

Adaptiiviset käyttöliittymät

Ville Heikkinen

Pro gradu –tutkielma



ITÄ-SUOMEN YLIOPISTO

Tietojenkäsittelytieteen laitos

Tietojenkäsittelytiede

Kesäkuu 2015

ITÄ-SUOMEN YLIOPISTO, Luonnontieteiden ja metsätieteiden tiedekunta, Kuopio
Tietojenkäsittelytieteen laitos
Tietojenkäsittelytiede

Heikkinen, Ville Joonas Juhani: Adaptiiviset käyttöliittymät Pro gradu –tutkielma, 48 s.

Pro gradu –tutkielman ohjaajat: FM Jaakko Parviainen ja FT, DI Keijo Haataja
Kesäkuu 2015

Käyttöliittymät ja käyttöliittymäsuunnittelu ovat muuttuneet paljon historiansa aikana ja ovat yhä jatkuvassa muutoksessa. Tähän muutokseen vaikuttavat esimerkiksi tietokoneteknologian kehittyminen sekä tehokkaampiin ja monipuolisempiin käyttöympäristöihin siirtyminen. Koska käyttöliittymä on olennainen osa järjestelmän käytettävyyttä, käyttöliittymäsuunnittelu on yleensä perustunut keskivertokäyttäjän käyttäjämallien noudattamiseen. Nykypäivänä käyttäjäryhmät voivat kuitenkin olla niin monipuolisia, että yhdellä staattisella käyttöliittymällä on mahdoton vastata kaikkien käyttäjäryhmien toiveisiin tai sallia heidän toimia tehokkaasti.

Sen sijaan, että osa käyttäjäryhmistä jätettäisiin huomiotta, voidaan tällaisessa tapauksessa hyödyntää tehokkaasti ns. älykkäitä käyttöliittymiä. Älykkäillä käyttöliittymillä tarkoitetaan käyttöliittymiä, jotka osaavat muokata ulkoasuaan tai toimintojaan niin, että ne sopivat paremmin sen hetkisen käyttäjän toimintamalleihin. Tämä voi tapahtua joko automaattisella adaptaatiolla tai käyttäjän suorittamalla manuaalisella kustomoinnilla.

Adaptiivisuudella pystytään saavuttamaan paremman käytettävyyden lisäksi myös erilaisia taloudellisia etuja myös järjestelmän tuottajan kannalta. Adaptiivisuus ei kuitenkaan ole automaattisesti käytettävyyttä tai taloudellisuutta parantava tekijä, vaan sen on oltava toimiva osa järjestelmän vaatimustenmäärittelyä ja huonosti toimiva adaptiivisuus voi olla jopa järjestelmää heikentävä ominaisuus.

Tämä tutkimus esittelee adaptiivisten käyttöliittymien toimintaperiaatteita sekä niiden hyviä ja huonoja puolia niin käyttäjän kuin järjestelmän kehittäjän kannalta. Tämän lisäksi käydään läpi muutamia esimerkkejä erilaisista adaptiivisista käyttöliittymistä, kuten personalisoiduista avustajista.

Lopussa pohdimme mahdollisia tulevia tutkimusaiheita adaptiivisten käyttöliittymien parista, kuten uusien teknologioiden vaikutusta adaptiivisiin käyttöliittymiin ja erilaisten sosiologisten ilmiöiden vaikutusta adaptiivisuuden hyödyntämiseen.

Avainsanat: adaptiivisuus, personalisointi, tekoäly, älykkäät käyttöliittymät

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Kuopio
School of Computing
Computer Science

Heikkinen, Ville Joonas Juhani: Adaptive User Interfaces

Master's Thesis, 48 p.

Supervisors of the Master's Thesis: M.Sc Jaakko Parviainen and Ph.D, M.Sc. (Tech.)

Keijo Haataja

June 2015

Abstract: User interfaces and user interface design have changed a lot during their history and keep evolving every day. The reasons for this can be found from computer technology that keeps improving and allowing us to work in more efficient and more versatile environments. Because user interface is such an essential part of a software, the design has usually been founded on the average user group and static work environment. These days the user groups and the working environments of a single software can be so divergent that it's impossible to answer all the requirements that the customers have with a single user interface.

