymmenen vuoden ajan puheteknologian on ennustettu olevan seuraava suuri teknologiamullistus, ja viimeisimpien visioiden mukaan puheteknologian myötä internet saadaan kaikkien ulottuville [Lai, 2000].

Myös käyttöliittymien opastejärjestelmiä on kehitetty vuosikymmeniä. Opasteiden luonne on muuttunut viimeisten parin vuosikymmenen aikana pelkästä sähköisestä dokumentaatiosta monitahoiseksi tukijärjestelmäksi. Kehitys on tuonut mukanaan käyttäjäläheisen ja käyttötilanteiden kontekstiin sidotun lähestymistavan. Opasteet ovat nykyään osa jokaista tietokonesovellusta ja vaikka opasteille on monenlaisia suunnittelu- ja testausohjeita, juuri puhekäyttöliittymien opasteista ei ole kovin paljon tutkimuksia tai kirjallisuutta. Luultavasti syynä on ollut sekä puhekäyttöliittymien vähäisempi suosio graafisiin käyttöliittymiin verrattuna että puhekäyttöliittymiin toistaiseksi toteutettujen opasteiden yksinkertaisuus.

Tässä pro gradu-työssä keskityn selvittämään, miten puhekäyttöliittymän opasteet tulisi suunnitella. Tutkin puhekäyttöliittymien opasteiden suunnittelua perehtyen

puhekäyttöliittymien ja opasteiden ominaisuuksiin ja suunnitteluohjeisiin ja käyttäen esimerkkinä yhtä mahdollista puheohjatun sovelluksen opasteiden toteutusta. Ennen suunnitteluun paneutumista esittelen yleisesti myös puheteknologian periaatteita. Vaikka puhekäyttöliittymien suunnitteluun ei ole vielä yhtä paljon ohjeita kuin graafisten käyttöliittymien suunnitteluun, pitäydyn tässä työssä pääasiallisesti puhekäyttöliittymien suunnittelua käsittelevissä lähteissä ja otan mukaan graafisen käyttöliittymän suunnittelun osalta lähinnä opasteita koskevia ohjeita. Toisessa vaiheessa esittelen näiden ohjeiden ja puheteknologian tämänhetkisen tilanteen perusteella puhekäyttöliittymän opasteille suunnittelemani suunnitteluohjeet. Samalla sovellan ohjeita Tampereen yliopistossa kehitetyn puhelimitse ohjatun Postimies-sähköpostinlukuohjelman opasteiden suunnittelussa.

## 2. Puhekäyttöliittymät ja opasteet

Puhekäyttöliittymiä ja opastejärjestelmiä ei ole monesti käsitelty alan kirjallisuudessa ja julkaisuissa samassa yhteydessä. Varsinaisesti puhekäyttöliittymien opasteita koskevia artikkeleita ja tutkimuksia on melko vähän. Tässä luvussa esitellään on-line-opasteen määritelmä sekä käsitellään puheen ja puhekäyttöliittymien ominaisuuksia sellaisinaan ja opasteiden suunnittelun näkökulmasta.

Tietokoneohjelman käyttöliittymällä käyttäjä pääsee vaikuttamaan ohjelman toimintaan ja saa tietoa ohjelman tilasta. Käyttöliittymä on usein graafinen mutta voi olla myös pelkästään ääneen perustuva, kuten on tässä työssä myöhemmin käsiteltävän Postimiesjärjestelmän käyttöliittymä. Käyttöliittymien suunnittelu ja käyttäjän todellisten tarpeiden täyttäminen on saanut tietotekniikan kehityksessä yhä enemmän huomiota, koska useihin teknisiin ongelmiin on löydetty ratkaisu esimerkiksi tietokoneen prosessoritehon ja muistitilan parantumisen myötä. Puheteknologian kehittyminen on ollut samansuuntaista. Esimerkiksi Kamm [1994] toteaa artikkelissaan, että puheteknologian kehittyessä päästään vähitellen siirtymään puhekäyttöliittymän syötteiden ja tulosteiden teknisen toteutuksen suunnittelemisesta siihen, että keskitytään ihmisen ja tietokoneen välisen mahdollisimman luonnollisen vuoropuhelun toteuttamiseen.

Puheentunnistusteknologia onkin kehittynyt dramaattisesti viime vuosina ja alan kaupalliset tuotteet ovat yleistymässä. Huolimatta tästä kehityksestä on todettu, ettei Kammin huomion mukainen siirtymä ole itsestään selvä, sillä pelkästään teknologian kehitys ei takaa hyvää menestystä. Aiemmasta kehityksestäkään ei välttämättä ole hyötyä: graafisten käyttöliittymien suunnittelussa ei ole sellaista mallia, jota voisi suoraan soveltaa puhekäyttöliittymien suunnittelussa. [Mané et al., 1998]

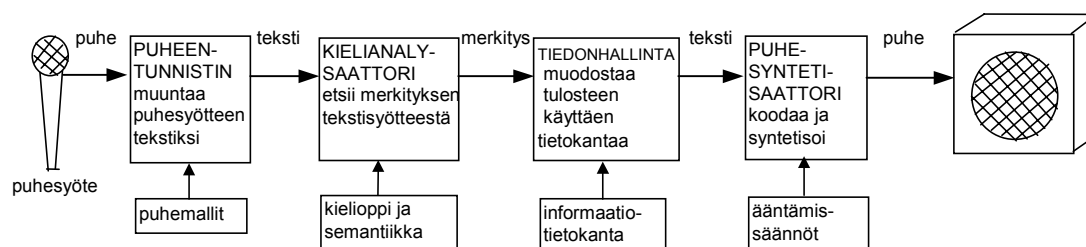
Opasteita on tutkittu aktiivisesti 1980-luvun alusta lähtien. Varhaisimmissa tutkimuksissa ilmeni, että käyttäjät eivät pitäneet opasteista, koska niiden avulla tehtävän suorittaminen sujui hitaammin kuin paperille painettujen käsikirjojen avulla. Silloiset ongelmat liittyivät vahvasti teknisiin puutteisiin, sillä eniten käyttäjät kaipaivat suurempia tietokoneen näyttöjä ja moni-ikkunointimahdollisuutta. 1980- ja 1990-lukujen vaihteessa nämä osa-alueet olivat kehittyneet siinä määrin, että käyttäjät alkoivat näyttää vastakkaista tulosta: käyttäjät pitivät opasteita hyvinä ja tehokkaina, erityisen tärkeänä pidettiin vuorovaikutteisuutta ja luontevaa siirtymistä sovelluksesta opasteisiin ja takaisin [Duffy et al., 1992, s.10-11]. Kehityssuunta on

sama kuin puhekäyttöliittymillä: tärkeimmäksi tavoitteeksi on tullut käyttäjän tarpeisiin vastaaminen.

Jotta puhekäyttöliittymien opasteille saadaan soveltuvat suunnitteluohjeet, on löydettävä sekä puhekäyttöliittymien että opasteiden suunnittelusta yhtenevät periaatteet. Puhekäyttöliittymää suunniteltaessa on ensimmäiseksi selvitettävä, onko todella perusteltua käyttää puhetta kommunikointikeinona kyseisessä sovelluksessa [Schmandt, 1994, s.154-178]. Toinen tärkeä periaate on ottaa puheen käyttö huomioon heti suunnittelun alusta alkaen, sillä puheen lisääminen valmiiseen käyttöliittymään on hankalaa [Schmandt, 1994, s. 154-178]. Nämä suunnittelussa tehtävät päätökset vaikuttavat myös opastejärjestelmään. On-line-opasteiden suunnittelussa pitää tietää kenelle opastettava ohjelma on suunnattu ja otettava käyttäjän taidot ja tiedot huomioon opasteissa [Duffy et al., 1992, s.16]. Puhekäyttöliittymän suunnittelussa käyttäjän taitojen huomioiminen on vähintään yhtä tärkeää. Puhekäyttöliittymille on löydettävissä ainakin kaksi selkeää käyttötilannetta. Ensimmäisessä tietokoneeseen ei ole muuta syöttötapaa kuin ääni: tällainen tilanne on esimerkiksi tietokoneohjelman käyttö puhelimen kautta tai kooltaan hyvin pienen laitteen ohjaaminen. Toisessa käyttäjä on keskittynyt ensisijaisesti johonkin muuhun tehtävään ja esimerkiksi käyttää silmiään tai käsiään siihen, jolloin puhe on tehokas keino työtehtävää ohjaavan ohjelman käyttämiseen tai muistiinpanojen tallentamiseen. Jälkimmäisessä ryhmässä käyttäjiä ovat erityisesti fyysiseltä toiminnaltaan rajoitetut tai tarkkaa käsityötä tekevät henkilöt [Schmandt, 1994, s. 154-178]. Jälkimmäiseen ryhmään voitaisiin katsoa kuuluvan myös sellaisten käyttäjien, joiden huomio pitää saada nopeasti kiinnitettyä tiettyyn tilanteeseen (kuten käyttämällä hälytysääntä vaaratilanteissa) [Java Speech, 1998]. Tilanteisiin, joissa käyttäjän on puhuttava toisen ihmisen kanssa yhtäaikaan käyttöliittymää käytettäessä, ei välttämättä ole järkevää toteuttaa puheentunnistus-sovellusta [Mané et al., 1996]. Puhetta ei tulisi käyttää myöskään silloin, jos käyttöympäristö on meluinen, tehtävän voi suorittaa ihan yhtä hyvin hiirellä tai näppäimistöllä tai esitettävän tiedon määrä on suuri [Java Speech, 1998]. Jos opasteisiin suunnitellaan useita käyttäjätasoja, pitää pahimmassa tapauksessa myös jokaiselle tasolle tarjota erilainen lähestymistapa opasteisiin. Siten työn määrä moninkertaistuu ja opasteiden selkeys kärsii [Duffy et al., 1992, s. 17]. Jotta käyttäjäluokitukset voidaan ottaa käyttöön, pitää joko käyttäjän tai sovelluksen selvittää käyttäjälle sopiva luokka, mikä voi olla hankalaa sekä käyttäjälle että sovellukselle. Käyttäjäluokituksista päästään eroon, jos käyttäjän kehitykseen adaptoidutaan portaattomasti.

Kun puhe on osa käyttöliittymää, joko sen tulosteina, syöteinä tai molempina, tarvitaan puheen käsittelyyn oma osansa ohjelman hallinnassa ja arkkitehtuurissa. Puheen-käsittelyä käsittelevissä lähteissä lasketaan puheenkäsittelyyn kuuluvan neljästä viiteen

erilaista teknistä osa-aluetta (riippuen lähteestä). Puheen koodaukseen kuuluu puhesignaalin pakkaaminen niin, että sitä voidaan tehokkaasti siirtää laitteiden välillä kapeillakin kaistanleveyksillä. Puhesynteesissä luodaan synteettisesti puhesignaali, jonka avulla tietokone voi kommunikoida käyttäjän kanssa. Puheentunnistuksessa taas muutetaan tietokoneen vastaanottama puhesignaali sellaiseen muotoon, että puhetta voidaan käsitellä tietokoneessa ohjelmallisesti. Puhujantunnistusta tarvitaan kun halutaan puhesignaalia analysoimalla erottaa puhujat toisistaan. Tyypillisesti puhujantunnistuksella pyritään estämään asiattomien pääsy salaisiin tai henkilökohtaisiin tietoihin. Puheen kääntäminen voidaan esittää viidentenä puheenkäsittelyyn liittyvänä teknologiana. Siinä yhdellä kielellä vastaanotettu puhesignaali käännetään ohjelmallisesti toiselle kielelle ja tuotetaan puhesynteesillä kuulijalle. Edellä esiteltyä viiteen osaan jakoa käyttää mm. Rabiner [1994], mutta mielestäni mainittu viides ala on kuitenkin puheenkäsittelyn kannalta puhesynteesin ja puheentunnistuksen yhdistämistä, enkä näe syytä sen käsittelyyn erillään muista. Kuvassa 1 on esitetty malli eräästä puheenkäsittelyjärjestelmästä joka sisältää neljä osaa.



Kuva 1. Yksinkertaistettu malli puheenkäsittelystä tietokoneohjelmassa. [Roe, 1994]

Käyttöliittymän opasteet voivat olla kiinteä osa sovellusta tai muodostaa oman kokonaisuutensa arkkitehtuurissa. Opasteet auttavat käyttäjää sovelluksen käytössä tarjoamalla sovelluksesta tietoa, joka voidaan jakaa osiin sen sisällön luonteen perusteella. Opasteista voidaan erottaa osa-alueita kuten on-line-opasteet, on-line-sanakirjat, on-line-dokumentaatio, opastusohjelmat (*tutorial*), apurit (*wizard*), virheilmoitukset tai virhetilanteessa sovelluksen tai käyttöjärjestelmän tuottamat toimintaohjeet [Duffy et al., 1992, s. 18-20]. Tässä tutkimuksessa keskitytään lähinnä on-line-opasteisiin, jotka liittyvät usein kiinteästi varsinaiseen sovellukseen. Kaikista mainituista alueista eniten on panostettu online-opasteiden kehittämiseen, koska ne on ohjelmistoteollisuudessa hyväksytyt osaksi käytettävyydeltään hyviä ohjelmistotuotteita [Turk & Corbin Nichols, 1996].

Puheen käyttö käyttöliittymässä vaikuttaa erityisesti käyttöliittymän suunnitteluun. Käyttöliittymäsuunnittelu on tehtävä huolellisesti myös silloin kun mukaan otetaan opasteet. Seuraavaksi esitellään yleisesti puheen ja opasteiden luonnetta ja merkitystä tietokoneohjelmassa ja sen käyttöliittymässä.



## 2.1 On-line-opasteen ja -opastejärjestelmän määritelmä

On-line-opaste on käyttäjälle reaaliaikaisena tarjottua relevanttia informaatiota. Tämän informaation tulee olla helposti ja tehokkaasti saatavilla ja sen pitää olla helposti siirrettävissä meneillä olevaan tehtävään sen sovelluksen sisällä, jota opaste koskee [Duffy et al., 1992, s. 18]. On-line-opaste auttaa käyttäjää tietokoneohjelman käytössä ja vastaa käyttäjän kysymyksiin [Goodall, 1991]. On-line-opasteet ovat tiiviitä, tarkkoja, interaktiivisia ja niin kiinteästi kuvaamansa sovelluksen osana kuin mahdollista [Turk & Corbin Nichols, 1996]. Opasteen tärkein tehtävä on tarjota käyttäjälle keinot ja tiedot päästä eteenpäin ongelmatilanteesta, johon hän on sovellusta käyttäessään joutunut.

Ero on-line-opasteiden ja on-line-dokumentaation välillä on sovelluskohtaista eikä rajaa voida aukottomasti määritellä. Intuitiivisesti voitaisiin ajatella on-line-dokumentaation kuvaavan sovellusta ja sen toimintaa laajemmin ja kokonaisvaltaisemmin kuin on-line-opasteiden, jotka taas sisältävät pikemminkin tilannekohtaista tietoa. On-line-opasteet kuuluvat usein läheisesti tietokoneohjelman käyttöliittymään ja niiden sisältö on tiiviimpää ja yksityiskohtaisempaa kuin on-line-dokumenttien sisältö [Turk & Corbin Nichols, 1996]. Opasteiden sisällön voi jakaa kolmeen osaan, proseduraaliseen, kuvaukselliseen ja tulkinnalliseen. Tässä jaossa proseduraalinen tieto määrittelee miten sovellusta käytetään ja mitä sillä voi tehdä; kuvauksellinen tieto esittelee sovelluksen sisällön, kuten käyttöliittymän komponentit; tulkinnallinen tieto kertoo sovelluksen eri tiloista ja syistä miksi mitään tapahtuu [Duffy et al., 1992, s.76]. Online-opasteiden tehtävä on käsitteiden ja yksityiskohtien kuvaaminen, kun opastusohjelmat ja apurit käsittelevät käyttäjien tehtäviä laajempina kokonaisuuksina [Turk & Corbin Nichols, 1996].

Opastejärjestelmä tarjoaa tehtäväkeskeistä informaatiota tehtävän suorittajalle siten, että tämä pystyy mahdollisimman helposti ja nopeasti palaamaan takaisin suorittamaansa tehtävään [Turk & Corbin Nichols, 1996]. Opastejärjestelmä koostuu tietosisällöstä, eli itse opasteteksteistä, sekä toiminnallisesta sisällöstä, jonka tarkoitus on auttaa käyttäjää löytämään opasteista tarvitsemansa tieto. Opastetekstien lisäksi opastejärjestelmässä on hyvä olla sisällysluettelo [Turk & Corbin Nichols, 1996].

Mahdollisuus suorittaa hakuja opasteteksteihin on suosituin opasteiden käyttömuoto [Goodall, 1991]. Jos hakusanalla löytyy useita opasteaiheita, käyttäjä voi selata niitä ja valita mieleisensä. Tyypillisesti käyttäjä pääsee selaamaan kaikkia opasteaiheita, jotka on järjestetty aiheiden otsikoiden mukaan. Itse opastetekstit toimivat usein

tilannekohtaisesti, eli tiettyä aihetta käsittelevää opastetta tarjotaan käyttäjälle ensimmäiseksi käynnistettäessä opastejärjestelmä kyseisen aiheen kohdalla.

Jakob Nielsenin suunnittelema käyttöliittymän heuristinen evaluointi tarkastaa käyttöliittymästä tärkeimmät käytettävyyteen liittyvät kohdat. Yksi näistä neljästätoista kohdasta koskee opasteita ja dokumentaatiota: “Opasteet ja dokumentaatio: Vaikka on parempi, että järjestelmää voidaan käyttää ilman dokumentaatiota, saattaa olla välttämätöntä tarjota käyttäjälle opasteet ja dokumentaatio. Kaiken opasteiden ja dokumentaation sisältämän informaation tulee olla käyttäjakeskeistä ja suppeaa ja sen pitää tukea helppokäyttöistä hakutoimintoa ja luetteloita käyttäjän konkreettiset tehtävät [Nielsen, 1994].”

Tässä työssä käytetään jatkossa yksinkertaisuuden vuoksi yleistä termiä *opaste* tarkoittaessa kuitenkin on-line-opastetta. On-line-opastejärjestelmään viitataan vastaavasti termillä *opastejärjestelmä*.

## 2.2 Puheen ominaisuudet käyttöliittymän ja opasteiden kannalta

Puhuminen on ihmiselle luonnollinen tapa kommunikoida esimerkiksi eleiden ja ruumiinkielen ohella. Vaikka puhuminen on yksi universaaleimmista ihmisen käyttäytymisen muodoista, puhuttu kieli ja sen eri säännöt ovat paikallisesti määräytyviä. Tämä tekee puheesta mielenkiintoisen osan tietokoneohjelman käyttöliittymää, sillä puhe ja sen kuunteleminen on suunniteltava käyttäjien yksilölliset ja yhteisölliset erot mielessä pitäen. Schafer [Schafer, 1994] kirjoittaa artikkelissaan osuvasti: “Aikoinaan, kun ihminen käytti eläimiä apunaan työnteossa, oli puheella ja äänillä suuri merkitys eläinten ohjailussa. Siitä lähtien, kun ihminen alkoi käyttää apunaan koneita, ilmassa on myös ollut halu pystyä ohjaamaan laitteita äänellä.”

Vaikka puhe rajapintana käyttöliittymässä on tehokas ja hyvin toteutettuna helppo käyttää, graafisiin käyttöliittymiin tottuneet käyttäjät pitävät koneelle puhumista usein vaivalloisena, jopa kiusallisena. Toisaalta käyttäjät olettavat puheikäyttöliittymien ymmärtävän puhetta paremmin kuin ne todellisuudessa osaavat. Tämän vuoksi esimerkiksi puheentunnistuksessa tapahtuvat virheet koetaan epämukavina ja työtä selvästi hankaloittavina [Java Speech, 1998]. Tästä puheen käytön ominaisuudesta voidaan johtaa opastejärjestelmän käynnistämiseksi asetettu käytettävyyksivaatimus, jonka mukaan opasteisiin on päästävä käsiksi helposti ja tehokkaasti [Duffy et al., 1992, s.83-91]. Avuntarpeessa kynnys kysyä tietokoneelta ei saa olla korkea ja mukavuustekijät on siten otettava huomioon varsinkin opasteiden käynnistystapaa suunniteltaessa.

Ihminen pystyy puhumaan huomattavasti nopeammin kuin kirjoittamaan, mikä on puheohjauksen etu moniin muihin syöttötapoihin nähden. Puhenopeutemme on yleensä useita satoja sanoja minuutissa, kun erinomainen konekirjoittaja yltää noin 80-100 sanaan minuutissa (esimerkit ovat englannin kielestä) [Schmandt, 1994, s.156-157]. Tämä ominaisuus on yhteneväinen opastejärjestelmän vaatimusten kanssa: opasteisiin on päästävä käsiksi vaivattomasti ja nopeasti [Duffy et al., 1992, s.83-91]. Toisaalta lukeminen on nopeampaa kuin puheen kuunteleminen. Tämän vuoksi tietokoneen on puhuttava lyhyesti ja ytimekkäästi, sillä pitkien puheenvuorojen sisäistäminen ja muistaminen on kuuntelijalle vaikeaa. Opasteiden osalta lyhyys ei ole itsestään selvä vaatimus: koska opasteen tarkoitus on tarjota käyttökelpoista tietoa käyttäjälle, ei opaste voi aina olla lyhyt. Opasteen tulisi siis olla lyhin mahdollinen siten että kaikki olennainen saadaan esitettyä.

Paitsi että puheen kuunteleminen on hidasta, puhe on lisäksi tilapäistä. Puhetta ei voi selata edestakaisin kuten tekstiä. Esimerkiksi puhemuodossa esitetty järjestelmän virheilmoitus on hetkellinen, toisin kuin graafisessa muodossa oleva virheilmoitus, joka pysyy näkyvässä näytöllä, kunnes käyttäjä poistaa sen eksplisiittisesti [Schmandt, 1994, s. 100-101]. Käyttäjän on siis keskityttävä intensiivisesti puhetta kuunnellessaan, mikä kuormittaa käyttäjän muistia ja saattaa näin hidastaa vuorovaikutusta ja vähentää tehokkuutta. Opasteen ollessa kyseessä muistinkuormitus on pidettävä rajoissa, sillä käyttäjän ollessa ongelmatilanteessa on opasteen sisältämä asia todennäköisesti käyttäjälle uutta asiaa. Uuden asian sisäistäminen kuormittaa myös muistia [Sweller 1988].

Puhe on myös luonteeltaan peräkkäistä, eli yksi asia esitetään kerrallaan. Toisin on graafisessa käyttöliittymässä, jossa käyttäjä voi muun muassa käydä läpi listan vaihtoehtoja pelkästään silmäilemällä. Luetteloivan puheen ohella esimerkiksi muotoa ja sijaintia tai muita verraten suuria tietomääriä kuvaava puhe on kömpelömpää kuin kohteen tarkastelu katseella tietokoneen näytöltä [Schmandt, 1994, s. 101-102]. Puheikäyttöliittymien opasteissakin tarvitaan luettelointia, kuten esiteltäessä käytettävissä olevia komentoja. Puheen peräkkäisyys on otettava tässä tilanteessa huomioon siten, että luettelo ei saa olla liian pitkä. Puheikäyttöliittymän suunnittelussa on pidettävä mielessä ihmisen muistin ja kognitiivisen prosessoinnin asettamat rajoitukset, joista yksi olennaisimpia on ihmisen työmuistin kuormituskyky [Kamm & Walker, 1997]. Vanhan säännön mukaan ihmisen työmuistissa pysyy tyypillisesti noin seitsemän (plus tai miinus kaksi) asiaa kerrallaan [Miller, 1956]. Tätä väitettä on viime aikoina kehoitettu välttämään ehdottomana tietomäärän rajana, sillä ihminen pystyy poimimaan isostakin tietomäärästä vain ne muutamat asiat, jotka hänen on muistettava

[LeCompte, 2000]. Puhekäyttöliittymissä on käytetty useita luettelointitekniikoita koko luettelon kerralla lukemisen lisäksi. Yleisesti luettelo on jaettu muutaman kohdan ryhmiin siten, että ryhmän lopussa kerrotaan käyttäjälle äänimerkillä, sopivalla prosodialla tai esimerkiksi sanoin "and more", että luettelo jatkuu vielä [Shriver et al., 2001].