Instead of ignoring the needs of some of the user groups, we can utilize artificial intelligence and intelligent user interfaces. With intelligent user interfaces we mean interfaces that can adapt to users behaviour by changing their appearance or functionality so that it suits the needs of the current user or work environment better. This change can happen either automatically by adaptation that's controlled by artificial intelligence or manually by the user.

Along with better usability, adaptive user interfaces can also benefit the manufacturer of the software financially. It's still important to remember that adaptivity is not a feature that will automatically improve the usability of any selected system, but it has to be a well designed part of the requirement specification. Badly designed adaptivity can even be a feature that weakens the overall usability of the system.

This thesis presents some of the basic functionalities of adaptive user interfaces and the possible negative and positive effects for both the user and the software developer. We will also go through some examples of different types of adaptive systems like personalized desktop assistants.

In the end we reflect upon the previous research and figure out a few new possible research topics on adaptive user interfaces.

Keywords: adaptivity, artificial intelligence, intelligent user interface, personalization,

Esipuhe

Tämä tutkielma on tehty Itä-Suomen yliopiston Tietojenkäsittelytieteen laitokselle keväällä 2015.

Ikuisuudelta tuntuneen ajan jälkeen tämä gradu alkaa näyttää valmistumisen merkkejä. Vakituisen työn ja gradun yhteen sovittaminen oli vaikeampaa kuin kuvittelin, mutta työporukka Foster Wheeler Energia Oy:llä oli todella ymmärtäväistä ja kannustivat saattamaan opiskelut loppuun saakka. Kiitokset siis työporukalle sekä varsinkin osaston johtajalle Ari Ojalalle sekä ”mentorille” Seppo Turuselle.

Kiitokset myös ohjaajilleni Jaakko Parviaiselle sekä Keijo Haatajalle, jotka sinnittelivät gradun ohjauksessa tänne loppuun saakka (ilmeisesti hermostumatta minuun), vaikka etenemä oli näinkin hidasta. Ilman Jaakon ja Keijon korjailuja teksti olisi ehkä juuri ja juuri luettavaa ja viittauksista tuskin kukaan ottaisi selkoa. Tiukkaa teki, mutta päästiinhän sinne maaliin lopulta!

Ja lopuksi tottakai kiitokset henkisestä tuesta kotiporukoille, eli isälle, äidille ja siskolle sekä avovaimolle Anskille.

Lyhenneluettelo

HCI	Lyhenne englannin kielisestä termistä human-computer interaction, joka tarkoittaa ihmisen ja tietokoneen välistä vuorovaikutusta ja sen tutkimusta.
IUI	Lyhenne englannin kielen sanoista Intelligent User Interfaces eli älykkäät käyttöliittymät. Voidaan käyttää myös viittauksena älykkäiden käyttöliittymien vuotuisen konferenssiin.
Makro/Skripti	Pieni tietokoneohjelman muoto, jolla voidaan mm. tehdä tietty määrä toimintoja ennalta määrättyssä järjestyksessä kohdejärjestelmässä.
UMoWS	User Model Web Service eli tietokannasta erillään oleva palvelu, joka suorittaa toimintoja tietokannasta löytyvän datan pohjalta eli tässä tapauksessa muodostaa tietokannasta löytyvän käyttäjäprofiilin pohjalta sille soveltuvan käyttöliittymän.
SOAP	Simple Object Access Protocol eli tietoliikenneprotokolla, joka mahdollistaa proseduurien etäkulutsumisen.
Workflow	Työnkulku eli jokaisen ihmisen uniikki työskentelytapa tietyn tehtävän parissa

Sisällysluettelo

1	Johdanto	1
2	Käyttöliittymät ja niiden tekoäly yleisesti	3
2.1	Käyttöliittymien nykytila	4
2.2	Tekoälyn hyödyntäminen käyttöliittymissä	7
2.3	Kustomoitavat käyttöliittymät	10
2.4	Adaptiiviset käyttöliittymät	13
3	Nykypäivän tarpeita ja ongelmia	18
3.1	Teknologiset ja taloudelliset edut	19
3.2	Käyttäjäprofiilit ja tietoturva	23
3.3	Sosiologiset ongelmat	25
4	Tunnettuja älykkäitä käyttöliittymäsovelluksia	27
4.1	Lumiere Project – Clippy	27
4.2	LookOut ja Microsoft-agentit	29
4.3	Siri, Cortana ja Google Now	30
4.4	Adaptiiviset verkkokaupat ja muut verkkopalvelut	32
5	Pohdinta ja jatkotutkimusaiheet	34
5.1	Adaptiivisuuden kannattavuus kaupallisesti ja tuotannollisesti	35
5.2	Universaali käyttöliittymätekoäly	37
5.3	Adaptiivisuuden hyödyntäminen	40
5.4	Adaptiivisuuden ymmärtäminen	42
5.5	Uudet teknologiat ja adaptiivisuus	43
	Viitteet	46

1 Johdanto

Ohjelmistot ja tietokonejärjestelmät ovat osa jokaisen ihmisen arkipäivää. Ne eivät ole enää vain välineitä työntekoon tai alasta kiinnostuneille harrastajille vaan osa vapaa-aikaa ja niitä käyttävät kaikki nuorista vanhuksiin. Ohjelmistot ovat myös siirtyneet pois työpöydiltä ja raskaista toimistokoneista matkapuhelimiin, autoihin, tabletteihin tai jopa jääkaappeihin. Tämän seurauksena ohjelmistojen käyttäjäryhmien laajuus ja monimuotoisuus on räjähtänyt kasvuun viime aikoina. Kun yhden järjestelmän käyttäjäryhmiä on lukemattomia erilaisia ja saman järjestelmän eri käyttöympäristöt voivat vaihdella todella laajalla skaalalla pöytätietokoneista kannettaviin tabletteihin on todennäköistä, että yksittäinen ja muuttumaton käyttöliittymä ei pysty vastaamaan järjestelmälle asetettuihin käytettävyyksivaatimuksiin. Ratkaisuna tällaiseen ongelmaan ovat adaptiiviset käyttöliittymät.

Tämä tutkielma tutkii käyttöliittymiä ja niiden adaptiivisuutta eri muodoissa. Luvussa 2 käydään läpi käyttöliittymien nykytilaa ja sitä, miksi adaptiiviset käyttöliittymät ovat tarpeellisia hyvän käytettävyyden kannalta. Luvussa käydään myös läpi erilaisia adaptiivisuuden tyyppejä sekä adaptiivisuuden taustalla piilevän tekoälyn eri muotoja.

Luvussa 3 perehdytään adaptiivisuuden hyötyihin ja mahdollisiin haittoihin niin käyttäjän kuin järjestelmän toteuttavan organisaationkin kannalta. Tässä luvussa listataan mm. käyttäjien välisiä eroavaisuuksia, joihin adaptiivisuudella pyritään vastaamaan ja kuinka tämä adaptiivisuus hyödyntää myös järjestelmän kehittäjää. Tämän lisäksi pohditaan adaptiivisuuden mahdollisia vaikutuksia tietoturvaan käyttäjän sekä käyttäjäorganisaation kannalta.

4. luvussa käydään läpi muutamia tunnettuja ja vähemmän tunnettuja adaptiivisia käyttöliittymäsovelluksia pyrkien tuomaan esille muutamia reaali maailman esimerkkejä adaptiivisuuden eri muodoista sekä käydään läpi mikä tekee niistä onnistuneita tai epäonnistuneita toteutuksia käyttäjän kannalta.

Lopuksi luku 5 sisältää yhteenvedon ja pohdintaa mahdollisista tulevista jatkotutkimusideoista adaptiivisten käyttöliittymien parissa.

2 Käyttöliittymät ja niiden tekoäly yleisesti

Käyttöliittymä on rajapinta käyttäjän ja ohjelman välillä, jonka avulla käyttäjä ja ohjelma voivat olla vuorovaikutuksessa keskenään. Se on usein yksi tärkeimmistä osista ohjelmistoja, koska käyttöliittymän onnistunut suunnittelu määrittelee usein kuinka helposti käyttäjä pystyy saavuttamaan tavoitteensa (joiden takia hän ylipäättään on käyttämässä ko. ohjelmaa). Tehokas tai muuten hyvin toimiva ohjelma huonolla käyttöliittymällä voi olla käyttäjän kannalta täysin arvoton.