Ihmisen pystyy suorittamaan kahta tai useampaakin tehtävää samanaikaisesti, minkä vuoksi puhe sopii luontevasti monen toiminnon oheen, kuten käsillään työtä tekevän ja toisaalle katsovan henkilön avuksi [Schmandt, 1994, s. 156-157]. Opasteiden kannalta tämäntyyppiset työtehtävät vaativat tilannekohtaisuutta: vaikka käyttäjällä olisikin erinomainen kyky suoriutua samanaikaisista tehtävistä, häntä kuormittavat jo suoritettava tehtävä ja siihen liittyvä sovellus. Käyttäjä ei voi kesken työtehtävän käynnistää opastejärjestelmää päätasolta ja yrittää itse etsiä tilannettaan kuvaavaa opastetta. Tilannekohtaisen opasteen on oltava käyttäjän saatavilla samassa työtilassa ja -tilanteessa [Goodall, 1991].

Puheen avulla voidaan ohittaa syviä valikkohierarkioita, jotka saattavat olla ongelmallisia graafisissa käyttöliittymissä [Java Speech, 1998]. Puheella voi nopeasti ja tehokkaasti sitoa yhteen komentoja ja niiden parametreja luonnollisen kuuloisiksi lauseiksi. Puhesovelluksen käyttö saattaa siten olla miellyttävämpää kuin dialogin avaaminen hiirellä monitasoisen valikkorakenteen kautta, erilaisten arvojen syöttäminen ja muuttaminen kentästä toiseen cursorilla siirtymällä ja dialogin sulkeminen. Puhe ei yleensä sovi ainoaksi syötemuodoksi tietokoneohjelmaan, joka perustuu runsaaseen grafiikkaan ja tiedon visualisointiin, sillä silloin komentojen antaminen sanoin saattaisi olla hankalaa.

Spatiaalinen tai geometrinen tieto on vaikeaa kuvata ja sitä on vaikea käsitellä pelkän puheen avulla. Tällaisen tiedon osalta suoramanipulaatiokäyttöliittymä, eli käyttöliittymä, jossa käyttäjä pystyy välittömästi käsittelemään tietoa (useimmiten visuaalisena objektina), on puhekäyttöliittymää tehokkaampi ratkaisu. Suoramanipulointi on puheen käyttöä tehokkaampaa silloin, kun manipuloitavat kohteet ovat näkyvissä ruudulla eikä niiden määrä ole liian suuri visuaalisesti käsiteltäväksi. Suoramanipulaatio toisaalta rajoittaa käyttäjän mahdollisuuksia samalla, kun se estää käyttäjää antamasta vääriä komentoja. Luonnollisen puheen tunnistamisen käyttö käyttöliittymässä sen sijaan helpottaa isojen objektimäärien käsittelyä ja myös antaa käyttäjälle mahdollisuuden muodostaa komennot siten kuin käyttäjä itse asian ymmärtää. [Cohen & Oviatt 1994]

Suunniteltaessa puheen käyttämistä tietokoneohjelmassa pitää myös muistaa, että puhe vie talletettaessa enemmän tilaa kuin teksti. Pitkien puheviestien käsittelyssä voi tiedoston sisältöä kuvaavien avainsanojen löytäminen olla hankalaa. Toisaalta graafiset elementit vievät muistitilaa vielä enemmän, joten graafiseen käyttöliittymään verrattuna pelkästään puheeseen perustuva käyttöliittymä voi olla muistinkulutuksen kannalta taloudellisempi ratkaisu. Puheen kanssa saattaa yksityisyys olla vaarassa: useimmiten puhe kuuluu kauemmas kuin teksti näkyy. Tämäkin ongelma ratkeaa puhetulosteiden osalta, jos käyttäjä pystyy käyttämään kuulokkeita.

Todennäköisesti edellä esiteltyjen puheen ominaisuuksien takia puhekäyttöliittymiä käsittelevissä lähteissä poikkeuksetta kehoitetaan pohtimaan, onko puhe välttämätön tai jollain tavalla erityisen suositeltava piirre käyttöliittymässä. Puheen käytön suunnittelu ja toteutus käyttöliittymään on yleisen mielipiteen mukaan työläs tehtävä, joten puheen käytölle vaaditaan hyvät perustelut. Toisaalta esimerkiksi Cohen & Oviatt [1994] vetoavat artikkelissaan käyttäjälähtöiseen suunnitteluun todeten, että puheen käyttöä voidaan pitää perusteltuna pelkästään sen mukavuuden vuoksi. Puhuminen koneen kanssa on siis pyrittävä järjestämään käyttäjälle mahdollisimman mukavaksi, jolloin puhe puolustaa paikkaansa käyttöliittymässä. Olennaisin opasteissa hyödynnettävä puheen ominaisuus on aiemmissa kappaleissa esitetyn perusteella sen vaivattomuus ja nopeus. Eniten rajoitteita asettaa puheen kuuntelemisen hitaus ja puheen tilapäisyys ja peräkkäisyys. Käyttötilanne on lopulta ratkaisevassa asemassa päätettäessä puheen käytöstä käyttöliittymässä.

Lopuksi huomautettakoon, että termiä *luonnollinen* käytetään useissa lähteissä kuvastamaan yleisesti puheen luonnollisuuteen liittyviä piirteitä, puhesynteesin luonnollisuutta ja toisaalta käyttäjän mahdollisuutta käyttää luonnollista puhetta. Jälkimmäisellä viitataan puheentunnistusjärjestelmän ominaisuuksiin, jotka luovat käyttäjälle mielikuvan tavallista, arkikielen puhetta ymmärtävästä tietokoneesta, jonka kanssa käyttäjän mahdollisuuksia ole puheen osalta mitenkään rajoitettu. Käytännössä täydellinen luonnollisen puheen tunnistaminen on kuitenkin hyvin vaikeaa toteuttaa [Mané et al., 1996]. Jatkossa tässä työssä viitataan luonnollisella puheella nimenomaan puhetulosteiden luonnollisuutta tukeviin ominaisuuksiin, luonnollisen puheen tunnistamisella käyttöliittymän kykyyn käsitellä luonnollista puhetta.

### **2.3 Kuunteleva ja puhuva tietokone**

Puhekäyttöliittymässä puheen tuottamiseen tarvitaan puhesynteesiä ja puheen kuuntelemiseen puheentunnistusta. Lisäksi puheen analysointiin tarvitaan kieliopin ja semantiikan käsittelyä. Näiden osien avulla puhekäyttöliittymä voidaan suunnitella

puheen tasolla vuorovaikutteiseksi ja vaikuttaa sen käytettävyyteen. Puhekäyttöliittymiin tarvittavan teknologian kehityksessä on vielä paljon haasteita: sanavarastojen pitäisi olla laajoja ja puhetta pitäisi kyetä tunnistamaan puhujasta riippumatta, vaikka puhuja käyttäisi arkikieltä pitämättä taukoja sanojen välissä ja puhuisi meluisassa ympäristössä.

Puheen prosessoinnista käytetään useita erilaisia termejä. Puheentunnistus ja puhesynteesi sisältyvät luonnollisen kielen ymmärtämiseen (Natural Language Understanding eli NLU) ja luonnollisen kielen tuottamiseen (Natural Language Generation eli NLG). Yhdessä nämä osa-alueet muodostavat luonnollisen kielen prosessoinnin (Natural Language Processing eli NLP). Edellä mainitun Natural Language -termin lisäksi käytetään vastaavassa yhteydessä myös termiä Spoken Language (kuten Spoken Language Understanding eli SLU), jolloin tarkoitetaan nimenomaan puhetta eikä tekstiä. Luonnollisen kielen prosessointi on mielenkiintoinen tutkimuskohde erityisesti kognitiivisten prosessien tutkijoille, joten alalla on myös kielitieteilijöitä ja psykologeja. [Price, 1996]

Seuraavissa kappaleissa on esitelty yleisellä tasolla vuoropuhelun ominaisuuksia, puhekäyttöliittymän suunnitteluun liittyviä piirteitä ja rajoituksia sekä puhesynteesin ja puheentunnistuksen periaatteita. Teknisiä periaatteita ei ole tarkasteltu kovin yksityiskohtaisesti. Asian käsittelyssä on pyritty kiinnittämään huomiota niihin puhesynteesin ja puheentunnistuksen ominaisuuksiin, jotka vaikuttavat käyttöliittymän suunnitteluun erityisesti opasteiden osalta.

### **2.3.1 Luonnollinen vuoropuhelu ja palaute**

Vuorovaikutukseen kuuluu puhuminen ja kuunteleminen, joten puhekäyttöliittymien suunnittelussa on pidetty pitkään tavoitteena luoda käyttäjälle sellainen ympäristö, joka noudattaa luonnollisen vuoropuhelun sääntöjä. Opastejärjestelmässä käytetään usein joko käyttäjä- tai järjestelmälähtöistä lähestymistapaa. Vuoropuhelua voidaan tavoitella esimerkiksi interaktiivisten apurien muodossa. Seuraavaksi käsitellään luonnollisen vuoropuhelun ominaisuuksia, asemaa käyttöliittymässä ja käyttömahdollisuuksia opastejärjestelmässä.

Vuoropuhelussa osapuolet keskustelevat useimmiten samasta aiheesta käyttäen vuorotellen puheenvuoroja. Puheenvuoron vaihtuessa pidetään usein lyhyt tauko, jonka pituus riippuu puheen temposta sekä puhujan kulttuurista. Toisinaan puhe on päällekkäistä, kuten väittelytilanteessa osapuolien keskeyttäessä toisiaan. Taukoja syntyy esimerkiksi puhujan hengittäessä, usein kuitenkin niin, että yhdellä

hengenvedolla puhuttu asia muodostaa yhden selkeän kokonaisuuden. Tauot sekä käyttäjän että järjestelmän puheessa on otettava huomioon, koska ne voidaan helposti tulkita väärin. Puheenvuorojen rajoja voi olla vaikeaa erottaa edellä mainittujen seikkojen takia, mutta ihmisten välisessä keskustelussa puheenvuorojen rajat löytyvät lisäksi puheenvuoron lopettamiseen viittaavan intonaation, viimeisen tavun venyttämisen, eleiden, fraasien ja katsekontaktin avulla. [Schmandt, 1994, s. 194-196]

Puhekäyttöliittymän puheeseen saattaa syntyä taukoja, koska puheen prosessoiminen on toisinaan hidasta. Käyttäjän näkökulmasta tauot voivat tarkoittaa sitä, että puheentunnistin joko prosessoisi saamaansa syötettä tai odottaa käyttäjän puhuvan lisää, tai että tunnistin ei ole joko kuullut tai ymmärtänyt käyttäjän puhetta kunnolla [Levow, 1997]. Käyttäjälle tulisi antaa välitöntä palautetta sovelluksen toiminnasta, jotta välttyään käyttäjälle epäselviltä tilanteilta.

Puhekäyttöliittymät voivat olla joko aloitteellisia (*system-initiative*), passiivisia (*user-initiative*) tai molempia (*mixed-initiative*) [Horvitz, 1999]. Aloitteellinen puhekäyttöliittymä johtaa vuoropuhelua ja kysyy aktiivisesti käyttäjältä seuraavaa toimintoa ja ohjaa käyttäjää. Aloitteellinen opastejärjestelmä pyrkii aktiivisesti tunnistamaan käyttäjän ongelmatilanteen ja tarjoamaan ratkaisua ongelmaan. Passiivinen käyttöliittymä antaa käyttäjän päättää ja näiden muotojen yhdistelmässä aloitteellisuus on eritasoista riippuen tilanteesta. Passiivinen opastejärjestelmä on pelkästään käyttäjän käynnistettävissä ja opastejärjestelmässä navigointi hoituu täysin käyttäjän toimesta. Puhekäyttöliittymä voidaan rakentaa adaptiivisesti aloitteelliseksi, jolloin järjestelmä seuraa keskustelun tai käyttöliittymän käytön rakennetta ja muuttaa sen mukaan toimintansa aloitteellisuuden tasoa [Heeman & Strayer, 2001]. Tarve tällaiselle toimintamallille syntyy halusta jäljitellä luonnollista vuoropuhelua: ihmisten välisessä keskustelussa ei aloitteenteko useinkaan ole yhden ihmisen vastuulla, vaan aloitteellisuus vaihtelee tilanteen mukaan. Järjestelmän tekemiä aloitteita suunniteltaessa pitää ottaa huomioon aloitteen tuomat kustannukset: mitä aloitteesta hyödytään ja mitä haittaa siitä voi olla; tämän arvioimisessa on hyvä tietää käyttäjän tilanne ja tavoite [Horvitz, 1999]. Opastejärjestelmässä aloitteellisuuden tasoa voidaan muuttaa esimerkiksi käyttäjän kokemuksen (käytännössä esimerkiksi käyttökertojen määrän) perusteella: vasta-alkajalle tarjotaan opaste aikaisemmin kuin kokeneelle käyttäjälle.

Luonnollisessa vuoropuhelussa pelkästään vuorossa oleva puhuja ei ole äänessä, vaan myös kuuntelijat osallistuvat keskusteluun tilanteesta ja kulttuurista riippuen enemmän tai vähemmän aktiivisesti. Puhujan asiaa myötäilevät hymähdykset ja lyhyet lausahdukset kehottavat puhujaa jatkamaan puhumista [Schmandt, 1994, s. 198].

Kuuntelijoiden antama palaute on toisinaan olennaista, jotta puhuja haluaa tai kykenee esittämään asiansa. Puhuja pysyy kuuntelijoiden kommentoimissa myös selvillä siitä, kuuntelevatko vuoropuheluun osallistujat. Puheen ollessa käyttöliittymän ainoa palautemuoto käyttäjälle jäävät pois tavallisessa ihmisten välisessä kanssakäymisessä tärkeät eleet kuten ruumiinkieli ja ilmeet. Intonaatio saattaa riippua jopa enemmän puhujan eleistä kuin lauserakenteista [Bolinger, 1983]. Esimerkiksi äänen noustessa kädet usein liikkuvat ylöspäin, painotusten kohdalla esiintyy rivakoita liikkeitä ja äänen laskiessa kohti lauseen loppua ruumiinkieli rauhoittuu [Kendon, 1980]. Tällä perusteella puheentunnistuksessa voidaan hyödyntää sitä tietoa, että osa eleiden välittämää informaatiota siirtyy intonaatioon, mutta puhesynteesiin oikean intonaation luominen on hankalaa, jos pelkkien lauserakenteiden analysoiminen ei riitä. Intonaatiota ja prosodiaa käsitellään myös edempänä puhesynteesin ohessa.

Käyttäjälle tärkeää palautetta voidaan antaa myös muussa muodossa kuin puheena. Esimerkiksi graafisen käyttöliittymän antama visuaalinen palaute napinpainalluksesta (nappi painuu alas ja ponnahtaa ylös) voidaan puhekäyttöliittymässä korvata napinpainalluksen äänellä, jollain muulla merkkiäänellä tai sanallisella palautteella. Puhekäyttöliittymien ääniä, jotka eivät ole puhetta, on myös tutkittu melko paljon. Puhekäyttöliittymien äänimaailma on oma aihealueensa, joten sen käsittely jätetään tämän työn ulkopuolelle. Aiheesta voi lukea lisää esimerkiksi Darvishin ja kumppanien artikkelista [Darvishi et al., 1995].

Kun käyttöliittymä perustuu ainoastaan puheeseen, käyttäjä saattaa kokea ongelmaksi sen, ettei tiedä, mitä pitäisi sanoa. Graafisessa käyttöliittymässä on etuna se, että sovelluksen sisältämät toiminnot ja käyttäjälle tarjolla olevat vaihtoehdot ovat käyttäjän nähtävissä. Puhekäyttöliittymässä törmätään tältä osin kahteen ongelmaan: käyttäjä joko kuvittelee sovelluksen ymmärtävän enemmän kuin se ymmärtää, tai käyttäjä ei ole tietoinen kaikista toiminnoista, joita olisi tarjolla. Yankelovich [1996] suosittelee huolellista käyttäjää ohjaavien tulosteiden suunnittelua, mutta toteaa myös dokumentaatiosta ja opasteista olevan apua. Paras tapa saada käyttäjä sanomaan juuri se, mitä pitää, on luetella tarjolla olevat komentovaihtoehdot. Tätä suositellaan vain, jos järjestelmään kuuluu keskeytyksen mahdollisuus (eli järjestelmä pystyy tunnistamaan käyttäjän puhetta myös puhetulosten aikana). Toinen tapa ohjata käyttäjää on antaa käyttäjälle vihjeitä puhetulosten lomassa. Vihjeissä annetaan käyttäjälle vain yleisimmät toimintavaihtoehdot. Esimerkiksi sähköpostinlukuohjelmassa voidaan yhden postin lukemisen lopuksi sanoa: "Voit sanoa esimerkiksi tuhoa tai seuraava." Puhetulosteet voidaan myös suunnitella siten, että jollei käyttäjä pienen hiljaisuuden jälkeen osaa vastata mitään tai antaa komentoa, tarjotaan käyttäjälle hieman edellistä tarkemmat ohjeet sisältävä puhetuloite. [Yankelovich, 1996]



Samanlaista lähestymistapaa suosittelevat myös Kamm ja Walker [1997] ja muistuttavat samalla, että tilannekohtaisuus vähentää lueteltavien valintojen määrää. Heidän mielestään toinen keino tiettyjen komentojen muistamiseksi on pitää pieni osajoukko tärkeimpiä komentoja aina valittavissa, myös eri sovelluksissa: esimerkiksi opasteen käynnistävä komento "help me out" voisi toimia universaalina komentona [Kamm & Walker, 1997]. Myös Shriver et al. [2001] esittelevät joukon universaaleiksi sopivia komentoja, joita käsitellään tässä työssä myöhemmin opasteiden suunnittelun yhteydessä.

Yankelovich, Levow ja Marx esittelevät SpeechActs-prototyypin suunnittelua ja testausta käsittelevässä artikkelissaan [Yankelovich et al., 1995] kolme luonnolliseen vuoropuheluun liittyvää haastetta. Ensimmäisenä he mainitsevat käyttäjän ongelman: käyttäjä ei tiedä mitä pitäisi sanoa. Erityisesti tämä ongelma korostui tilanteessa, jossa järjestelmä lopetti oman puhetulosteensa asettamatta käyttäjälle yhtään kysymystä ja osoittamatta seuraavaa toimintoa ja käyttäjän piti osata jatkaa sen jälkeen itsenäisesti. Ratkaisuksi he keksivät jatkaa järjestelmän puhetulostetta kysymyksellä: "Mitä sitten?". Tämän jälkeen käyttäjät osasivat ottaa taas ohjat omiin käsiinsä eivätkä jääneet odottamaan järjestelmän aloitetta. Opastejärjestelmässä voitaisiin käyttää vastaavaa toimintaa sujuvasti esimerkiksi kysymyksenä "Mistä haluat tietää lisää?" Seuraava SpeechActs-prototyypin kanssa vastaan tullut haaste liittyi prosodian käyttöön luonnollisen vuoropuhelun tuottamisessa. Käyttäjien mielestä järjestelmän myötäelävät hymähdykset kuulostivat kuitenkin keinotekoisilta ja metallisilta. Kolmantena kohtana Yankelovich ja muut pohtivat järjestelmän vuoropuhelun tempoa. Vaikka puhetulosteet olivat melko nopeita, ne eivät riittäneet ihmisten välistä vuoropuhelua jäljittelevään kommunikointiin. Tähän kirjoittajat ehdottivat ratkaisuksi sekä keskeytysmahdollisuutta että myös suoritusajoiltaan nopeita puhelimen näppäimillä tehtäviä pikavalintoja tyypillisimmille toiminnoille. Esimerkiksi opastejärjestelmän käynnistämiseen pikavalintanäppäin sopisi hyvin.

Edellä esitetyt vuoropuheluun ja palautteeseen kuuluvat kohdat edellyttävät ensinnäkin luonnollista puheenvuorojen vaihtoa, keskeyttämisen mahdollisuutta, järjestelmän palautetta sekä myös asettavat vaatimuksia syntetisoidulle puheelle, jossa pitäisi esimerkiksi ottaa huomioon luonnollisenkuuloiset hengitystauot ja puhujaa myötäilevät kommentit. Tasapuolinen vuoropuhelu ohjelman ja käyttäjän välillä ei kuitenkaan aina ole paras ratkaisu. Vuoden 1996 CHI-konferenssin eräessä työryhmässä [Mané et al., 1996] havaittiin seuraava seikka tutkittaessa joidenkin puhekäyttöliittymien käyttäjäkokemuksia: vaikuttaisi siltä, että käyttäjät haluavat tarkat ohjeet siitä, mitä järjestelmälle voi sanoa missäkin tilanteessa. Tutkituissa sovelluksissa ei kuitenkaan

tuettu luonnollista vuoropuhelua, joten havainnon voidaan olettaa koskevan vain käyttäjälähtöisiä, komentopohjaisia käyttöliittymiä. Sähköpostiohjelman puhekäyttöliittymän evaluointia käsittelevässä artikkelissa [Walker et al., 1998] esitetään, että käyttäjät pitävät aloitteellisesta käyttöliittymästä. Toisin sanoen sellaista ohjelmaa on helpompi käyttää, joka kysyy käyttäjältä seuraavan toiminnon eikä jää odottamaan, että käyttäjä etenee oma-aloitteisesti haluamaansa suuntaan. Samassa yhteydessä väitetään, että aloitteellisen käyttöliittymän käyttö on helpompi oppia ja sen toiminta on paremmin ennustettavissa.

Käytettävyyssiantuntija Ben Shneiderman on kritisoinut luonnollisen vuoropuhelun jäljittelyä väittämällä, ettei koneen ja ihmisen välinen vuoropuhelu voi koskaan vastata kahden ihmisen välistä vuoropuhelua, koska koneelta puuttuvat tunteet ja tunteet muodostavat suuren osan puheen prosodiasta [Shneiderman, 2000]. Mielestäni luonnollinen vuoropuhelu on miellyttävän tietokoneen käytön ja jo tutkimuksen mielenkiintoisuudinkin kannalta tavoittelemisen arvoinen asia, vaikkei sitä koskaan saavutettaisikaan tietokoneen kanssa keskusteltaessa. Tämän työn kannalta mielenkiintoinen aihe on se, ettei käyttäjä tiedä mitä sanoa, sillä se liittyy erityisesti kaikkiin opasteiden suunnittelun pääkohtiin. Käyttäjän on tärkeää tietää mitä sanoa opastuksen saamiseksi ja myös miten jatkaa puhesynteesillä annetun opasteen jälkeen.

### 2.3.2 Puhesynteesi

Puhesynteesi eli ääniaaltojen automaattinen tuottaminen on vanha keksintö. Vuonna 1971 Wolfgang von Kempelen rakensi urkupillejä apunaan käyttäen ensimmäisen ns. puhesyntetisaattorin [Mattingly, 1974]. 1930-luvulla kehitettiin ensimmäinen elektronisessa muodossa puhetta käsittelevä laite (*Vocoder*), joten nykyistä elektronista puhesynteesiä on myös kehitetty jo vuosikymmeniä [Klatt, 1987]. Viime vuosina puhesyntetisaattorit ovat kehittyneet tasolle, jolla puhesynteesin ymmärrettävyys on hyvä, mutta luonnollisuus ja äänenlaatu ovat vielä ongelmallisia [Klevans & Rodman, 1997, s. 2]. Puhesynteesiteknologioita on kolme: yhdistävä, mallintava ja artikulaatiivinen synteesi. Näitä synteesin muotoja esitellään seuraavassa lyhyesti. Lopuksi käsitellään yleistä puhesynteesin käyttötapaa, synteesin muodostamista tekstistä, puheen pakkaamista sekä joitakin puhesynteesiin liittyviä metodeita.