”Ohjelman tulisi olla käyttöliittymältään helposti ymmärrettävä. Ymmärrettävyys johtaa nopeaan ja helppoon käytettävyyteen, jonka johdosta käyttäjän on mahdollista keskittyä paremmin siihen, mitä on tekemässä parantaen työn tehokkuutta.” (Niemi V. 2013)

Valitettavasti hyvä käyttöliittymä ei ole universaali eikä missään nimessä yksiselitteinen käsite vaan toimiva käyttöliittymä ja sen hyvä käytettävyys on useiden tekijöiden summa. Siihen vaikuttavia tekijöitä ovat mm. käyttöympäristö sekä käyttäjät itse. Kun puhutaan käyttöliittymäsuunnittelusta, useat oppaat keskittyvätkin lähinnä käyttäjäryhmien sekä erilaisten käyttötapauksen tunnistamiseen ja keskittämään suunnittelu ns. keskivertoryhmää ja heidän työnkulkuansa (workflow) mukailevaksi.

Kun käyttöliittymäsuunnittelu perustuu ”keskivertokäyttäjään”, voi seurauksena olla monenlaisia ongelmia. Näitä on esitelty mm. tutkimuksessa (Benyon D. 1993), jossa mitattiin käyttäjien keskimääräistä hakuaikaa ja virhemäärää tietokannasta tietoa haettaessa. Tutkimuksesta käy selväksi, että siinä missä toinen käyttäjäryhmä selviää tehtävästä paljon paremmin graafisella valikkokäyttöliittymällä (pienempi virhemäärä), toinen ryhmä taas pystyi toimimaan huomattavasti tehokkaammin tekstipohjaisella käyttöliittymällä saavuttaen nopeamman hakuajan sekä pienemmän virhemäärän.

Tällaisessa tapauksessa suunnittelijat törmäävät ongelmaan, jossa yksittäisellä käyttöliittymällä ei pystytä vastaamaan käytettävyyksivaatimukseen täydellisesti.

Pelkkä graafinen käyttöliittymä rajoittaisi toisen ryhmän tehokkuutta ja voi johtaa muunlaisiin käytettävyysoongelmiin kun taas pelkkä tekstipohjainen käyttöliittymä voi osoittautua liian monimutkaiseksi osalle käyttäjistä (Benyon D. 1993).

Sen sijaan, että toinen käyttäjäryhmistä jätettäisiin huomiotta tai jouduttaisiin käyttämään aikaa ja resursseja loppukäyttäjien kouluttamiseen tai koko projektin käytettävyyksvaatimuksien uudelleenlaatimiseen, voidaan tällaisessa tapauksessa hyödyntää tehokkaasti ns. älykkäitä käyttöliittymiä (*intelligent user interfaces, IUI*). Niiden avulla käyttäjä voi muokata käyttöliittymän vastaamaan paremmin hänen työkulkuaan ja parantamaan työn tehokkuutta tai parhaassa tapauksessa ohjelma voi itse mukautua käyttäjälle paremmin sopivaksi tulkitsemalla käyttäjän ja ohjelman välistä interaktiota. Näiden tulkintojen avulla ohjelman sisäinen tekoäly voi joko muokata itse näkyvää käyttöliittymäänsä tai tarjota käyttäjälle vinkkejä tehokkaampaa toimimista varten. (Benyon D. 1993)

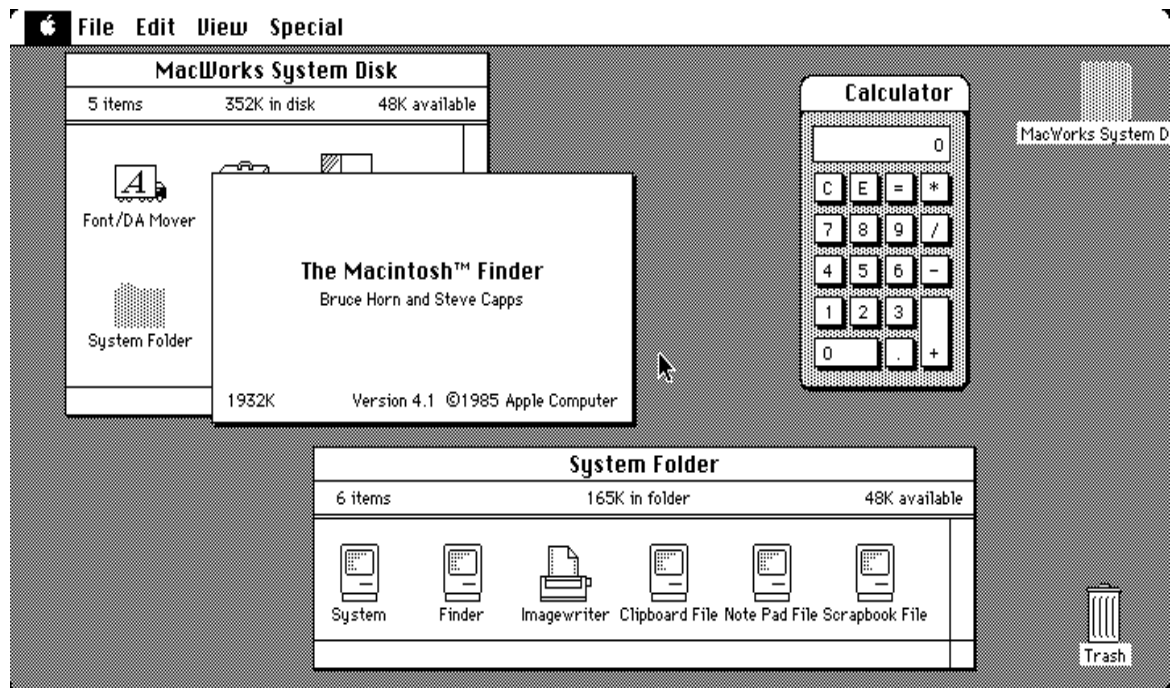
Luvussa 2.1 käydään läpi käyttöliittymien nykytilaa ja sitä, kuinka käyttöliittymät ja niiden suunnittelu ovat muuttuneet aikojen saatossa. Luvussa 2.2 käydään läpi tekoälyn hyödyntämistä käyttöliittymissä ja erilaisia haasteita, joihin sen avulla voidaan vastata käyttöliittymäsuunnittelun parissa. Tämän jälkeen käymme läpi käyttöliittymien kustomoitavuutta ja adaptiivisuutta sekä niiden ominaisuuksia luvuissa 2.3 ja 2.4.

2.1 Käyttöliittymien nykytila

Käyttöliittymät ja kaikki niihin liittyvä on muuttunut paljon aikojen saatossa ja on yhä jatkuvassa muutoksessa. Tähän ovat vaikuttaneet tietysti tietokoneteknologian kehittyminen ja tehokkaampiin käyttöympäristöihin siirtyminen, mutta myös käyttöympäristöjen sekä käyttäjien muuttuminen.

Vielä muutama vuosikymmen sitten, käyttöliittymäsuunnittelijoiden työkalupaletti oli melko pieni. Käytössä oli vain yksinkertaisia tekstikenttiä, nappeja ja pieniä ikoneja (kuva 1). Myöskään nykyisin jo ehkä itsestäänselvyyksinä pidetyt ohjekirjat ja parhaat käytännöt käyttöliittymäsuunnittelusta hakivat vasta omaa muotoaan, joten

suunnittelua tehtiin Windows Style Guiden tai Macintosh Human Interface Guidelinesin mukaan. Nykyään käytettävissä olevia työkaluja sekä oppaita on lukemattomia ja käyttöliittymäsuunnittelun psykologisella puolella on tehty paljon edistyksellistä tutkimusta, jonka avulla suunnittelijat pyrkivät ymmärtämään käyttäjiä ja heidän tarkoituksensa paremmin. (Tidwel 2005)



Kuva 1 - Macintosh Finder (Mac85)

Työkalujen ja ohjeistusten puute eivät missään nimessä olleet ainoat rajoittavat tekijät, joiden alla käyttöliittymäsuunnittelijat joutuivat työskentelemään. Jenifer Tidwel listaa kirjassaan mm. seuraavanlaisia haasteita (Tidwel 2005): värien vähäisyys, pienet resoluutiot, hitaat prosessorit, hitaat tai olemattomat verkkoyhteydet ja käyttäjien kokemattomuus.