Yksinkertaisin tapa tuottaa puhesynteesiä on nauhoittaa ennalta tarvittavat sanat tai lauseet ja soittaa niitä tarvittaessa, sellaisinaan tai yhdistellen. Yhdistävä synteesi tuottaa laadukasta ja luonnollisen kuuloista puhesynteesiä, mutta sanavarasto jää usein pieneksi. Toisaalta koska sanojen ääntämisasu ja painotukset saattavat muuttua eri tilanteissa, yhdistävä synteesillä tuotettu puhe ei vaikuta luonnolliselta

vuoropuhelutilanteissa. Siksi se sopii yleensä erilaisiin tietopalveluita tarjoaviin sovelluksiin ja ilmoitusluonteisiin puhelusteisiin. Yhdistävän synteessin ongelma on suuren ääninäytämäärän tarve, sillä sanavaraston pitää olla melko suuri jo pelkän arkikielen käytössä ja useimmista sanoista tarvitaan esimerkiksi erilaisia sijamuotoja. Yhdistävän synteessin sanavarastoa saadaan kasvatettua yhdistelemällä sanojen sijasta foneemeja kielen ääntämissääntöjen mukaan. Tällaisessa synteessissä korostuu foneemien taipumus muotoutua ympäristönsä mukaan, jolloin voidaan käyttää myös foneemipareja, jotta foneemien rajat saadaan kuulostamaan oikeilta. Voidaan myös tallettaa yleisimmät sanat kokonaisina ja soveltaa loppuihin sanoihin ääntämissäännöstöä. [Schmandt, 1994, s. 82-86]

Mallintava synteesi perustuu säännöstöön, jonka perusteella äänen lähettä ja taajuutta säädellään puheen tuottamiseksi. Yleisesti käytetyin mallintava puhesynteessimenetelmä on formantti synteesi, joka ensimmäisen kerran esiteltiin jo vuonna 1953 [Klatt, 1987]. Formanttiin synteisiin vaadittava sääntöjen määrä on laaja, mikä on osaltaan formantin synteessin ongelma. Toistaiseksi luonnollisen kuuloinen formantti synteesi on myös ollut vaikea toteuttaa. Formantin synteessin suosio on vähentynyt viime aikoina, kun yhdistävää synteesiä on taas alettu käyttää enemmän [Shadle & Damper, 2001].

Teoriassa tarkin tapa tuottaa puhetta keinotekoisesti on artikulaatiivinen synteesi, joka mallintaa suoraan ihmisen äänielimistöä [Shadle & Damper, 2001]. Käytännössä tämänkaltaisen monimutkaisen synteessin tuottaminen ei ole vielä riittävän kehittyntä, vaikka ensimmäinen artikulaatiivinen syntetisaattori kehitettiin jo vuonna 1958 [Klatt, 1987]. Aiemmin artikulaatiivisen synteessin keskeisin ongelma oli äänielimistön toimintaa riittävän tarkasti kuvaavan tiedon suuri määrä ja laskennallisen tehokkuuden heikkous suhteessa siihen [Kröger, 1992]. Edelleenkin artikulaatiiviseen synteisiin tarvittavan tiedon kerääminen on vaikea tehtävä, mutta artikulaatiivista synteesiä pidetään pitkällä tähtäimellä parempana ratkaisuna kuin paljon käytettyä yhdistävää synteesiä [Shadle & Damper, 2001]. Perusteluna Shadle ja Damper pitävät artikulaatiivisen synteessin suurempaa potentiaalia laadukkaana ja monimuotoisen puheen tuottamiseen. He myös korostavat puhesynteessin parametroinnin myönteistä kehitystä artikulaatiiviseen synteisiin keskittyvän tutkimuksen myötä.

Puhesynteisiin liittyy läheisesti puheen pakkaaminen tiedon siirtämisen ajaksi. Puheen pakkaamisen perusajatus esitettiin alunperin jo vuonna 1928: puhesignaalia siirretään vain juuri sen verran tietoa, jolla puhesignaali saadaan konstruoitua uudestaan määränpäässä. Puhesignaalin tallettamiseen vaadittavan tilan minimoimiseksi paras pakkaamistapa olisi ensin tunnistaa puhe ja tallettaa se tekstinä, jonka jälkeen puhe voitaisiin muodostaa uudelleen puhesynteisillä tekstistä. Laadukkaampaa puhetta

varten tarvitaan kuitenkin toistaiseksi enemmän informaatiota. [Schroeder, 1999, s. 3, 63]

Puheen analysointi ja synteessin tuottaminen muodostuvat sitä monimutkaisemmiksi mitä tehokkaammin puhetta pakataan, muun muassa siksi, että ihmisen puheessa on paljon hyödyttömiäkin informaatiota. Puheen pakkaamiseen valitun koodaustavan valinta vaikuttaa siihen, miten alhaisilla tiedonsiirtonopeuksilla digitaalista puhetta pystytään käsittelemään siten, että esimerkiksi puhesynteesi on vielä siedettävää ja ymmärrettävää kuunneltavaa eivätkä vasteajat kasva liian suuriksi. [Schafer, 1994]

Puheen siirtämisen ohella sen tallentaminen on myös tuonut mukanaan haasteen: digitaalisen puheenkäsittelyn aikana tavoitteena on puheen tallettaminen mahdollisimman tiiviissä muodossa, usein tekstinä. Alun perin puhesynteesin tuottamiselle tekstistä oli motivoijana sokeiden avustaminen [Schroeder, 1999 s. 16], mutta internetin käytön yleistyessä monia internetin kautta tekstimuodossa levitettäviä tietomateriaaleja voidaan ottaa monipuolisemmin käyttöön, kun käytetään tekstistä muodostettavaa puhesynteesiä [Schroeder, 1999, s. 86]. Puhesynteesi, jota tuotetaan tekstistä, syntyy kahdessa vaiheessa. Ensin teksti analysoidaan muuttamalla se foneettiseen tai muuhun kielelliseen esitysmuotoon. Toisessa vaiheessa ääniaallot tuotetaan syntetisaattorilla tämän foneettisen ja prosodiaan perustuvan tiedon avulla. Näistä vaiheista puhutaan yleensä nimillä *high-* ja *low-level synthesis* [Tatham & Morton, 1995].

Tekstin analysointiin puhesynteesiä varten kuuluu yleensä kolme vaihetta: Ensimmäisenä suoritettavassa normalisoinnissa, jota kutsutaan myös tekstin esikäsittelevävaiheeksi, tunnistetaan tekstistä erikoismerkit, kuten internetosoitteet, jotka noudattavat erityisiä lausumissääntöjä. Toisessa vaiheessa tehdään morfologinen analyysi, jossa erotellaan sanat morfeemeihin, kuten sanan vartaloon ja päätteeseen. Kullekin morfeemille etsitään ensin valmiiksi tallennettua ääntämismallia, ja jollei sitä löydy, sovelletaan kyseiseen morfeemiin ääntämissääntöä. Kolmannessa vaiheessa pyritään saamaan mukaan luonnollisuuden tuntua tuottamalla puhesynteesille oikeanlainen prosodia. Luonnolliseen puheeseen liittyy muun muassa eri äänteiden painotus, äänteiden muuntuminen riippuen äänteen sijainnista sanassa ja äännettä ympäröivistä muista äänneistä ja painotuksista ja intonaatio eli lauseen "sävelkulku". [Schmandt, 1994, s. 87-91]

Yksi puhesynteesin luonnollisuuden ja miellyttävyyden olennaisimpia tekijöitä onkin prosodia, jonka keinotekoinen tuottaminen on hankalaa, sillä sisällön vaikutusta puheeseen on vaikea päätellä. Yksinkertaisessa prosodiassa on tauot välimerkkien

kohdalla ja lauseiden loput nousevat tai laskevat lauseen tyyppin mukaan. Prosodian parametrinen mallintaminen on vaikeaa, koska ei tarkkaan tiedetä, miten prosodiaan liittyvät parametrit vaikuttavat toisiinsa, minkä vuoksi aihe vaatiikin tutkimusta tulevaisuudessa [Zue et al., 1996].

Moderniin puhesynteesiin kuuluu useita monimutkaisia metodeita ja algoritmeja. Yksi viime aikoina paljon käytetty metodi on HMM (Hidden Markov Model), jota alettiin käyttää jo 1970-luvun lopulla puheentunnistuksessa ja pian sen jälkeen puhesynteesissäkin. HMM perustuu tilakaavioon, jossa tilojen välisillä siirtymillä on omat todennäköisyytensä sekä rajallisella joukolla ääniteitä toimivat todennäköisyys-funktionsa [Rabiner, 1989]. Toinen mainittavan arvoinen metodi on ensimmäisen puhesyntetisaattorin Vocoderin jälkeensä jättämä LPC-metodi (*Linear Predictive Coding*), jota käytetään puhedatan pakkaamisessa, puheentunnistuksessa ja tekstistä muodostettavassa puhesynteesissä [Schroeder, 1999, s. 3].

Puhesynteesin laatua voidaan mitata sen ymmärrettävyydellä ja miellyttävyydellä. Ymmärrettävyyttä voidaan mitata kuullunymmärtämistesteillä. Yksi tapa selvittää puhesynteesin laatua on kuormittaa käyttäjien työmuistia puhesynteesillä: mitä kehnompaa puhe on laadultaan, sitä huonommin kuuntelija muistaa siinä kerrotut asiat, koska huonolaatuisen puheen ymmärtäminen vaatii enemmän keskittymistä. Samalla tavalla voidaan käyttäjän toiminnan vasteaikoja pitää indikaationa puheen laadusta, kuten myös käyttäjän henkilökohtaista kokemusta sovelluksesta. [Schmandt, 1994, s. 94-97]

Puhesynteesistä on saatu viime vuosina niin selkeää, että puhekäyttöliittymien osalta ongelmat eivät johdu käyttäjälle tarjottujen puhetulosteiden laadusta tai käyttäjän puheen kehnosta konversiosta analogisesta digitaaliseksi. Viimeisimmät tekstistä puhesynteesiä tuottavat sovellukset ovat saavuttaneet erittäin korkean tason puhesynteesin ymmärrettävyydessä [Cox et al., 2000].

### **2.3.3 Puhesynteesi käyttöliittymän tulosteena ja opasteena**

Käyttöliittymän osana puhesynteesi toimii palautteena käyttäjälle. Puhesynteesiä voidaan käyttää joko osassa palautetta, eli esimerkiksi korostamaan käyttäjälle tärkeitä ja olennaisia kohtia, tai siten, että ääni on ainoa tuloste käyttäjälle. Jälkimmäisessä tapauksessa kommunikointi tapahtuu pelkästään puhesynteesillä ja vastaavasti käyttäjän tuottamalla puheella. Puhesynteesin on oltava ymmärrettävää ja miellyttävää, erityisesti silloin kun tietokoneen ja ihmisen vuorovaikutus on riippuvainen puhesynteesistä.

Puhesynteesin hyvän laadun vuoksi ei näyttäisi olevan esteitä sen käyttämiselle myös opasteissa. Siten itse sovellus ja sen käyttötarkoitus ovat etusijalla puhesynteesiteknologiaa valittaessa, koska kaikki toteutettavissa olevat puhesynteesiteknologiat vaikuttavat sopivilta myös opasteisiin. Opasteiden sisältö (kts. luku 2.1) on usein tiedossa jo sovelluksen suunnitteluvaiheessa: proseduraalinen informaatio sisältää sovelluksen toimintaa kuvaavaa tietoa, joka on suunniteltava valmiiksi, kun taas kuvauksellinen tieto kertoo sovelluksen rakenteesta ja komponenteista, joten sekin on muuttumatonta tietoa. Sen sijaan tulkinnallinen tieto, joka opastaa käyttäjää ymmärtämään, miksi sovellus toimii milläkin tavalla eri tilanteissa, saattaisi sisältää dynaamista, esimerkiksi käyttäjän syöttämää tietoa. Kahden ensimmäisen ominaisuuden perusteella suurin osa opasteista voitaisiin yksinkertaisimmillaan toteuttaa valmiiksi nauhoitettuina lausekokonaisuuksina. Tekstinä tallennetut opasteet ovat tietysti muistikapasiteetin kannalta taloudellisempi vaihtoehto varsinkin, kun tekstistä tuotettu synteesikin on jo kelvollisen tasoista. Mutta dynaamisesti koottu tulkinnallinen tieto opasteissa edellyttää monimutkaisempaa puheen prosessointia ja puhesynteesiä luonnollisen puheen tuottamiseksi. Koska opasteen pitäisi myös pystyä vastaamaan käyttäjän kysymyksiin [Goodall, 1991], voidaan pitää mahdollisena, että erityisesti luonnollista puhetta käyttävässä käyttöliittymässä opastetekstitkin voivat olla dynaamisesti koottuja. Tutkimassani aineistossa en tosin törmännyt tämäntyyppisiin opasteisiin.

Miellyttävyys on yksi puhesynteesille asetettu tärkeä vaatimus. Useissa lähteissä käsitellyissä puhekäyttöliittymien evaluoinneissa koekäyttäjät ovat kiinnittäneet huomiota puhesynteesin konemaisuuteen ja metallisuuteen, vaikka sellainen “äänensävy” ei olekaan vaikeuttanut itse käyttöä. Epäilemättä sovelluksen käyttöastetta nostaa sen käytön mukavuus, jolloin puhesynteesi näyttelee tärkeää osaa puhekäyttöliittymässä. Puhesynteesi on kuitenkin iso - ja joskus ainoa - osa sovelluksen ilmiänsä, sitä, jonka käyttäjä ensimmäisenä havaitsee. Jos opastejärjestelmä ei tunnu käyttäjän mielestä miellyttävältä, saattaa koko opastejärjestelmä jäädä käyttämättä.

Yleisin puhekäyttöliittymien rajoittava tekijä on puhetulosteiden hitaus, minkä vuoksi tietokoneen tuottamassa puheessa pitäisi pyrkiä mahdollisimman lyhyeen esitysmuotoon [Schmandt, 1994, s. 106]. Sovellusten suunnittelussa pitäisi olla hyvin perillä siitä mitä tietoa käyttäjä tarvitsee, jotta vältytään tarjoamasta käyttäjälle hitaita ja lisäksi tarpeettomia puhetulosteita. Lisäksi on pohdittava, mitkä asiat esitetään puhetulosteina ja mitkä muilla tavoin. Sovelluksissa, joissa käyttäjä saa tulosteet pääasiassa näytölle, voitaisiin keskittyä antamaan käyttäjälle äänitulosteita vain kun halutaan erityisesti kiinnittää käyttäjän huomio johonkin asiaan. Jos puhe on ainoa vuorovaikutusmuoto sovelluksen ja käyttäjän välillä, on mahdollisuuksien mukaan

suunniteltava käyttäjälle välitön palaute kuhunkin tilanteeseen. Vuorovaikutusta voidaan tehostaa myös puhetulosten nopeuttamisella, lyhentämisellä ja vähentämisellä [Schmandt, 1994, s. 108-109].

Puhekäyttöliittymän suunnittelussa pitäisi mahdollistaa myös puhekäyttöliittymän antaman puhetulosten toistaminen tilanteissa, joissa käyttäjä ei ole saanut tulosteesta selvää tai ei muista useita käyttöliittymän luettelemia vaihtoehtoja [Schmandt, 1994, s. 111]. Toistaminen voisi olla hyvä tehdä myös hidastetusti käyttäjän sitä toivoessa. Epäselvyyksiä voivat aiheuttaa erityismerkinnät, joiden lausuminen vaatii erityisohjeita (tällaisia ovat esimerkiksi sähköpostiosoitteet), selvyyttä saattaa lisätä vielä erityyppisten puheäänien käyttö osoittamaan erilaista käyttötilannetta [Schmandt, 1994, s.111]. Esimerkiksi virheilmoitukset, valikkotekstit tai opasteet voitaisiin lukea erilaisilla äänillä, mikä parantaisi myös käyttäjän mahdollisuuksia hahmottaa käyttötilanne ja sijainti sovelluksessa. Näin esimerkiksi opasteiden antamat ohjeet ja esimerkit eivät voi sekoittua varsinaisessa sovelluksissa vastaan tuleviin samankaltaisiin käyttötilanteisiin.

### **2.3.4 Puheentunnistus**

Puheentunnistus on analogisen signaalin muuttamista sanoiksi, joita voidaan käyttää tietokonesovelluksessa joko sellaisenaan tai käsitellä edelleen puheen ymmärtämiseksi. Puheentunnistus perustuu laskennallisesti tehokkaaseen puheen parametriseen mallintamiseen ja puheen vertaamiseen ennalta tallennettuihin tunnistettavien sanojen malleihin erityisten algoritmien avulla. Laskentatehon parantuessa 1980- ja 1990-luvuilla tilastolliset menetelmät kuten HMM ja neuroverkot johtivat puheentunnistuksen alalla parannuksiin [Klevans & Rodman, 1997, s. 3-4]. Tällä hetkellä parhaat tulokset puheentunnistuksessa saavutetaan kun puhuja erottelee sanat tai tunnistettavat fraasit toisistaan (pitää niiden välissä pienen tauon), kun puheentunnistusjärjestelmä on adaptiivinen (puhujaa samalla sekä käyttää että kouluttaa järjestelmää) ja ympäristön hälyäänet ovat hyvin vähäiset [Klevans & Rodman, 1997, s. 3-4]. Puheentunnistusta kehitetään kuitenkin lähes päinvastaiseen suuntaan: puhetta pitäisi tunnistaa vaikka puhe olisi jatkuvaa, puhujaa tuntematon ja hälyäänet tyypilliset normaalille käyttötilanteelle. Seuraavaksi käsittelemme puheentunnistukseen kuuluvia vaihteita, kuten tunnistuksessa käytettäviä kielimalleja, puheen vaihtelevuutta kuvaavaa perpleksisyyttä ja vaihtelevuuden hallintaan käytettyjä menetelmiä, kuten HMM ja word spotting. Lisäksi tarkastelemme lyhyesti puhujantunnistusta, puheentunnistusjärjestelmän ominaisuuksia ja nykyisten puheentunnistimien tasoa.

Kun puhetta aletaan muokata puheentunnistukselle sopivaan muotoon, se muutetaan ensin analogisesta digitaalisen muotoon ja sen jälkeen viipaloidaan sopiviksi, useimmiten 10-20ms mittaisiksi näytteiksi [Schafer, 1994]. Näitä näytteitä analysoidaan käyttäen apuna akustisia malleja ja kielimalleja. Puhesignaalia käsiteltäessä siitä pyritään erottamaan ensin kaikki sellaiset tekijät pois, jotka aiheuttavat vaihtelevuutta: puhutaan puheen normalisoinnista [Schmandt, 1994, s.87-91]. Selvitettäessä näytteiden merkityksellistä sisältöä verrataan näytettä ennalta tallennettuihin näytteisiin, joiden sisältö on tiedossa. Tästä ennalta tallennetusta datasta käytetään nimitystä koulutusmateriaali [Schafer, 1994].

Kun on löydetty oletetut sanat sekvensseistä, niitä aletaan verrata puheentunnistusjärjestelmässä käytössä olevaan kielimalliin, joita on olemassa useita erilaisia. Yksinkertaisin malli sallii vain tiettyjen sanojen muodostaman tietyn järjestyksen. Yleiset kielimallit, jotka pyrkivät tunnistamaan yhä paremmin luonnollista kieltä, käyttävät tilannekohtaista kielimallia. Kielimallin rakentamiseen käytetään jotakin yleisesti hyväksyttyä korpusta, joka sisältää kattavan otoksen kohderyhmän käyttämistä sanoista. Viime aikojen menestynein kielimalliparadigma on trigram-kielimalli, jossa lasketaan seuraavan tunnistettavan sanan todennäköisyys kahden edellisen sanan perusteella. Kielimalleja voidaan vertailla käyttämällä kielimalleja puheentunnistuksessa ja laskemalla niille virheprosentti, mitä pidetään parhaana tapana, mutta sen käyttö vaatii aikaa ja rahaa. Kielimallin arvioimiseen voidaan käyttää myös lyhyttä testitekstiä. Tätä tapaa soveltaen yhdistämällä sanavaraston koko ja kielimalli saadaan kielimallille mm. perpleksisyysarvo, joka kertoo, miten monimutkainen tunnistustehtävä on kyseessä eli käytännössä, kuinka paljon eri tunnistettavia vaihtoehtoja puheesta saadaan. Perpleksisyysarvo on sitä suurempi, mitä monimutkaisempi malli on kyseessä. [Zue et al. 1996]

Sanaston koon kasvu aiheuttaa usein myös perpleksisyyden kasvun [Cohen & Oviatt 1994]. Sanaston koko riippuu kielen rakenteesta: esimerkiksi englannin kielelle riittää pienempi sanasto kuin kielille, joissa esiintyy paljon taivutusmuotoja [Zue et al. 1996]. Sen sijaan suomen kielessä sanojen taivutusta on paljon, joten sanaston kokokin kasvaa; suomen kielen kanssa onkin kannattavampaa käyttää tunnistusmenetelmää, jossa ei pyritä tunnistamaan kokonaisia sanoja, vaan tunnistettavat yksiköt on jaettu sanaa pienempiin osiin, äänneisiin [Miettinen, 1998].

Puheentunnistus on hankalaa, koska puheessa on paljon vaihtelevuutta. Puheeseen vaikuttavat ensinnäkin foneettiset muuttujat eli äänneiden vaihtelu ympäröivistä äänneistä riippuen. Toiseksi vaihtelua tuovat akustiset muuttujat, eli fyysinen ympäristö ja äänen vastaanottimen (kuten mikrofoni) ominaisuudet ja sijainti. Kolmantena



muuttujana on puhuja itse, tämän mielen- ja terveydentila, puheen tempo ja laatu. Näiden lisäksi vaikuttavat vielä puhujienväliset muuttujat, kuten puhujan käyttämä murre, sosiolingvistinen tausta ja äänenmuodostukseen liittyvien elinten rakenne. Puheentunnistusjärjestelmä yrittää mallintaa puheen vaihtelevuutta monin eri tavoin. Signaalinkäsittelyn tasolla on pyritty kehittämään malleja, jotka korostavat puhujasta riippumattomia ominaisuuksia. Akustisella ja foneettisella tasolla on puhujien vaihtelevuutta pyritty mallintamaan tilastollisin keinoin. Adaptiivinen puheentunnistus taas perustuu siihen, että puhujasta riippumatonta mallia käytetään sopeuttamaan järjestelmä kunkin puhujan puheeseen. [Zue et al. 1996]

Puheentunnistuksen vaihtelevuuden hallinnassa tunnetuin ja käytetyin tilastollinen metodi on HMM, eli *Hidden Markov Model* [Schmandt, 1994, s. 144-147]. Käytettäessä HMM:ää puheentunnistuksessa voidaan tuottaa luettelo vaihtoehtoista tunnistettavalle sanalle, joille kullekin on oma esiintymistodennäköisyytensä. Jos paras vaihtoehto ei ole oikea (joko käyttäjä tai järjestelmä hylkää sen), voidaan kokeilla seuraavaksi parasta vaihtoehtoa. HMM-metodin käyttöön voi kuulua myös parametrimallin säätäminen käytön myötä entistä paremmaksi. Tällä hetkellä suurin osa puheentunnistusjärjestelmistä käyttää puheentunnistuksessa tilastollisia menetelmiä, kuten HMM-metodia [Cox et al., 2000].