Vaikka Jenifer Tidwel (Tidwel 2005) listaakin nämä menneiden aikakausien ongelmiksi käyttöliittymäsuunnittelun parissa, on syytä muistaa, että vaikka jotkut ongelmat ovat harventuneet, se ei tarkoita ettei niitä olisi syytä huomioida suunnittelussa toimivaa käyttöliittymää. Voitaisiin esimerkiksi huomauttaa, että nykyinen mobiililaitteiden aikakausi, mobiiliverkkojen hitaus ja laitteiden pienet

näytöt asettavat suunnittelijoille edelleen samat haasteet kun suunnitellaan käyttöliittymää usealle eri alustalle. Käytännössä nykyajan päätelaitteiden monimuotoisuus (tabletit, puhelimet, kannettavat- ja pöytätietokoneet) onkin tehnyt toimivan käyttöliittymän suunnittelusta entistä vaikeampaa.

Teknologisen kehityksen tuomien muutoksien lisäksi myös käyttäjäkunta on muuttunut. Tietotekniikka ja graafiset käyttöliittymät ovat yhä enemmän osa jokaisen ihmisen päivittäistä arkea. Ohjelmistojen ja varsinkin verkkosivujen käyttäjäkunta laajenee mm. iän, koulutuksen, kulttuurin, kielen sekä jopa erilaisten fyysisten rajoitteiden suhteen jatkuvasti. Ja kuten Tidwel (Tidwel 2005) toteaa, myös käyttäjien odotukset käyttöliittymien ja käytettävyyden suhteen ovat korkeammalla kuin ennen. Kun erilaisia ohjelmistoja ja sovelluksia on kymmeniä tai jopa satoja samaa käyttötarkoitusta varten (esim. tekstinkäsittely, nettiseläus tai musiikin kuuntelu), on käyttäjällä varaa olla nirsompi valitsemansa ohjelman suhteen.

Edellä mainittujen muuttujien seurauksena ohjelmistot ja niiden käyttöliittymät ovat vuosi vuodelta kehittyneet yhä monimutkaisemmiksi. Esimerkiksi kuvassa 2 on kaksi Microsoft Word ohjelman yläpalkkia vuosilta 2000 ja 2010. Vuoden 2010 version yläpalkki sisältää saman verran pikakuvakkeita kuin aikaisempi versio, mutta tämän lisäksi myös 6 muuta välilehteä pikakuvakkeille eli 10 vuodessa tarpeelliseksi arvioitujen pikakuvakkeiden määrä on seitsenkertaistunut. Vaikka valtaosa käyttäjistä käyttää ohjelmissa kulloinkin vain pientä osaa komennoista tai työkaluista, suuri osa ohjelmistoista pyrkii tarjoamaan kaikki työkalut näkyville jatkuvasti käytön aikana. Näin ollen yhä suurempi osa näyttötilasta joudutaan omistamaan käyttöliittymäkomponenteille kuten valikoille ja erilaisille työkalupalkeille. Vaihteleva käyttäjäkunta yhdistettynä siihen, että käyttöliittymän suunnittelusta on usein vastuussa ohjelmoija eikä osaava käyttöliittymäsuunnittelija johtaa tilanteeseen, jossa lopullinen käyttöliittymä ei tue käyttäjän loogista työnkulkua tai pahimmassa tapauksessa edes ohjelman käyttötarkoitusta. (Stuerzlinger & al. 2006).

Word 2000 yläpalkin näkymä



Word 2010 - yksi seitsemästä yläpalkin välilehdistä



Kuva 2 - Word 2000 vrt. Word 2010 (Mic15)

Ohjelmien monimutkaisuus sekä käyttäjien erilaiset tarpeet tekevätkin jokaiselle käyttäjälle optimaalisen käyttöliittymän toteuttamisen todella vaikeaksi, ellei jopa mahdottomaksi (Stuerzlinger & al. 2006). Ratkaisua tähän ongelmaan on viime aikoina haettu tekoälyn sekä kustomoitavien käyttöliittymien avulla.