Tunnistettaessa puhetta voidaan etsiä siitä ennalta määrättyjä avainsanoja ja päätellä niiden perusteella mitä käyttäjä sanoo. Tätä puheentunnistuksessa käytettävää apukeinoa kutsutaan nimellä *“word spotting”* ja sitä käytetään tunnistettaessa luonnollista puhetta, jossa sanaesiintymien vaihtelevuus on suuri ja puheessa on paljon sanastoon kuulumattomia sanoja. Yksinkertaisimmillaan voidaan etsiä puheesta vain hyvin pieneen osajoukkoon kuuluvia sanoja, jotka esimerkiksi kertovat minkä toiminnon käyttäjä haluaa seuraavaksi valita. Monimutkaisemmassa tunnistuksessa poimitaan puheesta pois vain äännähdykset, jotka eivät varsinaisesti ole sanoja. [Rohlicek et al., 1993]

Puheentunnistuksessa käytetyn puheen parametrisoinnin myötä myös puhujantunnistus on tullut mahdolliseksi. Puhujantunnistukseen kuuluu useita osa-alueita. Yksi osa-alue on kielentunnistus, joka käsittää puhutun kielen tunnistamisen lisäksi myös puhujan murteen, yhteiskunnallisen aseman ja puhujan äidinkielen tunnistamisen. Toisena osa-alueena on puhujien erottelu, jota voidaan tarvita tunnistettaessa vain yhden puhujan ääntä puheensorinasta. Kolmantena osa-alueena mainitaan myös puhujan kasvojen mallintaminen puheen aikana (*lip-synching*, käsittäen huulet, kielen, suun ja leuan). Eniten tutkittu osa-alue on puhujan identifiointi äänen perusteella. Puhujantunnistus ja puheentunnistus ovat toisiaan täydentäviä: siinä missä puheentunnistuksessa on

keskityttävä kielellisten ilmaisujen tunnistamiseen, puhujantunnistus taas pyrkii erottamaan puheesta puhujalle omintakeiset puheen ominaisuudet. Puhujantunnistuksen suurimpia sovellusalueita on tähän mennessä ollut tietoturvan vuoksi vaadittava tietojärjestelmien tai suojattujen alueiden käyttäjien identifiointi. Samoin puhujantunnistusta voidaan käyttää esimerkiksi oikeudenkäynnissä todisteiden oikeellisuuden varmentamisessa. [Klevans & Rodman, 1997, s. 4-6]

Puheentunnistusjärjestelmään liittyy useita erilaisia ominaisuuksia, joita voidaan priorisoida järjestelmän suunnittelussa käyttötarkoituksesta riippuen ja joiden avulla puheentunnistusjärjestelmien tehokkuutta voidaan arvioida. Jotkut vanhimmat puheentunnistusjärjestelmät vaativat, että puhuja erottelee sanat pitämällä pienen tauon sanojen välissä; nykyisistä järjestelmistä suurin osa kykenee tunnistamaan jatkuvaa puhetta. Spontaania puhetta on vaikeampi tunnistaa kuin valmiista tekstistä luettua puhetta, koska spontaani puhe sisältää usein kirjakieleen kuulumattomia ilmaisuja, ääntämistavat vaihtelevat enemmän, tauot puheessa eivät ole ennalta arvattavissa eikä sanoja aina lausuta kokonaisina [Liu et al. 1998]. Puhujasta riippuvassa järjestelmässä käyttäjän pitää ensin antaa äänestään näyte ennen kuin järjestelmä kykenee tunnistamaan puhujan äänellä annetut komennot mahdollisimman hyvin [Zue et al. 1996]. Puhujasta riippumattomassa puheentunnistusjärjestelmässä pyritään tunnistamaan kenen tahansa puhetta mahdollisimman yleisen akustisten parametrien mallin avulla [Schafer, 1994].

Laajan sanavaraston käyttö vaatii tunnistimelta selvästi enemmän. Sellaiset puheentunnistusjärjestelmät, joilla on käytössään vain pieni sanavarasto (ja siten vain vähän samankaltaisia sanoja tunnistettavana) toimivat jo 1990-luvun puolivälissä lähes moitteettomasti, jolloin tunnistusvirheitä tapahtui vain noin yhden prosentin verran kaikesta tunnistuksesta [Schafer, 1994]. Vuonna 2001 isoja sanavarastoja käyttävät järjestelmät ovat päässeet jopa 28 %:n virhemäärään (*WER, word error rate*) [Padmanabhan et al., 2001]. Viime aikoina sekä kaupalliset yhtiöt että tutkimukseen keskittyvät instituutiot ovat ajaneet puheenkäsittelytekniikan kaupallistamista. Puheentunnistimissa on silti vielä kehitettävää: vielä toistaiseksi puheentunnistimille joudutaan usein määrittelemään tarkat käyttöolosuhteet, joissa järjestelmän luvataan toimivan tietyllä tunnistustasolla. Näihin olosuhteisiin kuuluvat mm. taustahälyn taso, puhujasta riippumattomuus, sallitut puhetyylit ja puheen spontaanius [Juang & Furui, 2000].

### **2.3.5 Puheentunnistus käyttöliittymässä ja opasteiden käytössä**

Puhekäyttöliittymien on todettu sopivan käyttötilanteisiin, joissa käyttäjä tarvitsee käsiään tai katsettaan (tai molempia) muuhun tarkoitukseen kuin tietokoneohjelman käyttämiseen [Cohen & Oviatt, 1994]. Esimerkkinä voidaan mainita vuonna 1987 Yhdysvalloissa F-16-lentokoneiden piloteilla tehty tutkimus [Howard, 1987], jonka mukaan pilotit pystyvät suorittamaan lentokoneen ohjaamossa esimerkiksi yksinkertaisia navigointitehtäviä tarkemmin ja nopeammin, jos niissä voidaan käyttää puhetta näppäimistön ja erilaisten ohjainten sijasta. Tämänkaltaisten järjestelmien tunnistustarkkuuden pitää olla hyvin korkea.

Korkeaan tunnistustarkkuuteen päästään helpommin erityisalojen kuin yleisten alojen sovellusten kanssa, koska erityisalojen sanastojen voidaan olettaa olevan suhteellisen suppeita. Esimerkiksi kirurgin sanellessa leikkaustoimenpiteitään puhetta tunnistavalle sovellukselle hän ei luultavasti ala yllättäen puhua esimerkiksi auton polttomoottorin korjauksesta. Näin voidaan luotettavammin päätellä epäselvästi lausuttu sana muuten tunnistetusta lauseesta. Erikoisalojen sovellusten perpleksisyysarvo on siis pienempi kuin yleisten alojen, mikä näkyy vahvimmin sanelusovelluksissa. Opastejärjestelmän sisäisen perpleksisyyden voidaan ajatella riippuvan paitsi varsinaisessa sovelluksessa käytetyn sanaston koosta, myös vuoropuhelun tasosta opastejärjestelmässä. Toisin sanoen jos käyttäjä voi vapaasti muotoilla kysymyksen opastejärjestelmälle, perpleksisyysarvo on varmasti korkea.

Puhesovellukset voidaan jakaa kahteen osaan myös sen mukaan, liittyvätkö ne telekommunikaatiopalveluihin [Juang & Furui, 2000]. Tällaisiin usein laajalla alueella käytössä oleviin puhelimella käytettäviin palveluihin kuuluvan sovelluksen odotetaan olevan puhujasta riippumaton, sillä käyttäjämäärät voivat olla hyvin suuria. Nykyisellä tunnistustarkkuudella telekommunikaatiosovellusten on käytettävä pieniä sanastoja, jolloin myös opastejärjestelmän on oltava yksinkertaisin komennoin käytettävissä.

Varsinkin telekommunikaatiopalveluihin kuulumattomien sovellusten puheentunnistuksessa yksi tärkeä tavoite on pyrkiä tunnistamaan täysin luonnollista kieltä. Tällöin käyttäjä voisi sanoa asiansa ihan miten tahansa, esimerkiksi jättää turhia toistoja pois ja olettaa että kuuntelija tietää mistä on kyse asiayhteyden perusteella ja käyttää erilaisia sanontoja ja kielikuvia (luonnollista kieltä käsiteltiin myös kappaleessa 2.2). Mikään järjestelmä ei kuitenkaan voi tunnistaa luonnollista puhetta täysin rajattomasti. Opasteiden rooli on tässäkin tärkeä: käyttäjä on houkuteltava sanomaan juuri oikeat sanat järjestelmälle. Vaikka moni järjestelmä saattaa olla siinä määrin joustava, että se tunnistaa muutamia synonyymeja ja samaa tarkoittavia fraaseja, koskaan ei voida kattaa kaikkia mahdollisia tapoja. Ongelmaksi voi muodostua se, että käyttäjä sanoo asiansa mielestään täysin selvästi eikä järjestelmä silti ymmärrä sitä. Yhtenä ratkaisuna voidaan

poimia yksittäinen sana käyttäjän syötteestä ja kysyä sille tarkennusta tai yksinkertaisesti pyytää käyttäjää muotoilemaan lauseensa uudestaan [Mané et al., 1996]. Aiemmin vuoropuhelun yhteydessä käsiteltiin vihjeiden esittämistä käyttäjälle, jotka paitsi kertovat käyttäjälle puheenvuoron vaihtumisesta myös ohjaavat käyttäjää puhumaan järjestelmän tunnistamilla sanoilla.

Erityisesti opastejärjestelmän puhetulosteet voivat joskus olla pitkiä. Käyttäjän pitäisi pystyä keskeyttämään sovelluksen puhetuloste, sillä se voi olla käyttäjälle hyödytön tai jo ennestään tuttu. Puheentunnistimelle voi olla ongelmallista käyttäjän keskeytysyrityksen tunnistaminen sovelluksen oman tulosteen seasta. Ratkaisuna on käytetty pienten taukojen liittämistä tulosteeseen, jolloin tulosteeseen kuulumattomat äänet osataan helpommin tunnistaa [Schmandt, 1994, s.109]. Sen sijaan jos käyttäjä keskeyttää tulosteen esimerkiksi äänitaajuuspuhelimien näppäimillä, on keskeytys paljon helpompaa erottaa puheen seasta sen erilaisen akustisen luonteen takia [Schmandt, 1994, s.109]. Mutta jos keskeytys ei ole mahdollista, on sovelluksen puhetulosteiden oltava mahdollisimman lyhyitä [Mané et al., 1996].

Sähköpostien lukemiseen tarkoitettun puhekäyttöliittymän (MailCall) käytettävyydestissä huomattiin, että vaikka kaikilla käyttäjillä oli ongelmia puheentunnistimen virheiden kanssa, opasteita käyttävät pystyivät paremmin pitämään yllä keskustelua. Joissakin tilanteissa puhelimen näppäinten käyttäminen komentojen antamiseen oli tosin nopeampi tapa kuin itse puhuminen, sillä MailCallissa on suuri sanavarasto, mikä hidastuttaa puheentunnistinta. Käytettävyydestin perusteella aloitteleville käyttäjille tarjottavia tulosteita pidennettiin, jotta niihin saatiin selkeämpää ja tarkempaa informaatiota. Tulosten perusteella suureenkin sanavarastoon ja vain puheeseen perustuva käyttöliittymä voi olla käytettävä ja hyödyllinen. [Marx & Schmandt, 1996]

Matkapuhelimien yleistyttyä puheohjatut käyttöliittymät voivat olla uusi askel tietoyhteiskunnan kehityksessä, jossa tiedon hajauttamista on pidetty taloudellisena ja tehokkaana ratkaisuna. Puheohjattu käyttöliittymä lisää tavoitettavuutta ja laajentaa mahdollisten käyttötilanteiden valikoimaa. Jos puheohjattujen käyttöliittymien määrä tämän vuoksi kasvaa, saattaa puhe-sovellusten opasteiden käyttö ja tarve entistä paremmille opasteille kasvaa muistettavien komentojen määrän kasvaessa.

### 2.3.6 Virheisiin varautuminen ja varmentaminen

Puhekäyttöliittymän käytössä yksi yleisimmistä ongelmista liittyy tunnistamisvaikeuksiin. Puheentunnistuksessa törmätään kolmenlaisiin virheisiin:

- käyttäjän puhumaa sanaa ei tunnisteta (*rejection*),

- puhuttu sana tunnistetaan joksikin muuksi kuin miksi se oli tarkoitettu (*substitution*) tai
- käyttäjän muut äänet kuten hengittäminen ja toisaalta taustalta kuuluvat äänet tunnistetaan virheellisesti sanoiksi (*insertion*).

Vaikka ongelmia syntyykin puheentunnistuksessa helposti, on ongelmien ehkäisemiseen useita tapoja. Ensinnäkin sellaisissa käyttöliittymissä, joissa käsitellään puhetta vain komentoina eikä vastaanoteta vapaasti muotoiltua sanelua, voidaan käytössä olevan sanavaraston hyvällä suunnittelulla välttyä useilta hankaluuksilta. Avainsanojen pitäisi selkeästi erota toisistaan, ja lisäksi voidaan kehittää tilannekohtaista sanavarastoa, joka tiettyssä tilanteessa tunnistaa vain tietyt sanat. Molemmat keinot vähentävät tunnistusvirheen mahdollisuutta. Lisäksi puhekäyttöliittymässä voidaan käyttää käyttäjäkohtaista tunnistusmallia, joka on ”herkistetty” kunkin käyttäjän omalle äänelle ja jolla pyritään myös välttämään tunnistusvirheitä. Käyttötilanteessa vallitsevia olosuhteita voidaan yrittää optimoida: mahdollisimman vähäinen taustahäly ja ääntä siirtävien laitteiden (kuten mikrofonin) laatu parantavat tunnistamista. Myös käyttäjän selkeä artikulaatio vähentää puheentunnistuksessa tehtäviä virheitä. [Schmandt, 1994, s. 160-165]

Virheiden tunnistaminen puhekäyttöliittymässä voi tapahtua joko käyttäjän tai sovelluksen toimesta. Yleensä käyttäjä on vastuussa virheiden tunnistamisesta [Mané et al., 1996], mutta myös sovellus voi jonkin verran tarkastella käyttäjän antamia syötteitä niiden kieliopin tai järkevyyden osalta: näistä jälkimmäisen tarkistaminen vaatii sovellukselta varsin monimutkaisen päättelyjärjestelmän [Schmandt, 1994, s. 166-169]. Käyttäjän tekemää tarkistusta voidaan helpottaa toteuttamalla toistamistoiminto, jolla sovellus pyynnöstä toistaa saamansa syötteen. Sovellus voi myös pyytää käyttäjää toistamaan koko syötteen tai osan siitä, esimerkiksi ”kaiuttamalla” tunnistetun osan syötteestä ja kysymällä tunnistuksessa hylätyn osan uudestaan: ”Mihin aikaan haluat kutsua x:n ja y:n palaveriin maanantaina?” [Schmandt, 1994, s. 166-169]. Käyttäjää voidaan pyytää muotoilemaan asiansa uudestaan, jos järjestelmällä on syytä epäillä virheellistä tunnistusta tai järjestelmä ei tunnistanut käyttäjän komentoa, mutta tämäntapainen ohje voi olla käyttäjälle liian hankala ja epätarkka [Mané et al., 1996]. Jos käyttäjä toistaa lauseen täysin samalla tavalla, todennäköistä on, että tunnistusvirhe tapahtuu uudestaan. Tällaisessa tilanteessa ei ole järkevää antaa käyttäjälle kerta toisensa jälkeen samaa virheilmoitusta, vaan parin lyhyen virheilmoituksen jälkeen lisätä virheilmoitukseen tarkempia ohjeita [Yankelovich et al., 1995]. Käyttäjää voi myös muistuttaa, ettei pidä liioitella puhetta vaan puhua ihan normaalisti.

Järjestelmissä, joissa on käytössä todennäköisyyslaskentaan perustuva tunnistusmenetelmä (esimerkiksi Hidden Markov Model), tunnistuksen yhteydessä

saadaan usein lista mahdollisista oikeista sanoista. Kun listan ensimmäinen sana on todettu virheelliseksi, voidaan ottaa listalta seuraavaksi todennäköisin vaihtoehto ja kokeilla sitä. Virheentunnistuksen kannalta oppiva järjestelmä on oivallinen, sillä se tallentaa käyttäjän tekemät korjaukset eikä tee samaa virhettä kahdesti (käyttäjän antama ääninäyte on yhdistetty oikeaan sanaan). [Mané et al., 1996]

Puhekäyttöliittymän virhetilanteita voidaan välttää hyvällä käyttöliittymän suunnittelulla. Jokaiselle toiminnolle voidaan pyytää käyttäjältä varmennus, joko eksplisiittisesti tai implisiittisesti. Eksplisiittinen varmennus tarkoittaa kysymyksen esittämistä käyttäjälle: “Oletko nyt varma että haluat...” johon käyttäjä voi vastata “kyllä” tai “ei” ja siten välttää tekemästä vääriä toimintoja. Implisiittisen varmennuksen voi toteuttaa esimerkiksi esittämällä käyttäjälle, mitä seuraavaksi aiotaan tehdä ja pitämällä sen jälkeen pienen tauon, jotta käyttäjällä on aikaa tarkistaa ja mahdollisesti perua toiminto. Periaatteena voidaan pitää toimintojen luonteen kahtiajakoa: toiminnot, jotka koskevat tiedon esittämistä, voidaan varmentaa implisiittisesti, kun taas toiminnot, jotka joko tuhoavat tietoa tai muuten tekevät peruuttamattomia toimenpiteitä, varmennetaan eksplisiittisesti. [Schmandt, 1994, s.166-169]

Implisiittistä varmennusta kannattaa käyttää myös välittömänä palautteena käyttäjälle. Jos järjestelmä alkaa suorittaa annettua komentoa antamatta käyttäjälle mitään palautetta, käyttäjä ei välttämättä tiedä, tapahtuuko sovelluksessa jotain. Graafisissa käyttöliittymissä käytetään tällaisessa tilanteessa usein välittömänä palautteena tiimalasin kuvaa cursorin paikalla, jota puhekäyttöliittymässä voisi vastata lyhyt äänituloste tai merkkiääni. Välittömällä palautteella pystytään luomaan miellyttävän vuorovaikutuksen tunnelma. [Kamm & Walker, 1997]

Ongelmia saattaa syntyä, kun käyttäjä ei tiedä, mitä hän voi sovellukselle sanoa: joko käyttäjä ei tiedä joitakin tarpeellisia toimintoja tai hän yrittää antaa sovellukselle sellaisia syötteitä, joita sovellus ei tunnista ollenkaan. Paras tapa tällaisten virheiden ehkäisyyn on esittää käyttäjälle mahdollisimman selkeästi kussakin tilanteessa käytettävät komennot [Yankelovich, 1996]. Luonnolliseen vuoropuheluun pyrkivissä käyttöliittymissä komentojen luettelointi on hankalaa, mutta toisaalta puheentunnistusjärjestelmä pystyy usein tunnistamaan useita erilaisia ilmaisuja samassa tilanteessa.

Täysin vastaavaan ongelmaan törmätään opasteiden maailmassa. Usein käyttäjän pitää itse tietää, mitä tietoa hän opasteista tarvitsee. Ongelmaa ei kuitenkaan ole aina helppo pukea sanoiksi, varsinkin jos käyttäjän tietomäärä ei riitä esimerkiksi käyttöliittymän eri osien tai operaatioiden nimeämiseen. Oikean opasteen löytäminen voi näin olla

melkoista seikkailua opasteviidakossa. Opastejärjestelmän tilannekohtaisuus auttaa tähän pulmaan, koska silloin järjestelmä on jollakin tasolla tietoinen käyttäjän aiemmista toimista sekä käyttöliittymän senhetkisestä tilasta. Tilannekohtaisuuden onnistunut toteuttaminen vaatii kuitenkin paljon työtä ja voi olla hyvinkin vaikeaa [Duffy et al., 1992, s. 118-119]. Epäonnistunut tilannekohtaisuuden toteutus tai sen puuttuminen kokonaan aiheuttavat ongelman tarjottavan tiedon laadussa ja määrässä. Tiedon paljoudesta käyttäjän on hankalaa löytää etsimänsä, toisaalta väärin menneet kontekstin arvaus tuo näkyville epärelevanttia tietoa.

#### **2.4 Puhekkäyttöliittymien tulevaisuus**

Käytettävyyssiantuntija Shneiderman [Shneiderman, 2000] väittää, että puhekkäyttöliittymät eivät sovi joka paikkaan, esimerkiksi auto-onnettomuuksien määrä on kasvanut kännyköiden käytön lisääntyessä, joten sähköpostien kuunteleminen autoa ajaessa ei ole välttämättä hyvä sovelluskehityksen alue. Tähän annetaan selityksenä se, että monimutkaista aivotyötä vaativien tehtävien suorittaminen sujuu parhaiten hiljaisuudessa, koska sellaisessa tehtävässä kuormittuva työmuisti sijaitsee samassa aivojen osassa, joka käsittelee puhetta ja kuuntelemista. Fyysisiä toimintoja sen sijaan hallitaan aivojen eri alueella, joten yhtäaikaisen puhumisen ja kävelemisen väitetään onnistuvan paremmin kuin puhumisen ja ajattelemisen. Samoin tietokoneen käyttäminen (kuten hiiren ohjaaminen) sujuu paremmin kuin puhuminen yhtäaikaisesti ajattelun kanssa, mikä on osoitettu samassa lähteessä käyttäjätestein.

Puhesovelluksien asiantuntija Lai [Lai, 2001] puolestaan on sitä mieltä, että tulevaisuuden tietokoneet pystyvät jäljittelemään ihmisten välistä keskustelua täydellisesti. Tällä hetkellä hankalimmalta asialta vaikuttaa keskustelun kulun jäljitteleminen: millä perusteella ja miten usein aiheet vaihtuvat toisiksi. Lai sanoo, että sellaisia puheenkäsittelyjärjestelmiä ollaan jo kehittämässä, jotka pyrkivät tunnistamaan ihmisen puheen todellisen sisällön tunnetiloineen kaikkineen. Hän muistelee keskustelua, jossa puheteknologiategollisuuden edustaja kertoi sekä hyviä että huonoja uutisia puheteknologian tulevaisuudesta verraten sitä television science fiction ohjelma Star Trekiin: Huonot uutiset ovat, että Star Trekissä esitetty hyvin edistyksellinen puheteknologia on asettanut ihmisten odotukset puheteknologiaa kohtaan hyvin korkealle. Hyvät uutiset ovat, että meillä on muutama sata vuotta aikaa päästä Star Trekin puheteknologian tasolle, ja siihen nähden olemme reilusti edellä aikataulusta.

Näinkin ristiriitaisten näkymien rinnalla opastejärjestelmien tulevaisuus näyttää selkeämmältä. Päämääränä on palvella käyttäjän tarpeita juuri riittävällä määrällä

tietoa, jota käyttäjä tarvitsee tehtävänsä suorittamisessa. Seuraavassa luvussa tavoitellaan tätä päämäärää puhekäyttöliittymän opasteiden suunnitteluohjeiden kanssa.



### 3. Puhekäyttöliittymän opasteiden suunnittelu

Puhekäyttöliittymän suunnittelussa tulisi keskittyä kolmeen seikkaan: suoritettavan tehtävän asettamiin vaatimuksiin, teknologian asettamiin vaatimuksiin ja sen tuomiin mahdollisuuksiin ja käyttäjäkunnan ominaisuuksiin. Mitä luonnollisempi vuorovaikutus käyttäjän kanssa saadaan aikaan, sitä paremmin saadaan käyttöön myös käyttäjän kyvyt ja taidot, joita hän on oppinut normaalissa jokapäiväisessä kommunikoinnissa [Kamm, 1994].

Tässä luvussa käsitellään edellä mainituista kolmesta seikasta pääasiassa ensimmäistä, eli opastejärjestelmään liittyvien tehtävien asettamia vaatimuksia ja niiden toteuttamiseksi annettuja ohjeita. Näitä ohjeita sovelletaan sähköpostin lukemiseen ja hallintaan tarkoitetun Postimies-ohjelman opasteiden suunnitteluun. Teknologian asettamia vaatimuksia on käsitelty luvussa 2 ja ne otetaan myös huomioon opasteiden suunnittelussa. Ennen opastejärjestelmän suunnitteluohjeiden käsittelyä perehdytään yleisesti Postimies-ohjelman arkkitehtuuriin, käyttöön ja käyttäjäkuntaan (aliluku 3.1) sekä pohditaan, milloin käyttäjä tarvitsee opasteita ja miksi (aliluku 3.2).

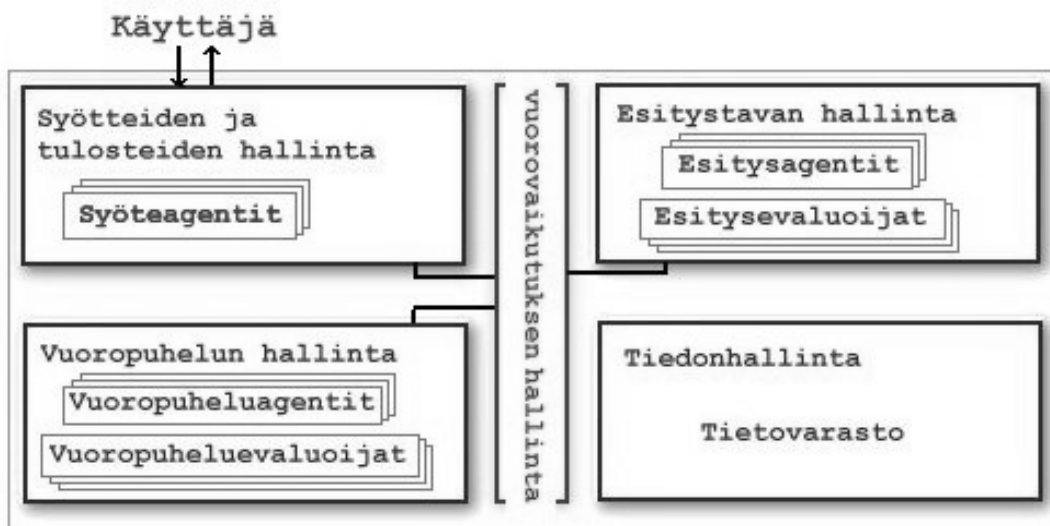
#### 3.1 Postimies

Postimies on Tampereen yliopiston Tietojenkäsittelytieteiden laitoksella tutkimusryhmässä *Speech-based and pervasive interaction* kehitetty puheella ohjattava sähköpostin lukemiseen tarkoitettu ohjelma. Projektissa on tutkittu puhtaasti puheeseen perustuvan käyttöliittymän suunnittelua ja toteutusta. Sähköpostin käsittelyn osalta on tutkimuksen kohteena ollut suurten sähköpostimäärien hallinta ja sähköpostiviestin sisältämän tekstin käsitteleminen syntetisaattorille sopivaan muotoon. Prosodian käyttöä puhesynteesissä ja suomenkielisyyden ja monikielisyyden vaatimia ominaisuuksia on myös tutkittu ja kehitetty.

Seuraavassa aliluvussa (3.4) esitellään suunnitelma yhdestä mahdollisesta Postimiehen opasteiden toteutuksesta samalla kun tarkastellaan puhekäyttöliittymän opasteiden suunnitteluun sopivia ohjeita. Koska Postimies on jo olemassa oleva sovellus, sen ohjelmistoarkkitehtuuria pyritään noudattamaan mahdollisimman tarkasti pitäen myöhemmin tehtävä toteutus- ja testaustyö mielessä. Myös olemassa olevaan käyttäjädokumentaatioon on syytä tutustua, sillä dokumentaation on toimittava yhdessä opasteiden kanssa ongelmitta [Goodall92].

Postimiestä käytetään puhelimitse sähköpostiviestien kuuntelemiseen ja sähköposti-laatikon hallintaan. Postimies tunnistaa puhetta ja vastaa puhesynteesillä, lisäksi puhelimen näppäimiä voi käyttää järjestelmän ohjaamiseen. Postimiehen syntetisaattoriin on toteutettu sekä mies- että naisääni eri kieliä varten. Käyttäjien kokemukset kahdesta eri äänestä ovat olleet positiivisia [Turunen & Hakulinen, 2000b]. Opasteet ovat luonteeltaan hieman erilaiset kuin järjestelmän muut tulosteet, joten voitaisiin ajatella, että opasteen lukisi toisenlainen ääni kuin muun sovelluksen tulosteet.

Postimiehen sovellusarkkitehtuurin pohjalla on eräs adaptiivisten puhekäyttöliittymien sovelluskehys, Tampereen yliopistossa kehitetty Jaspis. Jaspis tukee sekä tilannekohtaisuutta että monikielisyttä. Adaptiivisuus toteutuu Jaspiksessa järjestelmäkomponenttien tasolla: sekä syötteiden käsittely, dialoginhallinta että tulosteiden muodostaminen on mahdollista suorittaa käyttäjän tarpeisiin ja käyttötilanteeseen sopeutuen [Turunen & Hakulinen, 2000a]. Kuvassa 2 on esitetty Jaspiksen tarjoama sovelluskehys yleisellä tasolla.



Kuva 2: Jaspis-sovelluskehys

Jaspiksessa on toteutettu adaptiivisuus agenteilla (*agents*) ja evaluoijilla (*evaluators*). Nämä järjestelmäkomponentit voivat olla erikoistuneita eri tilanteisiin, kuten esitystavan tai vuoropuhelun hallintaan. Jaspiksen evaluoijat valitsevat kuhunkin tilanteeseen sopivan agentin. Jokaisella agentilla on profiili, joka kertoo sen ominaisuuksista. Evaluoija vertaa agentin profiilia käyttäjän profiiliin valitessaan sopivaa agenttia. [Turunen & Hakulinen, 2000a]

Jaspiksen esitysagentit (*presentation agents*) muuttavat konseptuaalisen tiedon puhetulosteeksi valiten samalla äänen, jolla puhe tulostuu. Postimieheissä on käytössä useita esitysagentteja. Yksi agentti esimerkiksi huolehtii viestien lukemisesta. Yhdellä

tehtävällä voi olla useita agenteja, jotka eroavat toisistaan esimerkiksi tulosteen esitystavassa tai tulosteen sisällön pituudessa. [Turunen & Hakulinen, 2000b]

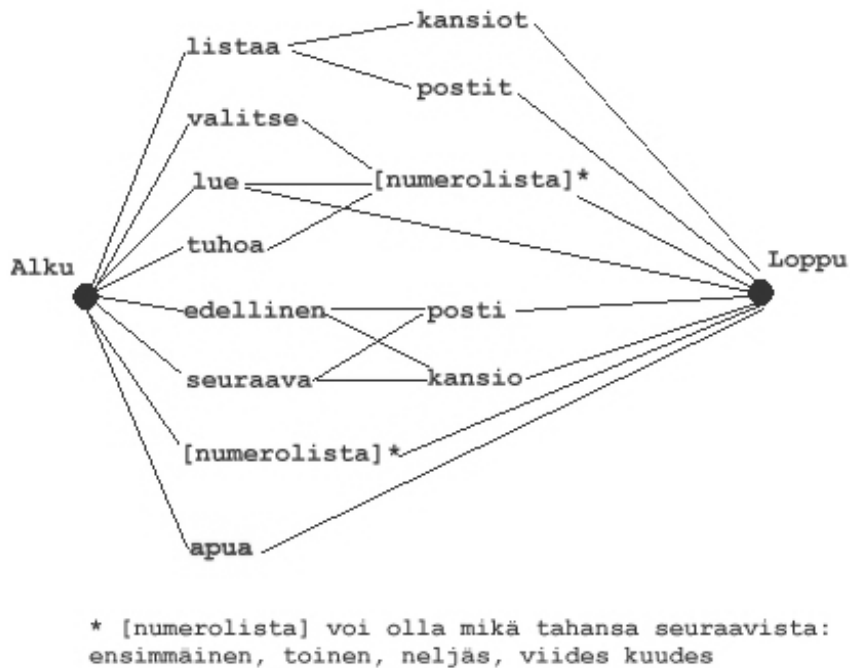
Agenttipohjaisissa sovelluksissa käytetään erilaisia agenteja suorittamaan erilaisia tehtäviä. Yksi agenteista voi olla erikoistunut opasteiden sisällön ja ajoituksen hallintaan (esimerkiksi sovelluksessa *Artificial Teacher* [Masthoff, 1997]). Postimiehessä nämä tehtävät kuuluvat evaluoijalle, esitystavasta vastaa esitysagentti. Opasteiden lukemiseen sopisi vähintään yksi oma esitysagentti. Jos halutaan ottaa toteutuksessa huomioon useat erilaiset käyttäjäryhmät, voidaan kullekin ryhmälle suunnitella oma esitysagentti. Tällöin voitaisiin kokeneille käyttäjille suunnitella hyvin lyhyesti puhuva agentti ja uusille käyttäjillä hieman enemmän kertova agentti. Jos halutaan mennä vielä pidemmälle, myös käyttäjäkohtaisia ominaisuuksia voitaisiin ottaa huomioon, kuten jos käyttäjä on tyypillisesti tarvinnut apua esimerkiksi pitkän hiljaisuuden jälkeen, tälle käyttäjälle tarjottaisiin opastetta nopeammin hiljaisuuden koittaessa.

Kaikki Postimies-järjestelmän komponentit tallentavat kaiken tietonsa samaan paikkaan, tietovarastoon (*information storage*), mikä mahdollistaa jokaisen järjestelmän komponentin pääsyn kaikkeen järjestelmässä olevaan tietoon [Turunen & Hakulinen, 2000a]. Opasteiden kannalta tämä arkkitehtuurinen ratkaisu on erinomainen, sillä käyttäjälle esitettävää opastetta valittaessa on kaikesta käyttäjään ja kyseiseen käyttökertaan liittyvästä tiedosta hyötyä.

Postimiestä käytettäessä käyttäjän on ensimmäiseksi kirjauduttava sisään Postimies-järjestelmään. Kun käyttäjä on tunnistettu, järjestelmä ottaa yhteyden käyttäjän sähköpostilaatikkoon ja hakee sieltä sähköpostiviestien otsikot. Tämän jälkeen postit lajitellaan automaattisesti maksimissaan kuuden viestin kansioihin esimerkiksi aiheen, lähettäjän tai saapumisajankohdan perusteella. Seuraavaksi käyttäjälle kerrotaan muutamalla sanalla kunkin kansion sisältö ja käyttäjä voi sitten valita kansioista mieleisensä. Kun käyttäjä valitsee yhden viestin, järjestelmä lukee viestin käyttäjälle osissa, joiden välillä käyttäjä voi navigoida vapaasti. Käyttäjä voi lukea viestejä niiden järjestysnumeron perusteella tai käyttämällä sanoja edellinen tai seuraava. Postimiehessä voi myös tuhota viestejä, mutta vastaamismahdollisuutta ei vielä ole. [Turunen & Hakulinen, 2000b]

Postimiehessä on kolmenlaisia puhetulosteita: järjestelmän tulosteita, näkymiä sähköpostiviesteihin ja itse sähköpostiviestit [Turunen & Hakulinen, 2000b]. Opasteet kuuluvat selvästi ensimmäiseen ryhmään, jonka toteuttaminen on syntetisaattorin kannalta kaikista helpointa, sillä opasteiden sisältö tiedetään jo ennalta.

Postimiehelle on kirjoitettu melko lyhyet käyttöohjeet. Ohjeista löytyy tietoa sisäänkirjautumisesta, puheentunnistuksesta (mukaan lukien kaavio tunnistettavista sanoista ja sanayhdistelmistä, ks. kuva 3) ja puhelimen näppäimien käytöstä. Ohjeet ovat lyhyet ja selkeät, joten samaa linjaa on hyvä noudattaa myös opasteissa. Sanaston esittämistä graafina tosin ei voida suoraan hyödyntää, vaan opasteisiin on suunniteltava sanaston esittäminen uudelleen.



Kuva 3: Postimiehen tunnistamat sanat ja sanayhdistelmät. Lukusuunta on vasemmalta oikealle.

### 3.2 Miksi ja milloin käyttäjä tarvitsee opasteita

Ennen Postimiehen opasteiden suunnittelemista on selvitettävä, missä tilanteessa käyttäjä saattaisi tarvita apua. Tietokoneen käyttö voidaan jakaa esimerkiksi Don Normanin seitsemään askeleen ihmisen ja tietokoneen välisessä vuorovaikutuksessa: päämäärän muodostaminen, aikomuksen suunnitseminen, toiminnan määrittäminen, toiminnan toteuttaminen, järjestelmän tilan havainnoiminen, järjestelmän tilan tulkitseminen ja tuloksen arvioiminen [Norman, 1990, 187-219]. Opasteiden käytön kannalta toiminnan määrittäminen ja järjestelmän tilan tulkitseminen ovat kriittisiä kohtia, joissa käyttäjä saattaa tarvita opasteita. Toiminnan määrittämiseksi käyttäjä voi Normanin mukaan joko muistaa toimintatavan edellisistä käyttökokemuksista, selvittää toimintatavan kokeilemalla ja päättämällä tai opetella asian toisen henkilön tai dokumentaation avulla [Norman, 1990, 187-219]. Vaikka Norman viittaa ohjeessaan

dokumentaatioon, on opasteiden osa ilmeinen selvitetessä, mitä milläkin toiminnolla tehdään. Käyttäjän toimintaa seurattaessa pitää pyrkiä tunnistamaan yleisimmät umpikujat, joihin käyttäjä saattaa joutua ja opasteita suunniteltaessa tarjota tarkka, suppea ratkaisujoukko näihin tilanteisiin. Toisin sanoen mahdolliset käyttäjän esittämät kysymykset pitäisi saada kartoitetuksi sekä löytää asiayhteydet, joissa näihin kysymyksiin törmätään.

Käyttäjiä seurattaessa on selvinnyt, että he kääntyvät opasteiden puoleen tyypillisesti silloin, kun ovat joutuneet ongelmatilanteeseen suorittamansa tehtävän parissa. Käyttäjät eivät hae apua opasteista halutessaan tietoa sovelluksen käyttökonseptista tai monimutkaisista ja laajoista aiheista. Silloin he tyypillisimmin lukevat käyttöohjeita ja muuta dokumentaatiota. [Goodall92]

Puhelinsovellusten standardointikomitea [TSSC, 2000] jakaa puhelinsovelluksen käyttäjän ongelmatilanteet kolmeen eri tyyppiin: käyttäjä on oikeassa paikassa sovellusta, mutta tarvitsee selvennystä toimintoihin; käyttäjä on väärässä paikassa sovellusta ja tarvitsee apua navigoinnissa; käyttäjä haluaa poistua sovelluksesta. Ratkaisuksi ensimmäiseen ehdotetaan opasteen tarjoamista tai edellisen syötteen esittämistä uudelleen, toiseen sovelluksen päätasolle palaamista tai edellisen toiminnon peruuttamista ja kolmanteen kehotusta ottaa yhteyttä asiakaspalveluhenkilöstöön tai mahdollisuutta sanoa ”goodbye” ja lopettaa puhelu. Myöhemmin artikkelissa edellisen toiminnon peruuttamismahdollisuudesta luovutaan, koska peruuttavat toiminnot voidaan ymmärtää eri tavoin tai ne eivät ole välttämättä mahdollisia. Eksplisiittistä lopetusta pidetään em. lähteessä ehdottoman tärkeänä, koska silloin käyttäjä voi varmistua siitä, että kaikki hänen syöttämänsä tiedot on rekisteröity. Pelkkä puhelun katkaiseminen jättää käyttäjälle epävarman tunteen operaation onnistumisesta.

Edellä esitellyn perusteella opastetta voitaisiin käyttää luontevimmin tilanteessa, jossa käyttäjä on tietoinen tilanteesta ja sijainnista, jossa hän sovelluksessa on, mutta haluaa tietää, mitä hän voi tehdä tai mitä ovat syyt ja seuraukset senhetkiselletilanteelle. Avun saamiseksi käyttäjä voisi luonnollisesti sanoa ”apua”, mutta toinen vaihtoehto on, että sovellus tunnistaa käyttäjän ongelmatilanteen ilman käyttäjän eksplisiittistä avunpyyntöä. Tällaisia opasteen käynnistäviä tilanteita voisivat olla käyttäjän hiljaisuus, käyttäjän antama virheellinen kommento tai käyttäjän antama tunnistamaton kommento. Viimeisessä näistä tapauksista saattaisi olla sopivaa, jos järjestelmä pyytäisi käyttäjää toistamaan, sekä samalla tai seuraavassa tulosteessa antaisi vihjeen, esimerkiksi: ”Voisitko toistaa? Sano jos tarvitset apua.” Tässä tulosteessa on yhdistetty virheenkorjaus ja opasteeseen ohjaava vihje. Opasteeseen ohjaava vihje sopii liitettäväksi myös implisiittiseen ja eksplisiittiseen varmennukseen (kts. luku 2.3.6). Ero

vihjeen ja varmennuksen välillä on tapahtumaan liittymisessä: vihje kertoo mahdollisuudesta tehdä jotain, varmennus jostakin, jota ollaan jo aikeissa tehdä. Molemmat ovat virhetilanteisiin varautumiseen kuuluvia toimintoja.

Pelkästään puheeseen perustuvassa käyttöliittymässä navigoiminen on vaikeampaa kuin visuaalisessa käyttöliittymässä, jossa tilallinen tieto on helpommin hahmotettavissa [Arons, 1991]. Suunniteltaessa sovelluksessa navigointia on otettava huomioon, että käyttäjä ei tunne yhtä helposti olevansa eksyksissä, jos hän saa seurata koko ajan valmista polkua, kuin jos vapaa navigoiminen on mahdollista [Zellweger, 1989]. Pelkkään puheeseen perustuvassa rakenteeltaan hierarkkisessa käyttöliittymässä navigointi muistuttaa monilta osin hypertekstissä navigointia.

### **3.3 Puheikäyttöliittymän ja opasteiden suunnitteluohjeet ja soveltaminen**

Käyttöliittymän suunnittelusta on kirjoitettu paljon erilaisia suunnitteluohjeita ja -periaatteita ja selvitetty miten käyttöliittymiä ja niiden käytettävyyttä pitäisi testata. Erityisesti puheikäyttöliittymien suunnittelusta on myös kirjoitettu jonkin verran, kuten opasteiden suunnittelustakin, mutta puheikäyttöliittymien opasteet ovat jääneet lähes täysin vaille huomiota. Seuraavassa kootaan yhteen puheikäyttöliittymille ja opasteille annettuja suunnitteluohjeita. Suunnitteluohjeet konkretisoidaan soveltamalla niitä olemassa olevan sähköpostien lukemiseen tarkoitettun ohjelman opasteiden suunnitteluun.

Tärkeimpiä tämän suunnittelutyön osa-alueita ovat opastejärjestelmän käynnistymisen, opasteiden valitsemisen, opastejärjestelmässä navigoimisen ja opasteista poistumisen määrittely ja itse opasteiden esitystavan ja sisällön suunnittelu. Kullekin sanastosta löytyvälle komennolle suunnitellaan oma opasteensa. Sen lisäksi suunnitellaan proseduraaliset opasteet, jotka kertovat, miten mikin toimintoketju suoritetaan. Kaikki virhetilanteet käydään läpi siltä varalta, että käyttäjä tarvitsee niissä erityisiä opasteita. Myös sellaiset tilanteet, joissa järjestelmä itse suorittaa operaatioita virheistä toipumiseksi, saattavat vaatia omat opasteensa siltä varalta, että käyttäjä ei pysty muuten selvittämään, mitä järjestelmässä tapahtuu.

Puheikäyttöliittymälle asetettu vaatimus on, että sellaiset operaatiot, joita käyttäjä tyypillisimmin suorittaa, pitäisi suunnitella mahdollisimman helpoiksi ja kivuttomiksi [Mané et al., 1996]. Yksi tällainen operaatio on opasteiden käyttö, joka voidaan jakaa edelleen pienempiin operaatioihin suunnittelun helpottamiseksi. Duffy, Mehlenacher ja Palmer [Duffy et al., 1992, s. 69-120] määrittelevät kahdeksan käyttäjän sekundääristä toimintoa opastejärjestelmän parissa:

1. ongelman havainnoiminen ja ymmärtäminen,
2. opasteen avaaminen tai opastejärjestelmän käynnistäminen,
3. oikean opasteotsikon valitseminen,
4. informaation läpikäyminen, jolloin tarkistetaan vastaako aihe etsittyä asiaa,
5. informaation ymmärtäminen,
6. oikean tiedon valitseminen,
7. uuteen aiheeseen navigoiminen, mikäli ensimmäisen aihe ei vastaa etsittyä ja
8. tiedon soveltaminen varsinaisessa tehtävässä.

Tyypillisimmät, konkreettiset operaatiot, joiden suoritusta voidaan mielestäni helpottaa ovat opastejärjestelmän käynnistäminen, oikean opasteen valitseminen, opasteen kuunteleminen tai lukeminen, uuteen opasteeseen navigoiminen, opastejärjestelmästä poistuminen ja opasteen soveltaminen ongelmatilanteeseen. Edellä mainittujen toimintojen pohjalta kirjoittajat esittävät melko tarkan kahdeksan kohdan suunnittelumallin opasteille. Mallissa painotetaan erityisesti käyttäjäryhmien määrittelemistä ja niihin tutustumista, sovelluksen ja opasteiden rajapinnan huolellista suunnittelua, käyttäjäystävällistä ulkoasua ja asettelua, sekä navigointiin ja sovellettavuuteen panostamista. Puhekäyttöliittymän suunnittelu voi osittain sujua noudattamalla graafiselle käyttöliittymälle tarkoitettuja suunnitteluohjeita, mutta osa puhekäyttöliittymän ominaisuuksista vaatii erityistä suunnittelua. Duffy ja muut ovat osoittaneet ehkä osuvimman vaatimuksen opastejärjestelmälle: opastejärjestelmän pitää olla todella avuksi eikä tuntua välttämättömältä pahalta, jotta käyttäjät hyväksyvät sen ja todella käyttävät sitä [Duffy et al., 1992, s.117].

Seuraavassa on käsitelty pääasiassa sekä Duffyn ja muiden opasteille tarkoitettuja suunnitteluohjeita ja edellä mainittuja toimintoja, että Manén ja muiden esittelemiä puhekäyttöliittymän suunnitteluohjeita siten, että olen soveltanut kutakin ohjetta opasteiden suunnitteluun. Duffyn ja kumppanien kahdeksassa suunnitteluohjeessa opasteiden kirjoittaminen oli jaettu kahdeksi kohdaksi, jotka yhdistin yhdeksi. Lisäsin myös yhden uuden kohdan, opastejärjestelmästä poistumisen, joka ei ole mielestäni puhekäyttöliittymissä täysin triviaalia. Lisäksi olen ottanut huomioon useita muita opasteille ja puhekäyttöliittymille suunnattuja ohjeita sekä soveltanut aiemmin tässä työssä esitettyjä huomioita ja faktoja. Kutakin ohjetta käytetään Postimiehen opasteiden suunnitteluun soveltuvilta osin.

### **3.3.1. Käyttäjän tarpeet**

Aluksi selvitetään, mitä käyttäjät tarvitsevat opastejärjestelmältä. Esimerkiksi haastatellaan testikäyttäjiä tai tarkkaillaan käyttötilannetta, jotta saadaan selville millaisia kysymyksiä käyttäjät haluaisivat esittää opastejärjestelmälle [Duffy et al.,

1992, s. 61]. Ohjelmistokehitystä suositellaan nykyään tehtäväksi iteratiivisesti, jolloin ensimmäistä prototyyppiä tai toteutusta evaluoidaan ja sen jälkeen korjataan palautteen perusteella ohjelmiston virheitä ja ongelmakohtia. Tässä työssä ei käsitellä varsinaisesti ohjelmistotuotantoa, mutta mainittakoon, että opastejärjestelmään pätevät samat ohjelmistonsuunnitteluprosessin säännöt kuin muidenkin ohjelmistojen suunnitteluun, erityisesti niiltä osin, joissa tehdään ohjelmistokehitystä käyttäjän kanssa ja ehdoilla.

Jos käyttäjäkyselyitä tai -testejä ei syystä tai toisesta voida tehdä tai opasteita suunnitellaan valmiiseen sovellukseen, voidaan sovelluksen käytössä mahdollisesti ilmeneviä kysymyksiä lähteä selvittämään sovelluksen toiminnallisista ja sisällöllisistä määrittelyistä. Hyvänä lähtökohtana toimivat esimerkiksi sovellukselle suunnitellut käyttötapaukset (*use cases*), tilakaaviot, skenaariokuvaukset ja muut sovelluksen dokumentit, tai itse käyttötilanteesta tehdyt taltioinnit (käyttölokkit tai videonauhoitteet). Duffyn ja kumppanien mukaan opasteen pitää tarjota vastaukset proseduraalisiin, kuvauksellisiin ja tulkinnallisiin kysymyksiin, sekä myös siihen, miten opastejärjestelmää käytetään [Duffy et al., 1992, s. 80-83]. Postimies on ollut koekäytössä kutakuinkin kaksi vuotta, joten käyttäjäkokemuksia siitä on varmasti kertynyt. Käyttäjätutkimusta ei tämän työn puitteissa kuitenkaan ole mahdollista järjestää, vaan opasteiden suunnittelu pohjautuu Postimies-järjestelmän tutkimiseen.

Jotta tiedetään mitä käyttäjät kysyvät, pitää myös tietää keitä käyttäjät ovat. Jos verrataan opasteen käyttäjää paperille painettua dokumentaatiota tarkastelemaan käyttäjään, edellinen on malttamattomampi etsiessään tietoa ja olettaa avun löytyvän vähällä vaivalla [Duffy et al., 1992, s. 31]. Kohderyhmän valinta on tehtävä sovellusta suunniteltaessa, ja jos kohderyhmiä on useita, voidaan suunnitella kullekin ryhmälle erilaiset opasteet. Asiantuntijoiden mielestä noviisikäyttäjät turvautuvat harvoin opasteisiin, johon on syynä todennäköisesti huonot kokemukset opasteista, jotka "kaappaavat" käyttäjät opastejärjestelmään, josta ei pääse pois kuuntelematta ensin pitkiä ja aikaa vieviä opasteita [Mané et al., 1998]. Usein käyttäjät "tietävät" opasteiden käytön vaativan aikaa ja yrittävät ratkaista ongelman itse [Mané et al., 1998].

Postimiehen käyttäjiä ovat pääasiassa Tampereen yliopiston työntekijät ja opiskelijat, molemmat pääsääntöisesti tietojenkäsittelytieteiden henkilökuntaa, pää- tai sivuaineopiskelijoita. Käyttäjiä on jonkin verran myös yliopiston ulkopuolelta. Sähköpostin käyttö on käyttäjille ennestään tuttua graafisten käyttöliittymien kautta eikä tietokoneiden parissa toimiminen ole täysin outo asia. Puhelimen käyttö perinteiseen puhumistarkoitukseen on yhtä tuttua kuin muullekin väestölle. Puhekkäyttöliittymistä saattaa tietojenkäsittelytieteiden opiskelijoilla olla kokemuksia, sillä puheteknologiaa opetetaan Tampereen yliopistolla kahdella kurssilla. Postimiehen käyttäjä-



dokumentaatiossa todetaan, että Postimiehestä on pyritty tekemään niin käyttäjäystävällinen, ettei käyttöohjeita tarvita, joten käyttäjäkunnalta ei silti odoteta mitään erityisosaamista. Sama pyrkimys koskee täten myös Postimiehen opasteita.

### 3.3.2. Opastejärjestelmän käynnistyminen

Opasteeseen pääsy tulee suunnitella hyvin, koska käyttäjä on usein jo opastetta tarvitessaan pulassa eikä enää kaipaa lisää ongelmia. Opasteeseen johtavan reitin pitää olla aina tunnistettavissa, samassa paikassa ja aina saatavilla muttei haitata sovelluksen muuta käyttöä [Duffy et al., 1992, s.84]. Jos sovelluksessa on modaalisia, opastetta tarvitsevia dialogeja, niihin pitää liittää oma reittinsä opasteeseen. Palattaessa takaisin opasteesta on modaalisuuden oltava edelleen voimassa, ts. käyttäjä ei saa edetä opasteen jälkeen kuittaamatta ensin dialogia. Myös puhekäyttöliittymässä voi olla tilanteita, joissa käyttäjän on joko keskeytettävä koko toimintoketju tai esimerkiksi vastattava johonkin järjestelmän asettamaan kysymykseen. Duffyn ja kumppanien mukaan opasteeseen voi olla myös useita yhtäaikaan voimassa olevia reittejä. Tällaisia voisivat olla esimerkiksi puhekäyttöliittymässä yhtäaikaisesti voimassa olevat puhelimen tietyn, opasteen käynnistävän näppäimen painallus ja sanan ”apua” sanominen.

Puhekäyttöliittymän suunnittelussa on oletettava, ettei käyttäjä tiedä järjestelmän aktiivista sanavarastoa, ja ohjattava käyttäjää antamaan oikeita syötteitä järjestelmälle [Mané et al., 1996]. Opasteen aktivoivan käskyn pitäisi olla se, mikä käyttäjälle ensimmäisenä tulee mieleen ja myös sellainen, että se erottuu selvästi muista järjestelmän komennoista, jotta puheentunnistin kykenisi tunnistamaan sen aina virheettömästi. Teoriassa voitaisiin syöttää järjestelmään opastejärjestelmän käynnistäviksi avainsanoiksi kaikki mahdolliset avuntarpeeseen viittaavat sanat, mutta käytännössä se saattaisi johtaa opastejärjestelmän jatkuvaan turhaan käynnistymiseen. Myös hiljaisuus voidaan tulkita avuntarpeeksi. Silloin voidaan selvästi kysyä käyttäjältä tarvitseeko tämä apua ja ohjata käyttäjä vastaamaan joko kyllä tai ei. Jos opasteiden selaaminen vaatii komentoja, nekin on otettava huomioon. Esimerkiksi jos käyttäjälle tarjotaan lisätietoja joistakin asioista, järjestelmän tulosteessa on selvästi ilmoitettava lisätietovaihtoehdot, sekä millä komennoilla niihin pääsee käsiksi.

Virheentunnistus- ja korjausprosessi voidaan jakaa muun muassa neljään osaan: virheen tunnistamiseen, virhetilanteen diagnosoimiseen, korjauksen suunnitteluun ja sen toteutukseen [Mané et al., 1996]. Opastejärjestelmän käynnistäminen liittyy läheisesti tähän prosessiin, sillä virhetilanteen korjauksen yhteydessä saattaa olla järkevää käynnistää virhetilanteeseen liittyvä opaste. Ajatellaan, että käyttäjä on tilanteessa,

jossa hän on antanut komennon, jota puheentunnistin ei ole tunnistanut. Puheentunnistin ilmoittaa dialoginhallinnalle, että saatiin tunnistamaton syöte (virheen tunnistaminen) ja dialoginhallinnassa tiedetään, mitä operaatiota käyttäjä oli suorittamassa. Korjaukseksi suunnitellaan käyttäjältä uudelleen kysyminen, mutta käyttäjälle tarjotaan myös tietä opasteisiin esimerkiksi ilmoittamalla: "Sano apua, jos et tiedä mitä tehdä." Samanlaista toimintaa voitaisiin käyttää myös käyttäjän oltua riittävän kauan hiljaa.

Puheeseen perustuvassa MailCall-sähköpostiohjelmassa sisään kirjautumisen jälkeen käyttäjälle kerrotaan, että hän voi koska tahansa kysyä "mitä voin sanoa" tai painaa puhelimen näppäintä "0" saadakseen apua [Marx & Schmandt, 1996]. Shriver ja muut [Shriver et al., 2001] esittelevät artikkelissaan muutamia universaaleja puhekäyttöliittymän komentoja, joiden ideana on tarjota käyttäjälle mahdollisuus käyttää tiettyjä fraaseja missä tahansa käyttöliittymän osassa. Yksi näistä komennoista on "Now what?", jolla käynnistetään aina paikallinen opaste. Vastaava toiminto "Mitä nyt?" on jo toteutettu Postimieheen.

Kansainvälinen komitea "Telephone Speech Standards committee" on koonnut ehdotuksensa komentoiksi kaikkiin (englanninkielisiin) puhekäyttöliittymiin [TSSC, 2000]. Yksi näistä on "Help", joka on usein universaaliksi määritelty komento ja jonka avulla tulee saada lisätietoja liittyen sovelluksen tämänhetkiseen tilaan. Universaalilla tarkoitetaan tässä yhteydessä komentoa, joka on voimassa koko sovelluksen ajan, eli missä tahansa tilanteessa käyttäjä voi sanoa "help". Universaalit komennot ovat hyödyllisiä muun muassa siinä mielessä, että kun käyttäjä oppii tietyt universaalit komennot, jotka ovat voimassa aina, käyttäjän kognitiivista kuormitusta vähennetään. Artikkelissa ehdotetaan siis, että nämä universaalit komennot olisivat yhteneväiset myös sovellusten välillä. Tarkoituksena on auttaa käyttäjää selviämään ongelmatilanteista ja siten edistää käytettävyyttä ja käyttäjätyytyväisyyttä. Artikkelin kirjoittanut komitea on tätä kirjoittaessani valmistamassa standardia, jossa määritellään tällaiset universaalit komennot. Muita ehdotettuja komentoja "help":in lisäksi ovat: "end call" tai "good-bye", "repeat", "transfer to operator" tai "operator", "wrong turn" tai "back up" ja "start over". Komentojen käyttöä testattiin ja neljä komentoa oli selvästi yleisesti käytettyjä: "good-bye", "help", "operator", "repeat".

Standardointikomitean kansainväliseksi tarkoitettun ehdotuksen ja muista puhekäyttöliittymistä saatujen esimerkkien puitteissa ehdotan, että "apua" tai "auta" kelpaisi suomenkieliseksi universaaliksi vastineeksi sanalle "help". Postimieheen on jo toteutettu fraasien "Apua!" ja "Mitä nyt?" tunnistus, mutta myös fraasit "Auta!" ja "Mitä voin sanoa?" voidaan lisätä opasteen käynnistävien komentojen listaan.

Postimieheissä myös äänitaajuuspuhelimien näppäin "8" on suunniteltu käynnistämään opasteen. Koska käyttäjä saattaa haluta tutustua opasteisiin myös yleisesti, komennolla "Opasteet" opastejärjestelmä käynnistyy suoraan päätasolle tarjoamatta ensin tilannekohtaista opastetta. Päätasolta käyttäjän on helpompi hahmottaa koko opastejärjestelmän sisältö ja etsiä itse haluamansa toiminnon opaste.

Kolmas opasteen käynnistävä tekijä on käyttäjän hiljaisuus. Jos puheentunnistimelle ei tule riittävän pitkään aikaan mitään syötettä, opastejärjestelmä käynnistyy automaattisesti. Postimiehen opastejärjestelmällä on parametrina reaktioaika. Reaktioajalla tarkoitetaan sitä aikaa, mikä odotetaan ennen kuin opaste annetaan automaattisesti. Reaktioajan täyttymistä laskeva ajankuluun sidottu laskuri nollataan aina, kun joko käyttäjältä saatava syöte loppuu tai järjestelmän antama tuloste loppuu. Laskuri nollataan myös silloin, kun tunnistin ei tunnista saamaansa syötettä, sillä siinä tapauksessa voidaan olettaa, että tunnistin antaa virheilmoituksen käyttäjälle. Reaktioaikoja voidaan säädellä parametrisesti sovelluksen adaptoituessa käyttäjäprofiilin mukaan, mutta periaatteena voidaan pitää esimerkiksi neljää erimittaista reaktioaikaa: lyhyt, keskipitkä, pitkä ja hyvin pitkä. Näitä suhteellisia aikoja sovelletaan esimerkinomaisesti jatkossa käsiteltäessä reaktioaikaa.

Opastejärjestelmän reaktioaika on uusilla käyttäjillä oletusarvoisesti keskipitkä ja vanhoilla käyttäjillä pitkä tauko. Kun käyttäjästä on alettu kerätä käyttäjäprofiiliin tietoja, lisätään sinne myös käyttäjän kokemusta kuvaavat parametrit: käyttökertojen määrä ja opasteaktivoitien määrä. Reaktioaika pitenee käyttäjän kokemuksen myötä. Yksi vaihtoehto on myös lyhentää reaktioaikaa, jos käyttäjälle joudutaan tarjoamaan opastetta usein; tosin on hieman paradoksaalista, että paljon opasteita kuunnellut käyttäjä saa kuunnella niitä vielä useammin, mutta toisaalta jonkin asian käyttäjälle täytyy olla vielä epäselvää, jos opasteet aktivoituvat hänelle usein.

Neljäs opastejärjestelmän käynnistäjä on virhetilanne. Kun käyttäjä antaa komennon, jota järjestelmä ei tunnista, liitetään virheilmoituksen perään (lyhyin) tilannekohtainen opaste. Jos tilannekohtaista opastetta ei ole, annetaan yleinen opaste. Tämän ominaisuuden osalta olisi mielenkiintoista testata, miten käyttäjät reagoivat, jos opaste annetaan joko heti ensimmäisestä tunnistusvirheestä tai vasta kun käyttäjä on yrittänyt ensin uudelleen.

### 3.3.3. Opasteen valitseminen

Opastetekstejä voi olla yleisiä ja tilannekohtaisia. Yleisen opasteen tulisi olla aina käyttäjän saatavilla. Yleinen opaste on samanlainen kautta koko sovelluksen, kaikissa

sovelluksen tiloissa. Lisäksi yleinen opaste on sekundäärinen tilannekohtaiseen opasteeseen verrattuna: jos tilanteeseen löytyy tilannekohtainen opaste, se tarjotaan käyttäjälle ensisijaisesti.

Opasteen valinnan voi tehdä joko järjestelmä tai käyttäjä. Missä tahansa tilanteessa käyttäjä tuntee olonsa epävarmaksi sovelluksen käytön suhteen, pitäisi järjestelmän pystyä tarjoamaan sellainen opaste, joka auttaa käyttäjää eteenpäin. Tilannekohtaisen opasteen valintaa varten järjestelmän on tiedettävä jotain käyttäjän tilanteesta. Jos adaptiivisessa järjestelmässä otetaan huomioon eri käyttäjät, on hyvä tietää myös jotain käyttäjän kokemuksesta ja historiasta sovelluksen käytössä. Käyttäjän voi valita opasteen, jos opasteet on esitetty jonkinlaisen valintamekanismin avulla [Duffy et al., 1992, s.92]. Esimerkiksi graafisista käyttöliittymistä tuttu puumuotoinen luettelo sopii hyvin hierarkkisten opasteiden valintamekanismiksi. Käyttöliittymän pitää olla olemukseltaan yhdenmukainen jokaisessa tilassa, josta voidaan valita opasteaiheita ja opastejärjestelmän tulee noudattaa samoja sääntöjä toimintamallissaan kuin muunkin käyttöliittymän [Duffy et al., 1992, s.92].

Myös puhekäyttöliittymän opastejärjestelmässä voi päätasolla olla mahdollisuus valita mikä tahansa opasteteksteistä. Yksinkertaisin ja selkein tyyli on luetella opasteiden otsikoita, joista käyttäjä voi valita mieleisensä. Tässä on muistettava, että luettelo voi olla pitkä ainoastaan, jos käyttäjän on mahdollista keskeyttää luettelon kuunteleminen tai jokaisen vaihtoehdon jälkeen on lyhyt käyttäjän kuuntelemiseen tarkoitettu tauko. Jos luetteloista on tehtävä lyhyitä, ei usean opasteen esittämiseen jää muuta vaihtoehtoa kuin järjestää ne hierarkkisesti siten, että muutama aihe on yhdellä tasolla. Hansen ja muut [Hansen et al., 1996] katsovat luetteloivan tyylin katsottu sopivan sellaisiin ympäristöihin, joissa halutaan minimoida tunnistusvirheiden määrä; jos halutaan jäljitellä mahdollisimman luonnollista vuoropuhelua, pitäisi käyttäjän antaa itse sanoa, mihin hän haluaa apua. Postimiehessä on mielestäni järkevää käyttää luetteloivaa tyyliä, koska sähköpostien hallinta on jokapäiväisenä tehtävänä sellainen, jossa suuri tunnistusvirheiden määrä haittaa tehtävän suorittamista merkittävästi. Teknisen toteutuksen osalta on muistettava, että jos yhdestä tilannekohtaisesta opasteesta pääsee toiseen tai opastejärjestelmän muihin toimintoihin (kuten hakuun tai sanastoon), on myös suunniteltava varsinaiseen sovellukseen palaaminen hyvin ja yritettävä välttää navigointiongelmia (ks. aliluku 2.3.6).

Puhekäyttöliittymän suunnittelussa tulisi ensisijaisesti ajatella, että virheet ovat tunnistimen eivätkä käyttäjän virheitä [Mané et al., 1996]. Kuten kaikessa muussakin puhekäyttöliittymän toiminnassa, opasteidenkin osalta pitäisi varautua siihen, että käyttäjän antama kommento on ymmärretty väärin. Opastejärjestelmässä tähän voisi

varautua siten, että ensimmäinen järjestelmän antama tuloste olisi vain hyvin lyhyt, mutta sellainen, että käyttäjä ymmärtää tulleen nyt opastejärjestelmään. Siten käyttäjälle annetaan mahdollisuus poistua niin pian kuin mahdollista mikäli opastejärjestelmään siirtyminen oli virheellinen toiminto. Tunnistimen virheistä toipuminen tulee suunnitella hyvin [Mané et al., 1996]. Opastejärjestelmässä tunnistusvirheet eivät onneksi ole yhtä kohtalokkaita kuin esimerkiksi varsinaisen sovelluksen käytössä tehtävät virheelliset tuhoamiset, mutta usein toistuvat virheet harmittavat ja turhauttavat käyttäjää.

Postimiehessä on aluksi tärkeää tarjota käyttäjälle aina oikea opaste. Koska Postimiesjärjestelmä on adaptiivinen ja tilanne- ja käyttäjäkohtaisen tiedon saaminen myös opastejärjestelmään on helppoa yhteisen tietovaraston ansiosta, mahdollisuuksia monien erilaisten opasteiden tarjoamiseen on paljon. Postimies tarvitsee yhden opasteisiin erikoistuneen evaluoijan (kts. kuvaus Postimiehen arkkitehtuurista, aliluku 3.3), joka päättää, mikä opaste milloinkin käyttäjälle annetaan. Käyttäjäprofiili kertoo käyttäjän kokemuksen määrän sovelluksen käytöstä, minkä perusteella evaluoija valitsee opasteen pituuden ja määrittelee järjestelmän reaktioajan. Mikäli käyttäjää joudutaan jossain tilanteessa opastamaan kahdesti tai useammin, saatetaan käyttäjän profiiliin muuttaa kokemustaso yhtä alemmaksi (mistä seuraa pykälän pidemmät opasteet ja lyhyempi reaktioaika).

Käyttäjälle tarjotaan opasteiden valintamekanismiksi yksinkertaisesti luetteloa opasteiden pääaiheista. Yhden pääaiheen valitseminen antaa käyttäjälle uuden luettelon pääaiheen alle sijoitetuista aliaiheista. Hierarkiataso pyritään pitämään kahdessa, sillä muuten voidaan joutua ratkomaan navigointiongelmia eksymisen vuoksi. Aiheet on esitetty samassa yhteydessä kuin varsinaiset opastetekstit (aliluvussa 3.4.5).

### **3.3.4. Opasteiden esitystavan suunnitteleminen**

Graafisessa käyttöliittymässä opasteiden näytöt tulisi suunnitella siten, että käyttäjän on helppo silmäillä ne läpi, joten vierityspalkit eivät ole suotavia ja ikkunoiden tulisi olla "sopivankokoisia" [Duffy et al., 1992, s.104]. Puhekäyttöliittymissä tämä vaatimus kiinnittää huomion opastevalikoiden ja opastetekstien pituuteen: pitkiä elementtejä tulee välttää joka tilanteessa ja tieto pitää jakaa sopivan pieniin osiin. Opasteessa pitää välttää myös useita hierarkiatasoja ja on varottava opasteen kehittymistä omaksi monimutkaiseksi sovellukseksi, koska silloin sen käytettävyys varsinaisen sovelluksen apuna kärsii [Duffy et al., 1992, s.89].

Sen sijaan puhekäyttöliittymässä erittäin olennainen asia, ääni, tulee tässä vaiheessa mukaan suunnitteluun. Opasteille on valittava puheääni, joka voisi olla erilainen kuin muun sovelluksen ääni erottamaan opastejärjestelmä selkeästi muusta järjestelmästä. Silloin käyttäjän ei tarvitsisi pelätä tuhoavansa tärkeää tietoa oikeasti, jos hän sanoisi opastejärjestelmässä ”tuhoa” halutessaan tietää lisää jonkin tiedon tuhoamisesta. Esimerkiksi hypertekstiä lukevassa sovelluksessa on käytetty naisääntä lukemaan hyperlinkillä merkityt tekstit, mikä auttaa käyttäjiä huomaamaan linkit paremmin [Kitamura & Asakawa, 1999]. Mielenkiintoista on, että puhesynteesi on usein meluisassa ympäristössä ymmärrettävämpää kuin ihmisen puhe. Monessa yhteydessä järjestelmän varoitukset halutaan mieluummin kuulla synteettiseltä kuulostavalla äänellä, jotta varoituksiin pystyttäisiin reagoimaan mahdollisimman nopeasti [Cohen & Oviatt, 1994]. Opasteiden esitystapaa suunniteltaessa on otettava huomioon myös prosodia, kuten sanojen ja lauseiden osien painottaminen ja puheen hidastaminen tarvittavissa kohdissa. Opasteiden on oltava selkeitä, jolloin myös lausumiseen tulee kiinnittää huomiota.

Nopeat puhetulosteet saattavat ohjata käyttäjää puhumaan nopeammin, mikä vaikeuttaa puheentunnistusta; hitaat puhetulosteet saattavat turhauttaa käyttäjää ja käyttäjä alkaa puhua puhetulosteen päälle, mikä edelleen vaikeuttaa puheentunnistusta tai estää sen joissakin järjestelmissä [Hansen et al., 1996]. Puhekäyttöliittymän tulosteita voidaan esittää joko suppenevasti tai inkrementaalisesti [Yankelovich, 1996]. Suppenevassa esitystavassa oletetaan, että kokeneelle käyttäjälle riittää tiiviimmin esitetty tieto kuin vasta-alkajalle. Inkrementaalisessa esitystavassa tulostetta laajennetaan, jos käyttäjä ei reagoi ensin esitettyyn tulosteeseen. Tapoja voidaan soveltaa opastetekstien suunnittelussa esimerkiksi siten, että ensimmäisessä opastetekstissä kerrotaan vain lyhyesti käyttäjän tai järjestelmän valitsemasta aiheesta ja jatkossa opastetekstiä lavennetaan mikäli käyttäjä tarvitsee vielä tarkempaa tietoa. Tällä tavalla pystytään tarjoamaan myös eritasoisia opasteita taidoiltaan ja tiedoiltaan eritasoisia käyttäjiä varten. Liian monimutkaista toiminnallisuutta on silti vältettävä kuten tämän aliluvun alussa esitettiin.

Masthoff ja Gupta [Masthoff & Gupta, 2002] jakavat opasteteksteissä esitetyt käyttöliittymän toimintojen kuvaukset osiin, joista kunkin kohdalla päätetään erikseen, näytetäänkö sitä käyttäjälle. Koska opasteissa olevan tiedon tulee olla uutta, kuvauksen osa tarjotaan käyttäjälle vain, jos sitä ei ole näytetty hänelle aiemmin tai jos hän ei ole käyttänyt kuvaukseen liittyvää toimintoa aiemmin. Koska käyttäjä voi unohtaa saamansa tiedon, pitäisi ottaa huomioon myös edellisestä kuvauksen esityskerrasta kulunut aika. Jos käyttäjä on keskeyttänyt opasteen, esittämättä jäänyt osa tarjotaan myös seuraavalla kerralla. Lähteessä esitellyssä multimodaalisessa sovelluksessa

opasteet olivat äänitulosteita. Kyseisen sovelluksen käytettävyydestä ilmeni, että kenellekään koekäyttäjistä ei tarjottu yhtä olennaista opastetta, koska järjestelmä virheellisesti oletti esittäneensä opasteen jo käyttäjille. Tämän perusteella opasteen esittämistä vain kerran kullekin käyttäjälle ei voi välttämättä pitää hyvin soveltuvana sääntönä.

Postimiehen opasteiden esitystapaan kuuluu kaksi ominaisuutta: opasteet tulostetaan eri äänellä kuin järjestelmän muut (opasteiden kanssa samankieliset) tulosteet ja opasteet esitetään käyttäjälle inkrementaalisesti. Postimiehen syntetisaattorille on toteutettu sekä mies- että naisääni, jotka valitaan opasteille siten, että vastakkaisen sukupuolen ääni lukee opastejärjestelmän tulosteet. Eri ääntä käytetään siksi, että opasteen esittämä asia on luonteeltaan erilaista (ohjaavaa) kuin muun järjestelmän tulosteet ja siksi, että käyttäjä huomaa olevansa opastejärjestelmässä. Kolmaskin syy eri äänen käyttöön on: yleisen opasteen teksteissä Postimieheen viitataan kolmannessa persoonassa.

Inkrementaalinen esitystapa tarkoittaa postimiehessä sitä, että ensin (kokeneelle) käyttäjälle tarjotaan lyhyin opaste, sitten keskipitkä ja lopuksi pisin opaste. Ensimmäiseksi annetun opasteen jälkeen odotetaan lyhyen tauon verran ja tarjotaan sitten seuraavaa opastetta. Ennen pisimmän opasteen tarjoamista odotetaan keskipitkän tauon verran. Jos käyttäjä on aloittelija, ei lyhintä opastetta tarjota ollenkaan ja ennen pisimmän opasteen antamista odotetaan pitkän tauon verran. Jättämällä lyhyin opaste pois vältytään aloittelevan käyttäjän turhautumiselta, koska hän todennäköisesti joutuisi usein kuitenkin odottamaan keskipitkää opastetta. Jos viimeisenkin opastetekstin jälkeen seuraa pisimmän tauon mittainen hiljaisuus, käyttäjältä kysytään, haluaako tämä poistua opastejärjestelmästä. Jos tämän kysymyksen jälkeen seuraa edelleen pisimmän tauon mittainen hiljaisuus, opastejärjestelmä suljetaan, koska voidaan olettaa että käyttäjä ei enää ole aktiivinen. Inkrementaalisuudella tavoitellaan käyttäjän tietoihin ja taitoihin adaptoitumista ja miellyttävyyttä.

### **3.3.5. Opasteiden suunnitleminen ja toteuttaminen**

Opastetekstit jakautuvat proseduraaliseen ja kuvaukselliseen tietoon. Proseduraalinen tieto pitää sisällään päämäärätietoisia ohjeita, jotka perustuvat käyttäjakeskeisiin toimintoihin. Tarkennukset, yksityiskohdat proseduureille pitää olla käyttäjän saatavilla, mutta korkeintaan yhden hierarkiatason päässä. Kuvauksellinen tieto kertoo järjestelmän kiinteistä elementeistä ja tulkinnallinen tieto analysoi käsillä olevaa tilannetta ja kuvaa tilanteeseen liittyviä ominaisuuksia. Opasteen esittämisen tiedon modulaarisuutta ja hypertekstiin perustuvaa rakennetta on pidetty käyttäjien

keskuudessa hyvänä asiana. Opasteen tulee olla sekä tarkka että suppea. [Duffy et al., 1992, s.109-112]

Ben Shneiderman korostaa kirjassaan käyttäjakeskeistä ajattelua kiinnittämällä huomiota siihen, kuinka käyttäjälle on valitettavan usein jätetty vastuu omien tavoitteidensa yhdistämisestä sovelluksesta löytyviin usein teknisesti esitettyihin toimintoihin [Shneiderman, 1997]. Jos käyttöliittymä on suunniteltu edellä mainitulla tavalla, opasteille on varmasti käyttöä, kun käyttäjä yrittää selvittää, miten saada tietty tehtävä suoritettua. Opasteet on joka tapauksessa pyrittävä kirjoittamaan siten, ettei enää niistä löydy käyttäjälle käsittämättömiä teknisiä esityksiä.

Opastetekstin suunnittelu on tärkein osa opastejärjestelmää, koska jollei opasteteksti auta käyttäjää mahdollisimman hyvin ja nopeasti käyttäjän ongelmassa, opastejärjestelmää ei käytetä. Optimaalinen tapa tukea oppimista on tarjota käyttäjälle opasteissa uutta ja relevanttia tietoa [Masthoff & Gupta, 2002]. Opasteissa käytettävän kielen tulee olla selkeää. Jos puhetulosteissa käytetään monipuolista kieltä, se voi yllyttää käyttäjää puhumaan puheentunnistimen sanavarastoon kuulumattomilla sanoilla entistä enemmän [Hansen et al., 1996]. Opasteen tulee olla houkutteleva, helposti ymmärrettävä ja jokaisen aiheen opaste on pidettävä lyhyenä [Goodall, 1991]. Jos opaste on pitkä, käyttäjä kuvittelee, että myös aihe on vaikea. Jos käsiteltävä aihe on monimutkainen, on se hyvä käsitellä ensin suurpiirteisesti ja sen jälkeen antaa käyttäjälle mahdollisuus syventyä kuhunkin yksityiskohtaan [Goodall, 1991]. Suurpiirteiset tiedot aiheesta voidaan antaa käyttäjälle esimerkiksi yleisissä ja tarkemmat tiedot tilannekohtaisissa opasteissa.

Puhekäyttöliittymän opasteisiin ei tarvitse suunnitella hyperlinkkejä niiden varsinaisessa ilmiössä, vaan uuteen asiaan johtavat reitit voidaan luetella esimerkiksi kunkin opastetekstin lopussa. Tällä tavalla ohjataan käyttäjää kuuntelemaan aina yksi puhetuloste loppuun, ennen kuin siirrytään seuraavaan asiaan. Keskeyttämisen mahdollisuus voidaan tietysti silti pitää tarvittaessa mukana. Käyttäjää on ohjattava kohti tunnistusmallista löytyvää luonnollisen kielen ilmausta, mutta ei pidä antaa liian monta vaihtoehtoa kerralla [Mané et al., 1996]. Koko järjestelmän kaikkien tulosteiden syntaksin ja semantiikan tulisi olla yhdenmukaista ja tulosteet tulisi pitää lyhyinä, jotta käyttäjäkin puhuisi lyhyesti. Tulosteiden ei pitäisi olla liian samankaltaisia.

Käyttöliittymän suunnittelussa on otettava huomioon, että säännöllinen käyttäjä oppii nopeasti [Mané et al., 1996]. Opasteiden osalta tämä on hyvin olennainen kohta. Opasteita tulisi hyvässä järjestelmässä olla eritasoisia: ensikertalaisia ja kokeneempia käyttäjiä varten. Jos järjestelmä on sellainen, jota tyypillisesti käytetään säännöllisesti,



kuten sähköpostijärjestelmä, voitaisiin opasteiden tarkkuutta keventää ja opasteita lyhentää käyttäjän käyttömäärän kasvaessa. Jos kohderyhmään kuuluu enimmäkseen asiantuntevia käyttäjiä, kuten yliopiston tietojenkäsittelytieteiden opiskelijoita, voi kaikista opasteista suunnitella melko tiiviitä ja lyhyitä.

Filosofi H. P. Grice määritteli 1970-luvulla vuoropuhelulle “Gricen maksimit”, jotka kuvaavat niitä tavoitteita ja oletuksia mitä ihmisillä on keskustelulle [Grice, 1975]. Gricen neljä maksimia ovat:

- määrän maksimi, eli puhumme niin paljon kuin haluamme, emme enempää emmekä vähempää,
- laadun maksimi, eli puhumme totta
- suhteellisuuden maksimi, eli puhumamme asiat liittyvät keskusteluun ja ovat sen osalta relevantteja ja
- tapojen maksimi, eli antamamme tieto on selkeää ja yksiselitteistä.

Useissa 1980-luvun ja 1990-luvun vuoropuhelua käsittelevässä kirjoituksessa kritisoidaan Gricen maksimeja niiden ylimalkaisuudesta ja jopa harhaanjohtavuudesta. Jossain määrin niiden antamat oletukset voidaan kuitenkin pitää mielessä suunniteltaessa puhekäyttöliittymän opasteita, sillä tietokonesovelluksen opastejärjestelmän tulee useimmiten toimia kuten ihanteellinen keskustelukumppani.

Postimiehen yleinen opaste kertoo käyttäjälle ensin ne sanat ja näppäinäänet, jotka ovat voimassa kaikkialla sovelluksen sisällä. Yleisessä opasteessa kerrotaan lisäksi, miten käyttäjä pääsee kuuntelemaan myös muista toiminnoista kertovia opasteita. Jos opasteessa on käytössä sanasto tai hakutoiminto, yleisessä opasteessa on kerrottava, miten niitä pääsee käyttämään. Tässä työssä ei suunnitella Postimiehelle sanastoa eikä hakutoimintoa.

Seuraavassa on esitetty opasteet siten, että ensin on vahvennetulla tekstillä opasteen käynnistävät tekijät, sitten inkrementin numero (jos kyseiselle opasteelle on toteutettu inkrementaalinen rakenne) ja lopuksi varsinainen opasteteksti. Kun opastetekstissä on esitetty sana heittomerkkien välissä, sana on järjestelmän tunnistama komento. Heittomerkkien välissä oleva sana tulee lukea selvästi painottaen, jotta käyttäjä ymmärtää kyseessä olevan avainsanan. Ensimmäisenä on esitetty Postimiehen yleinen opaste, jonka avulla käyttäjä saa kuvan myös toimintoketjuista. Jos käyttäjä sanoo saman avainsanan uudestaan, tarjotaan seuraavan inkrementin opasteteksti tai kaikkien inkrementtien jälkeen yleinen opaste. Uusi opasteteksti tarjotaan aina kunnes käyttäjä itse poistuu opastejärjestelmästä tai lopettaa puhelun. Käyttäjää opastetaan käyttämään primäärejä komentoja (kunkin komentolistan ensimmäinen), mutta Postimies tunnistaa myös joukon synonyymejä komenoille (komentolistan loput komennot). Komenoista

"palaa päätasolle" ja "kartta" kerrotaan tarkemmin navigoinnin yhteydessä seuraavassa luvussa 3.4.6.

### Postimiehen toteutettavat toiminnot

Käyttäjä sanoo: "**apua**", "**auta**", "**mitä nyt**" tai "**mitä voin sanoa**" *eikä käsillä olevaan tilanteeseen löydy omaa opastetta* tai

käyttäjä sanoo: "**opasteet**" tai "**opaste**" tai

järjestelmä: *käyttäjälle on jo tarjottu tilannekohtaista opastetta (seuraava kohta), ja käyttäjä sanoo "**lisää apua**", "**apua**", "**auta**", "**mitä nyt**" tai "**mitä voin sanoa**":*

"Tämä on Postimiehen opaste. Kun haluat poistua opasteesta, sano 'sulje opaste'." (lyhyt kuuntelutauko) "Voit aina sanoa 'apua', kun tarvitset opastusta. Voit aina palata Postimiehen alkuun sanomalla 'palaa päätasolle'. Voit myös aina sulkea puhelimen, kun haluat lopettaa Postimiehen käytön. Apua voit saada aiheista: 'kansiointi' tai 'postien käsittely' tai 'kartta' tai 'ongelmat'. Mistä haluat tietää lisää?"

*Tämä komento käynnistää opastejärjestelmän.*

Käyttäjä on Postimiehen päätasolla ja sanoo "**apua**", "**auta**", "**mitä nyt**", "**mitä voin sanoa**" tai

järjestelmä: *käyttäjä on Postimiehen päätasolla ja reaktioaika täyttyy tai käyttäjä on opastejärjestelmässä kohdassa kansiointi:*

1. inkr: "Voit sanoa 'listaa kansiot' tai 'seuraava kansio' tai 'edellinen kansio'."
2. inkr: "Sähköpostisi on jaettu kansioihin, joita voit valita sanomalla 'listaa kansiot' tai 'seuraava kansio' tai 'edellinen kansio'."
3. inkr: "Olet Postimies-sähköpostiohjelman päätasolla. Sähköpostisi on jaettu kansioihin, joita voit valita sanomalla 'listaa kansiot' tai 'seuraava kansio' tai 'edellinen kansio'. Voit lopettaa Postimiehen käytön sulkemalla puhelimen. Jos tarvitset lisää apua, sano 'lisää apua'."

*Tämä komento ei käynnistä opastejärjestelmää.*

Aina käyttäjän sanoessa "**lisää apua**" käynnistetään yleinen opaste (ja opastejärjestelmä).

Käyttäjä on kansiossa ja sanoo "**apua**", "**auta**", "**mitä nyt**", "**mitä voin sanoa**" tai

Järjestelmä: *käyttäjä on kansiossa ja reaktioaika täyttyy tai käyttäjä on opastejärjestelmässä kohdassa postien käsittely:*

1. inkr: "Voit sanoa 'listaa postit' tai 'lue seuraava' tai 'lue edellinen' tai esimerkiksi 'tuhoa kolmas'."

2. inkr: " Voit siirtyä toiseen kansioon sanomalla 'seuraava kansio' tai 'edellinen kansio'. Voit sanoa myös 'listaa postit' tai 'lue seuraava' tai 'lue edellinen' tai esimerkiksi 'lue kolmas' tai 'tuhoa kolmas'. "
3. inkr: "Voit sanoa 'lue seuraava' tai 'lue edellinen' tai esimerkiksi 'lue kolmas' tai 'tuhoa kolmas'. Voit valita postin sanomalla esimerkiksi 'valitse kolmas' ja sen jälkeen lukea valitun postin sanomalla 'lue'. Voit siirtyä toiseen kansioon sanomalla 'seuraava kansio' tai 'edellinen kansio' tai listata kansiot sanomalla 'listaa kansiot'. Jos tarvitset lisää apua, sano 'lisää apua'. Voit lopettaa Postimiehen käytön sulkemalla puhelimen."

*Tämä komento ei käynnistä opastejärjestelmää.*

Käyttäjä on Postimiehen päätasolla ja sanoo **"kartta"** tai **"missä olen"**:

"Olet Postimiehen päätasolla."

*Tämä komento ei käynnistä opastejärjestelmää.*

Käyttäjä on Postimiehessä jossakin kansiossa ja sanoo **"kartta"** tai **"missä olen"**:

"Olet Postimiehen <n:nnessä> kansiossa."

*Tämä komento ei käynnistä opastejärjestelmää.*

Käyttäjä on Postimiehessä jossakin postissa ja sanoo **"kartta"** tai **"missä olen"**:

"Olet Postimiehen <n:nnessä> kansiossa, <n:nnessä> postissa."

*Tämä komento ei käynnistä opastejärjestelmää.*

### **Opastejärjestelmään toteutettavat toiminnot**

Järjestelmä: *käyttäjä on joko opastejärjestelmän päätasolla tai kohdissa kansiointi, postien käsittely tai ongelmat ja reaktioaika täyttyy:*

1. inkr: "Voit sanoa 'kansiointi' tai 'postien käsittely' tai 'kartta' tai 'ongelmat'.
2. inkr: "Kun haluat poistua opasteesta, sano 'sulje opaste.'" (lyhyt kuuntelutauko) "Voit sanoa 'kansiointi' tai 'postien käsittely' tai 'kartta' tai 'ongelmat'. Mistä haluat tietää lisää?"
3. inkr: "Kun haluat poistua opasteesta, sano 'sulje opaste.'" (lyhyt kuuntelutauko) "Voit aina sanoa 'apua', kun tarvitset opastusta. Voit aina palata Postimiehen alkuun sanomalla 'palaa päätasolle'. Voit myös aina sulkea puhelimen, kun haluat lopettaa Postimiehen käytön. Apua voit saada aiheista: 'kansiointi' tai 'postien käsittely' tai 'kartta' tai 'ongelmat'. Mistä haluat tietää lisää?"

Käyttäjä sanoo: **"kansiointi"** tai **"kansiot"**:

"Postimies kansioi sähköpostisi automaattisesti enintään kuuden postin kansioihin otsikon, lähettäjän tai saapumisajankohdan perusteella. Postimies listaa kansiot, kun sanot 'listaa kansiot'. Voit valita kansion sanomalla 'edellinen kansio' tai 'seuraava kansio'. Kun kansio on valittu, voit sanoa 'listaa postit'. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**listaa kansiot**", "**listaus**" tai "**listaa**":

"Komennolla 'listaa kansiot' Postimies luettelee sinulle kaikki kansiot, joihin se on lajitellut postisi. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**edellinen kansio**" tai "**valitse edellinen kansio**":

"Komennolla 'edellinen kansio' Postimies siirtyy edelliseen kansioon. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**seuraava kansio**" tai "**valitse seuraava kansio**":

"Komennolla 'seuraava kansio' Postimies siirtyy seuraavaan kansioon. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**postien käsittely**", "**käsittely**" tai "**postit**":

"Postimiehen avulla voit lukea tai tuhota sähköposteja. Postit on numeroitu yhdestä kuuteen. Voit valita postin sanomalla 'edellinen posti' tai 'seuraava posti' tai esimerkiksi 'valitse kolmas'. Voit lukea kolmannen postin sanomalla 'lue kolmas' tai jos kolmas posti on valittu, voit sanoa vain 'lue'. Voit tuhota postin sanomalla 'tuhota kolmas'. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**edellinen posti**" tai "**valitse edellinen posti**":

"Komennolla 'edellinen posti' Postimies siirtyy edelliseen postiin siinä kansiossa, jossa olet. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**seuraava posti**" tai "**valitse seuraava posti**":

"Komennolla 'seuraava posti' Postimies siirtyy seuraavaan postiin siinä kansiossa, jossa olet. Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**valitse**", "**valinta**", "**postin valinta**", "**valita**", "**valitse ensimmäinen**", "**valitse toinen**", "**valitse kolmas**", "**valitse neljäs**", "**valitse viides**" tai "**valitse kuudes**":

"Komennolla 'valitse' Postimies valitsee haluamasi postin siinä kansiossa, jossa olet. Jos sanot 'valitse kolmas', Postimies valitsee kansion kolmannen postin. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"edellinen "**:

"Komennolla 'edellinen kansio' Postimies siirtyy edelliseen kansioon. Komennolla 'edellinen posti' Postimies siirtyy edelliseen postiin siinä kansiossa, jossa olet. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"seuraava "**:

"Komennolla 'seuraava kansio' Postimies siirtyy seuraavaan kansioon. Komennolla 'seuraava posti' Postimies siirtyy seuraavaan postiin siinä kansiossa, jossa olet. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"lue", "lukea", "lukeminen", "postin lukeminen", "lue ensimmäinen", "lue toinen", "lue kolmas", "lue neljäs", "lue viides" tai "lue kuudes"**:

"Komennolla 'lue' Postimies lukee haluamasi postin siinä kansiossa, jossa olet. Jos sanot 'lue kolmas', Postimies lukee kansion kolmannen postin. Postimies lukee valitun postin pelkällä komennolla 'lue'. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"tuhoa", "tuhoaminen", "postin tuhoaminen", "tuhota", "tuhoa ensimmäinen", "tuhoa toinen", "tuhoa kolmas", "tuhoa neljäs", "tuhoa viides" tai "tuhoa kuudes"**:

"Komennolla 'tuhoa' Postimies tuhoaa haluamasi postin siinä kansiossa, jossa olet. Jos sanot 'tuhoa kolmas', Postimies tuhoaa kansion kolmannen postin. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"kartta" tai "missä olen"**:

"Komennolla 'kartta' Postimies kertoo missä olet. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"ongelmat" tai "ongelma"**:

"Jos Postimies ei ymmärrä mitä sanot, yritä puhua vähän hitaammin ja selvemmin, mutta älä kuitenkaan liioittele. Jos Postimies ei vielä kukaan ymmärrä, tarkista käytätkö oikeita sanoja, sillä Postimiehen sanavarasto on pieni. Mistä haluat tietää lisää?"

Käyttäjät sanoo: **"palaa päätasolle", "palaa", "palaa alkuun" tai "palaa"**:

"Komennolla 'palaa päätasolle' Postimies palaa alkuun. Kaikki suoritettujen toimien jäävät voimaan. Mistä haluat tietää lisää?"

Järjestelmä: *käyttäjä on opastejärjestelmässä ja pisin reaktioaika täyttyy kolmannen inkrementin jälkeen:*

"Haluatko, että opaste suljetaan?"

- Käyttäjän vastattua "**kyllä**" tai "**joo**" tai "**sulje opaste**" tai (pisin) hiljaisuus:

"Opaste on suljettu."

- Käyttäjän vastattua "**ei**" tai "**en**" tai "**älä sulje**":

"Mistä haluat tietää lisää?"

Käyttäjä sanoo: "**sulje opaste**", "**sulje**", "**poistu**" tai "**lopetä**":

"Opaste on suljettu."

Käyttäjä on opastejärjestelmässä, eikä käyttäjän antamaa komentoa pystytä tunnistamaan:

1. inkr: "Voitko toistaa?"
2. inkr: esitetään kyseessä olevaan opastejärjestelmän tasoon kuuluva opasteteksti, 1. inkrementti.
3. inkr: "Anteeksi, mutta en saa selvää. Jos haluat sulkea opasteen, sano "sulje opaste". Voitko toistaa mitä sanoit?"
4. inkr: esitetään kyseessä olevaan opastejärjestelmän tasoon kuuluva opasteteksti, 2. inkrementti, jos sellainen kuuluu tasoon. Muuten palataan tämän kohdan 1. inkrementtiin.
5. inkr: "En saa vieläkään selvää. Puhu selkeästi, mutta älä liioittele. Jos haluat sulkea opasteen, sano "sulje opaste". Voit lopettaa Postimiehen käytön sulkemalla puhelimen. Voitko sanoa vielä uudestaan?"
6. inkr: esitetään kyseessä olevaan opastejärjestelmän tasoon kuuluva opasteteksti, 3. inkrementti, jos sellainen kuuluu tasoon. Muuten palataan tämän kohdan 1. inkrementtiin.

Loputtomasti toistuvan tunnistusvirheen kanssa vastuu jatkosta jätetään käyttäjälle. Jos puheentunnistus on niin toivotonta, että käyttäjä ei pysty ohjaamaan opastejärjestelmää, ei hän myöskään kykene käyttämään Postimiestä kovin sujuvasti. Todennäköisesti käyttäjä lopulta luovuttaa ja sulkee puhelimen.

Postimiehen opasteet tarvitsevat kolme erilaista esitysagenttia: lyhyt agentti, keskipitkä agentti ja pitkä agentti. Lyhyt agentti huolehtii 1. inkrementin, keskipitkä agentti 2. inkrementin ja pitkä agentti 3. inkrementin tulosteiden esityksestä. Evaluoija valitsee agentteja käyttöön paitsi inkrementin järjestysluvun mukaan (ensin tarjotaan lyhin, sitten keskipitkä ja sitten pisin, kuten aliluvussa 4.2.3 esitettiin) myös käyttäjän kokemuksen mukaan.

Puheenvuoron vaihtumisen tunnistaminen voi olla sekä käyttäjälle että puheentunnistimelle hankalaa. Käyttäjää voidaan rohkaista käyttämään luonnollista puheenvuoron päättävää intonaatiota käyttämällä sellaista myös puhetulosteissa [Hansen et al., 1996]. Postimiehen puhetulosteissa on käytetty tällaista intonaatiota. Puhetulosteet voidaan lopettaa merkkiääneneen, jolloin käyttäjä tietää milloin on hänen vuoronsa [Hansen et al., 1996]. Postimiehessä puheenvuoron vaihtuminen osoitetaan myös merkkiäänellä.

### 3.3.6. Navigointimallin toteuttaminen

Opasteissa voi yleensä kulkea joko päätasolta hierarkiassa syvemmälle yhden aiheen parissa tai vertikaalisti aiheesta toiseen jonkinlaisen valitsimen avulla. Jos käytössä on hakutoiminto tai paljon hyperlinkkejä opasteiden välillä, on aiheesta toiseen liikkuminen käyttäjän mielikuvituksen vallassa eikä käyttäjän liikkumista voida ennustaa. Vapaa navigoiminen helpottaa usein etsityn tiedon löytymistä. Siksi lopullisena tavoitteena on mielestäni suunnitella opastejärjestelmä siten, että se tukee vapaata liikkumista. Koska ajan ja paikan tajun menettäminen hyperavaruudessa on mahdollista, mielestäni myös puhekäyttöliittymien navigointiongelmaan voisi tarjota ratkaisuksi kartan sovelluksesta. Käyttäjän suorittaman tehtävän onnistumisen kannalta ei liene kovin hedelmällistä, jos käyttäjälle ainoa keino selvittää navigointiongelma on palata takaisinpäin. Kartta voidaan esittää esimerkiksi komentoketjuna tai jos mahdollista, ohjelman eri osien nimistä koostuvana ketjuna.

Selvitettäessä millainen navigointimalli opastejärjestelmällä pitäisi olla voidaan teettää koekäyttäjillä skenaarioita, minkä perusteella voidaan sitten tehdä ensimmäinen arvaus siitä miten opasteissa navigoimisen pitäisi edetä [Duffy et al., 1992, s.113-115]. Tässä pitäisi käyttää käytettävyyss- ja käyttäjätestejä apuna mahdollisimman usein, sillä mikä tuntuu suunnittelijasta selvältä voi kokemattomasta käyttäjästä olla hyvinkin hämmentävää. Liian monen navigointimahdollisuuden tarjoaminen voi myös tehdä käyttäjän etenemisen vaikeaksi. Aluksi opaste pitäisi toteuttaa mahdollisimman yksinkertaisella navigointimallilla, ja vasta käyttäjätestien jälkeen tarvittaessa lisätä monimutkaisuutta [Duffy et al., 1992, s.113]. Myöskään hypertekstilinkkejä ei pidä käyttää liiaksi, koska opasteen pitäisi tarjota käyttäjälle mahdollisimman relevanttia tietoa saman tien, eikä siten että käyttäjän on sitä etsittävä itse [Goodall, 1991].

Puhekäyttöliittymän suunnittelussa on otettava huomioon mahdollisuus ettei käyttäjä tiedä vastausta järjestelmän kysymykseen tai ei ymmärrä kysymystä [Mané et al., 1996]. Opastejärjestelmän ollessa kyseessä vastaavanlaisia tilanteita ei pitäisi syntyä, mutta jos näin kuitenkin käy, on vaihtoehtoina joko kysyä käyttäjältä uudelleen eri

tavalla muotoillulla kysymyksellä, palata takaisin opastejärjestelmän päätasolle tai poistua opastejärjestelmästä. Käyttäjälle on tarjottava varmistusdialogeja usein, varsinkin jos erehtymisen mahdollisuus tai erehdyksestä koituva vahinko on suuri. Itse opastejärjestelmän käytöstä ei suoranaisesti pitäisi seurata suurta vahinkoa, sillä siinä itsessään ei tehdä ohjelman kulkuun vaikuttavia valintoja. Sen sijaan opasteen ymmärtäminen väärin voi aiheuttaa epäsuorasti vahinkoa. Siihen ei kuitenkaan opastejärjestelmän varmistusdialogeilla pystytä vaikuttamaan. Pikemminkin, ottaen huomioon että opasteiden selaamista pidetään tyypillisesti aikaa vievänä puuhana, varmistusten käyttö pitäisi minimoida.

Marxin ja Schmandtin [1996] käsittelemässä puheeseen perustuvassa sähköpostin-lukuohjelmassa käytettyjen opasteiden alussa käyttäjälle kerrotaan hänen sijaintinsa sovelluksessa kertomalla miten hän sinne päätyi. Seuraavaksi tarjotaan kyseisessä tilanteessa käytettävissä olevat komennot. Yksi globaali komento on täydellinen keskeytys ja uudelleen aloitus "aloita alusta".

Postimiehen navigointimalli on yksinkertainen. Yleisestä opasteesta löytyy kolme pääaihetta, joiden alta pääsee kuuntelemaan kaikkien tunnistettavien sanojen opastetekstit sanomalla tunnistettavan sanan. Yleiseen opasteeseen pääsee silloin kun tilannekohtaista opastetta ei ole tarjolla tai sanomalla "lisää apua". Tilannekohtaisiin opasteisiin pääsee sanomalla pelkästään "apua" (ja siis myös yleisen opasteen kautta). Jos käytetään komentoa "aloita alusta", pitäisi ratkaista perutaanko esimerkiksi postien tuhoaminen vai palataanko pelkästään päätasolle. Jotta epäselvältä komennolta vältytään, Postimieheessä käytetään pelkästään komentoa "palaa päätasolle", jonka kanssa mitään tehtyjä operaatiota ei tarvitse perua. Komennolla "kartta" ja "missä olen" käyttäjä saa koska tahansa kuvauksen sijainnistaan ja valinnoistaan.

### **3.3.7 Opastejärjestelmästä poistuminen**

Puhekäyttöliittymistä on tehty sellaisia havaintoja, että käyttäjät eivät aina tiedä, koska on heidän vuoronsa puhua. Varsinkin jos puhekäyttöliittymä on suunniteltu järjestelmäjohtoiseksi, on hyvin tärkeää kertoa käyttäjälle, koska opasteesta on poistuttu eli koska käyttäjä saa antaa seuraavan komennon. Myös luonnollista vuoropuhelua käyttävässä järjestelmässä on hyvä tehdä käyttäjälle selväksi, mistä lähtien hänen komentonsa vaikuttavat jo oikeaan järjestelmään, eivätkä ole esimerkiksi vain opastejärjestelmässä tehtyjä lisäkysymyksiä.

Opastejärjestelmästä poistuminen voi tietysti tapahtua automaattisesti, kun opaste on annettu käyttäjälle eikä muuta etenemisvaihtoehtoa ole. Mutta silti käyttäjälle on



kerrottava, että opaste loppui, ole hyvä ja jatka. Loppumerkkinä voi olla myös merkkiäni, jos sopivaa sanallista ilmaisua ei löydy. Jos keskeytys on mahdollista, pitää käyttäjälle antaa opastejärjestelmän käynnistyessä ohje siitä, millä komennolla opastejärjestelmän voi keskeyttää ja miten opasteesta pääsee pois.

Postimiehen opastejärjestelmän sulkeminen tehdään niin, että käyttäjä tietää, kun järjestelmästä poistutaan. Kuten edellä opastetekstien kohdalla esitettiin, poistuminen ilmoitetaan käyttäjälle sanomalla "Opaste on suljettu."

### **3.3.8. Opasteiden soveltaminen käytännössä**

Jo opasteita suunniteltaessa on otettava huomioon opasteen sovellettavuus. Opasteesta saadun informaation pitää olla mahdollisimman helposti sovellettavissa käsillä olevaan asiaan varsinaisessa sovelluksessa [Duffy et al., 1992, s.116]. Käytännössä tämä tarkoittaa sitä, että opasteessa on käytettävä juuri niitä komentoja ja muutenkin samantapaisia ilmaisuja, joita puhekäyttöliittymäkin ymmärtää. Sen lisäksi opasteessa on pyrittävä kertomaan täsmälleen riittävästi käsillä olevasta asiasta.

Graafisille käyttöliittymille on myös asetettu vaatimus, että käyttäjän täytyy voida käyttää sekä opastetta että sovellusta yhtä aikaa [Duffy et al., 1992, s.116]. Koska puhe on luonteeltaan peräkkäistä, yhtäaikaista käyttöä vaikeutuu. Lähimmäksi tätä vaatimusta päästään, jos käyttäjä voisi liikkua opastejärjestelmän ja varsinaisen sovelluksen välillä rajattomasti ja molemmat vielä säilyisivät siinä tilassa, mihin ne kulloinkin jätetään. Tässä tapauksessa kumpaankin järjestelmään pitäisi toteuttaa jonkinlainen muistutin, joka kertoisi kulloisenkin tilanteen, sillä visuaalista palautetta tilanteesta käyttäjä ei saisi. Luulen, että riittävän yhtäaikaista käytön mahdollistavat tilannekohtaiset, opasteet, jotka ovat aina saatavilla ja joista käyttäjä pääsee myös etenemään muihin opasteisiin ja etsimään itse haluamansa tiedon.

Postimiehen opasteita ei tässä työssä sovelleta käytännössä. Toteutus ja evaluointi jätetään mielenkiintoiseksi jatkotutkimusaiheeksi.

## 4 Yhteenveto ja ajatuksia

Tässä työssä koottiin puhekäyttöliittymän opasteiden suunnitteluohjeisto, jonka runko muodostettiin Duffyn ja kumppanien opasteille tarkoitettujen ohjeiden pohjalta. Suunnitteluohjeistoon otettiin mukaan seitsemän kohtaa, joista viimeisin, opastejärjestelmästä poistuminen, oli oma lisäykseni Duffyn luetteloon. Jokaisessa kohdassa huomioitiin tämän työn toisessa luvussa esitetyt puheteknologian asettamat vaatimukset. Suunnitteluohjeiden lomassa esiteltiin myös useita sekä opasteiden että puhekäyttöliittymän tulosteiden suunnittelulle asetettuja vaatimuksia. Lopputuloksena saatiin monitahoiset ohjeet, joita noudattamalla puhekäyttöliittymän opasteiden suunnittelija joutuu ottamaan huomioon monta erilaista seikkaa. Opasteiden rakenteeseen ja sisältöön vaikuttaa kuitenkin paljon sen sovelluksen toteutus, johon opasteet lopulta liitetään.

Opasteiden suunnitteleminen sähköpostiohjelma Postimieheen oli suunnitteluohjeiden kokoamisen jälkeen melko suoraviivainen tehtävä. Koska sähköpostien hallinta on jokapäiväinen ja nopeutta ja vaivattomuutta vaativa tehtävä, Postimiehen opasteiden tärkein vaatimus oli tiivis esitysmuoto ja selkeät komennot. Tilannekohtaisuus ja erilaiset käyttäjätasot ja inkrementaalisuus toivat opasteisiin joustavuutta ja käyttäjäkohtaista adaptiivisuutta. Muutamien muiden opastejärjestelmien ratkaisut, joihin suunnitteluohjeissa viitattiin, olivat hyvin samansuuntaisia tässä työssä tehtyjen ratkaisujen kanssa. Mitä enemmän puhekäyttöliittymältä vaaditaan luonnollisen vuoropuhelun tuntua, sitä enemmän vaaditaan myös opastejärjestelmän suunnittelulta. Postimiehen suunnittelussa ei toistaiseksi ole pyritty kovin mutkikkaaseen luonnollisuuden jäljittelyyn, joten myös opastetekstien muotoilu oli selkeä ja yksinkertainen. Virheisiin varauduttiin suunnitteleamalla opasteteksteistä selkeät ja luettelemalla käyttäjälle vaihtoehdot mahdollisista komennoista. Virheidenkorjaus hoidettiin inkrementaalisilla ohjeilla. Myös muutamat globaalit komennot suunniteltiin helpottamaan käyttöä. Luonnollisen vuoropuhelun vaikutusta suunnitteluun pohdittiin tässä työssä lähinnä teoreettisesta näkökulmasta.

Opasteiden suunnittelu puhekäyttöliittymään ei ole ollenkaan niin yksinkertainen tehtävä kuin tällä hetkellä saatavilla olevien ohjeiden määrästä – tai niiden puutteesta – saattaisi luulla. Pelkästään puhetulosteiden suunnittelemisessa on otettava huomioon useita puheen ominaisuuksiin ja puheteknologiaan liittyviä kohtia, kuten puheentunnistimen virheet, jotka ovat hyvin erityyppisiä graafisen käyttöliittymän virheisiin verrattuna. Opastejärjestelmän on oltava käyttäjälle pelastava oljenkorsi, joten sen toiminnan on oltava varmaa ja mahdollisimman virheetöntä.

Puhekäyttöliittymän opasteissa itse asiassa yhdistyvät kahden erilaisen teknologian tarpeet ja heikkoudet: puhekäyttöliittymät ovat edelleen alttiita virheille ja opastejärjestelmissä virheiden sietokyky saattaa olla jopa heikompi kuin varsinaisessa sovelluksessa. Lisäksi puheteknologian jatkuva kehitys monimutkaistaa hyvien ohjeiden suunnittelua: teknologian asettamat vaatimukset opastejärjestelmälle muuttuvat jatkuvasti hieman.

Seuraavaksi tässä työssä esitetyt suunnitteluohjeet pitäisi evaluoida toteuttamalla ensin Postimiehelle suunnitellut opasteet ja sitten suunnittelemalla ja toteuttamalla opasteille käytettävyydesti. Vaikka kokoamani ohjeet nojaavatkin vahvasti muutamiin alan kirjoihin, tutkimusartikkeleihin ja muihin julkaisuihin, ohjeiden käyttökelpoisuutta voidaan varmuudella arvioida oikein suoritettuna käytettävyydestin analysoinnin jälkeen. Tässä työssä esitetyille tai vastaaville puhekäyttöliittymän opasteiden suunnitteluohjeille on varmasti käyttöä tulevaisuudessa, varsinkin jos puheteknologia lopultakin nousee jo seuraavan sukupolven menestyksekkääksi ilmiöksi.

## Viiteluettelo

- [Arons, 1991] Arons, B. *Hyperspeech: Navigating in Speech-only Hypermedia*. In Proceedings of Hypertext '91, ACM, New York, 1991: 133-146.
- [Bolinger, 1983] Bolinger D. *Intonation and gesture*. In: *American Speech*, 58 (2), 1983: 156-174.
- [Cohen & Oviatt, 1994] Cohen, P., Oviatt, S. *The Role of Voice in Human-Machine Communication*. In *Voice Communication Between Humans and Machines*. Roe, D., Wilpon, J. (editors). National Academy Press, Washington D.C., 1994: 34-75.
- [Cox et al., 2000] Cox, R. V., Kamm, C. A., Rabiner, L. R., Shroeter, J., Wilpon, J.G. *Speech and Language Processing for Next-Millennium Communications Services*. In *Proceeding of IEEE*, vol. 88 (8), August, 2000.
- [Darvishi et al., 1995] Darvishi, A., Munteanu, E., Guggiana, V., Schauer, H., Motavalli, M., Rauterberg, M. *Designing environmental sounds based on the results of interaction*. In *Human-Computer Interaction Interact'95*, Chapman & Hall, UK, 1995.
- [Duffy et al., 1992] Duffy, T., Mehlenacher, B., Palmer, J. *Online Help: Design and Evaluation*. Ablex Publishing Corp., New Jersey, 1992.
- [Goodall, 1992] Goodall, S. D. *Online help: a part of documentation*. In Proceedings of the 10th annual international conference on Systems documentation, ACM, October, 1992: 169 - 174.
- [Goodall, 1991] Goodall, S. D. *Online help in the real world*. In Proceedings of the ACM Ninth International Conference on Systems Documentation, ACM, October, 1991: 21-29.
- [Grice, 1975] Grice, H. P. *Logic and conversation*. In Cole, P., Morgan, J. L. (editors). *Syntax and semantics*, vol. 3: Speech acts, Academic Press, Orlando, FL, 1975: 41-58.

- [Hansen et al., 1996] Hansen, B., Novick, D. G., Sutton, S. *Systematic Design of Spoken Prompts*. In Proceedings of Conference of Human Factors in computing systems, ACM, April, 1996: 157-164.
- [Heeman & Strayer, 2001] Heeman, P. A., Strayer, S. E. *Adaptive Modeling of Dialogue Initiative*. In Proceedings of the NAACL Workshop on Adaption in Dialogue Systems, Pittsburgh, PA, 2001: 79-80.
- [Horvitz, 1999] Horvitz, E. *Principles of Mixed-initiative User Interfaces*. In Proceedings of the CHI'99, ACM SIGCHI conference on Human factors in computing systems, Pittsburgh, PA, 1999: 159-166.
- [Howard, 1987] Howard, J. A. *Flight testing of the AFTI/F-16 Voice Interactive Avionics System*. In Proceedings of the Military Speech Tech, Arlington, Va., Media Dimensions, 1987: 76-82.
- [Java Speech, 1998] *Designing effective speech applications*. In Java Speech API Programmer's Guide, Sun Microsystems, Inc., 1998.  
<http://java.sun.com/products/java-media/speech/forDevelopers/jsapi-guide/>.  
20.11.2002.
- [Juang & Furui, 2000] Juang, B.-H., Furui, S. *Automatic Recognition and Understanding of Spoken Language – A First Step Toward Natural Human-Machine Communication*. In Proceedings of the IEEE, vol. 88 (8), August, 2000: 1142-1149.
- [Kamm, 1994] Kamm, C. *User Interfaces for Voice Applications*. In Roe, D., Wilpon, J. (editors). *Voice Communication Between Humans and Machines*. National Academy Press, Washington D.C., 1994: 422-442.
- [Kamm & Walker, 1997] Kamm, C. A., Walker, M. A. *Design and evaluation of spoken dialog systems*. In Proceedings of the IEEE Workshop on Speech Recognition and Understanding, 1997: 11-18.
- [Kendon, 1980] Kendon, A. *Gesticulation and speech: Two aspects of the process of utterance*. In M.R. Key (editor), *The Relation Between Verbal and Nonverbal Communication*, Mouton, 1980: 207-227.

- [Kitamura & Asakawa, 1999] Kitamura, K., Asakawa, C. *Speech Pointer: A non-visual user interface using speech recognition and synthesis*. In Proceedings of IEEE international conference on Systems, Man, and Cybernetics, vol. 3, October, 1999: 1083-1088.
- [Klatt, 1987] Klatt, D. *Review of Text-to-Speech Conversion for English*. In Journal of the Acoustical Society of America, JASA vol. 82 (3), 1987: 737-793.
- [Klevans & Rodman, 1997] Klevans, R. L., Rodman, R. D. *Voice Recognition*. Artech House Publishers, 1997.
- [Kröger, 1992] Kröger, B. *Minimal rules for articulatory speech synthesis*. In Proceedings of EUSIPCO92, 1992: 331-334.
- [Lai, 2000] Lai, J. *Conversational Interfaces*. In Communications of the ACM, vol. 43 (9), 2000: 24-27.
- [Lai, 2001] Lai, J. *When Computers Speak, Hear and Understand*. In Communications of the ACM, vol. 44 (3), 2001: 66-67.
- [LeCompte, 2000] LeCompte, D. C. *3.14159, 42 and 7±2: Three Numbers that (Should) Have Nothing to Do with User Interface Design*. In ITG Publication, vol. 3 (2), August, 2000.  
[http://www.internettg.org/newsletter/aug00/article\\_miller.html](http://www.internettg.org/newsletter/aug00/article_miller.html). 17.11.2002.
- [Lemmetty, 1999] Lemmetty, S. *Review of speech synthesis technology*. Helsinki University of Technology, 1999.
- [Levow, 1997] Levow, G.-A. *Making Sense of Silence in Speech User Interfaces*. In Proceedings of the CHI '97 Workshop on Speech User Interface Challenges, ACM, 1997.
- [Liu et al., 1998] Liu, D., Nguyen, L., Matsoukas, S., Davenport, J., Kubala, F., Schwartz, R. *Improvements in spontaneous speech recognition*. In Proceedings of the DARPA Broadcast News Transcription and Understanding workshop, February, 1998.
- [Mané et al., 1998] Mané, A., Boyce, S., Karis, D., Yankelovich, N. *Speech User Interface Design Challenges*. In SIGCHI Bulletin, vol 30 (2), ACM, 1998: 30-34.

- [Mané et al., 1996] Mané, A., Boyce, S., Karis, D., Yankelovich, N. *Designing the User Interface for Speech Recognition Applications*. In SIGCHI Bulletin, vol. 28 (4), ACM, 1996: 29-34.
- [Marx & Schmandt, 1996] Marx, M., Schmandt, C. *MailCall: Message Presentation and Navigation in a Nonvisual Environment*. In Proceedings of CHI'96, ACM Press, New York, NY, USA, 1996: 165-172.
- [Masthoff, 1997] Masthoff, J. *An agent-based interactive instruction system*. Eindhoven University of Technology, Eindhoven, The Netherlands, 1997.
- [Masthoff & Gupta, 2002] Masthoff, J., Gupta, A. *Design and Evaluation of Just-in-Time Help in a Multi-Modal User Interface*. Poster in International Conference on Intelligent User Interfaces, San Francisco, January, 2002.
- [Mattingly, 1974] Mattingly, I. G. *Speech synthesis for phonetic and phonological models*. In Current trends in linguistics, Thomas A. Sebeok (editor), vol. 12, Mouton, The Hague, 1974: 2451-2487.
- [Miettinen, 1998] Miettinen, M. (toim.). *Kieliteknologia Suomessa*. Yliopistopaino, Helsinki, 1998.
- [Miller, 1956] Miller, G. A. *The magical number seven, plus or minus two: some limits on our capacity for processing information*. In The Psychological Review, vol. 63, 1956: 81-97.
- [Nielsen, 1994] Nielsen, J. *Heuristic Evaluation*. In Nielsen J., Mack, R. L. (Eds.). Usability Inspection Methods, John Wiley & Sons, New York, NY, 1994.
- [Norman, 1990] Norman, D. A. *The Design of Everyday Things*. New York Currency Doubleday, 1990.
- [Padmanabhan et al., 2001] Padmanabhan, M., Saon, G., Zweig, G., Huang, J., Kingsbury, B., Mangu, L. *Evolution of the Performance of Automatic Speech Recognition Algorithms in Transcribing Conversational Telephone Speech*. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Budapest, Hungary, 2001.

- [Price, 1996] Price, P. *Spoken Language Understanding*. In Survey of the State of the Art in Human Language Technology. 1996.  
<http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html>. 20.11.2002.
- [Rabiner, 1994] Rabiner, L. R. *Voice Communication Between Humans and Machines - An Introduction*. In Voice Communication Between Humans and Machines, National Academy Press, 1994: 5-12.
- [Rabiner, 1989] Rabiner, L. R. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. In Proceedings of the IEEE, vol. 77 (2), February, 1989: 257-285.
- [Roe, 1994] Roe, D. B. *Deployment of Human-Machine Dialogue Systems*. In Voice Communication Between Humans and Machines, National Academy Press, 1994: 373-389.
- [Rohlicek et al., 1993] Rohlicek, J. R., Jeanrenaud, P., Ng, K., Gish, H., Musicus, B., Siu, M. *Phonetic training and language modeling for word spotting*. In Proceedings of IEEE Int. Conf. Acoustics, Speech, and Signal Processing, II, 1993: 459-462.
- [Schafer, 1994] Schafer, R. W. *Scientific Bases of Human-Machine Communication by Voice*. In Voice Communication Between Humans and Machines, National Academy of Press, 1994: 15-33.
- [Schmandt, 1994] Schmandt, C. *Voice Communication with Computers: Conversational Systems*. Van Nostrand Reinhold, New York, 1994.
- [Schroeder, 1999] Schroeder, M. R. *Computer speech: recognition, compression, synthesis*. Springer, Berlin, 1999.
- [Shadle & Damper, 2001] Shadle, C. H., Damper, R. I. *Prospects for Articulatory Synthesis: A Position Paper*. 4<sup>th</sup> ISCA Tutorial and Research Workshop on Speech Synthesis, Scotland, 2001.
- [Shneiderman, 1997] Shneiderman, B. *Designing the User Interface*. Addison-Wesley Publishing Company, 1997.



- [Shneiderman, 2000] Shneiderman, B. *The limits of speech recognition*. In Communication of the ACM, vol. 43 (9), 2000: 63-65.
- [Shriver et al., 2001] Shriver, S., Toth, A., Zhu, X., Rudnicky, A., Rosenfeld, R. A *Unified Design for Human-Machine Voice Interaction*. In Proceedings of ACM CHI 01 Conference on Human Factors in Computing Systems, 2001.
- [Shriver et al., 2000] Shriver, S., Black, A., Rosenfeld, R. *Audio Signals in Speech Interfaces*. In Proceedings of the 6<sup>th</sup> International Conference on Spoken Language Processing, October, 2000.
- [Sweller, 1988] Sweller, J. *Cognitive load during problem solving: Effects on learning*. In Cognitive Science, 12, 1988: 257-285.
- [Tatham & Morton, 1995] Tatham, M, Morton, K. *Speech Synthesis in Dialogue Systems*. In Proceedings of the ESCA Workshop on Spoken Dialogue Systems Theories and Applications, Grenoble, 1995: 221-224.
- [TSSC, 2000] Telephone Speech Standards Committee and Common Dialog Tasks Subcommittee, *Universal Commands for Telephony-Based Spoken Language Systems*. <http://www.acm.org/sigchi/bulletin/2000.2/telephonepaper.pdf>.  
20.11.2002
- [Turk & Corbin Nichols, 1996] Turk, K. L., Corbin Nichols M. *Online Help Systems: Technological Evolution or Revolution?* In Proceedings of the ACM 14<sup>th</sup> International Conference on Systems Documentation, October, 1996: 239-242.
- [Turunen & Hakulinen, 2000a] Turunen, M., Hakulinen, J. *Jaspis - A Framework for Multilingual Adaptive Speech Applications*. In Proceedings of 6th International Conference of Spoken Language Processing (ICSLP 2000), October, 2000.
- [Turunen & Hakulinen, 2000b] Turunen, M., Hakulinen, J. *Mailman – a Multilingual Speech-only E-mail Client based on an Adaptive Speech Application Framework*. In Proceedings of Workshop on Multi-Lingual Speech Communication (MSC 2000), 2000: 7-12.
- [Walker et al., 1998] Walker, M. A., Fromer, J., Di Fabrizio, G., Mestel, C., Hindle, D. *What Can I Say?: Evaluating a Spoken Language Interface to Email*. In

Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems, 1998: 582-589.

[Yankelovich, 1996] Yankelovich, N. *How Do Users Know What to Say?* In ACM Interactions, vol. 3 (6), November/December, Sun Microsystems, 1996.

[Yankelovich et al., 1995] Yankelovich, N., Levow, G.-A., Marx, M. *Designing SpeechActs: Issues in Speech User Interfaces*. In CHI'95 Proceedings, ACM, 1995.

[Zellweger, 1989] Zellweger, P. T. *Scripted documents: a hypermedia path mechanism*. In Proceedings of Hypertext '89, ACM, 1989: 1-14.

[Zue et al., 1996] Zue, V., Cole, R., Ward, W. *Speech Recognition*. In Survey of the State of the Art in Human Language Technology. 1996.  
<http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html>. 20.11.2002.