2.2 Tekoälyn hyödyntäminen käyttöliittymissä

Mikä tahansa tietokoneella suoritettava työskentely voidaan jakaa neljään tekijään : käyttäjä, tehtävä, järjestelmä ja ympäristö (Shackel 1990). Järjestelmän suunnittelijan tavoite on saavuttaa harmonia näiden neljän tekijän välillä, joista jokainen on muunneltavissa tasapainon saavuttamiseksi esimerkiksi käyttäjiä kouluttamalla tai parantamalla käyttöympäristön ergonomiaa (Benyon D. 1993). Mikäli tällaista tasapainoa ei pystytä saavuttamaan eri tekijöiden välille, pystytään tekoälyä hyödyntämään ns. kolmantena osapuolena, jonka avulla pystytään tasapainottamaan kahden eri tekijän välistä interaktiota esimerkiksi muokkaamalla järjestelmää paremmin käyttöympäristöön sopivaksi tai opastamalla käyttäjää tehtävän suorittamisessa.

Vielä 1990-luvun puolessa välissä tekoäly oli melko harvinainen apuväline käyttöliittymän toteuttamisessa, ainakaan näkyvissä määrin. Pikkuhiljaa tekoälyn eri muodot alkoivat kiinnittää käyttöliittymäsuunnittelijoiden huomion käyttöliittymien kasvaessa päivä päivältä monimutkaisemmiksi. Esimerkiksi Microsoft haki apua käyttäjien opastamiseen ja erilaisten käyttötilanteiden havainnointiin todennäköisyyksille kuten Bayesin-verkosta. (Horvitz E. & al. 1998)

Tekoäly tai sen tuomat ominaisuudet itsessään eivät ole automaattisesti ohjelman käytettävyyttä parantava tekijä. Siinä missä kaikki muut käyttöliittymäelementit, myös tekoälyn pitää olla suunniteltu toimivaksi osaksi käyttöliittymää. Tämä periaate valitettavasti itsessään luo pienen ristiriidan kun yritetään arvioida tekoälyn toimivuutta verrattuna tekoälyttömään käyttöliittymään. Ne eivät ole suoraan verrannollisia, koska jos ohjelma on suunniteltu toimimaan tekoälyn avustamana, tekoälystä riisuttu versio on ns. epätäydellinen. Sama ongelma pätee luonnollisesti myös lisättäessä tekoälyä ohjelmaan joka ei sitä tarvitse. (Höök K. 1998)

Kuten Henry Lieberman muistuttaa tutkimuksessaan (Lieberman H. 2009) tekoäly käyttöliittymän toimintaperiaatteena on myös hieman ristiriidassa Human-Computer Interaction koulukunnan ajatuksen kanssa, jonka päämääränä on tehdä tietokoneista helpompia käyttää. Tekoälyn (AI, Artificial Intelligence) päämäärä taas on mallintaa ihmisen ajattelua ja ilmentää sitä tietokoneissa. Usein nämä kaksi päämäärää voivat vaikuttaa vastakkaisilta ja se synnyttää huolen siitä, että tekoälyn heuristiset algoritmit voivat johtaa epäluotettavuuteen käyttöliittymässä ja viedä pois päätöksentekovoimaa pois käyttäjältä. Esimerkkeinä tällaisista tapauksista, joissa tekoälyn ja helppokäyttöisyyden päämäärät eivät kohtaa Lieberman mainitsee mm. Microsoftin animoidun ”Clippy” Office-apurin sekä Newton käsialantunnistusjärjestelmän. Toisaalta kuten David Benyon huomauttaa tutkimuksessaan (Benyon D. 1993), myös ihminen on omanlaisensa adaptiivinen järjestelmä, joka sopeutuu tarpeen tullen ympäristöönsä ja muihin järjestelmiin. Vaikkei ihmismäinen käytös siis aina ole täydellistä adaptiivisuutta ja mahdollisesti silloin tällöin ongelmia aiheuttavaa niin adaptiivisuudella ja ihmismäisellä käytöksellä on siis myös positiivisia yhdistäviä tekijöitä.

Päinvastoin kuin monimutkaisemmat järjestelmät kuten Clippy, valtaosa kaupallisesti menestyneimmistä älykkäistä käyttöliittymistä ovat pieniä ja yksinkertaisia sovelluksia, jotka eivät pyri luomaan olettamuksia käyttäjistä vaan perustuvat tarkkaan tietoon käyttäjästä tai käyttävät muista käyttäjistä kerättyä dataa pohjana adaptoitumiseen (Höök K. 1998). Esimerkiksi Netflixin (ei paras mahdollinen esimerkki kun otetaan huomioon heidän korkeatasoiset algoritminsa,

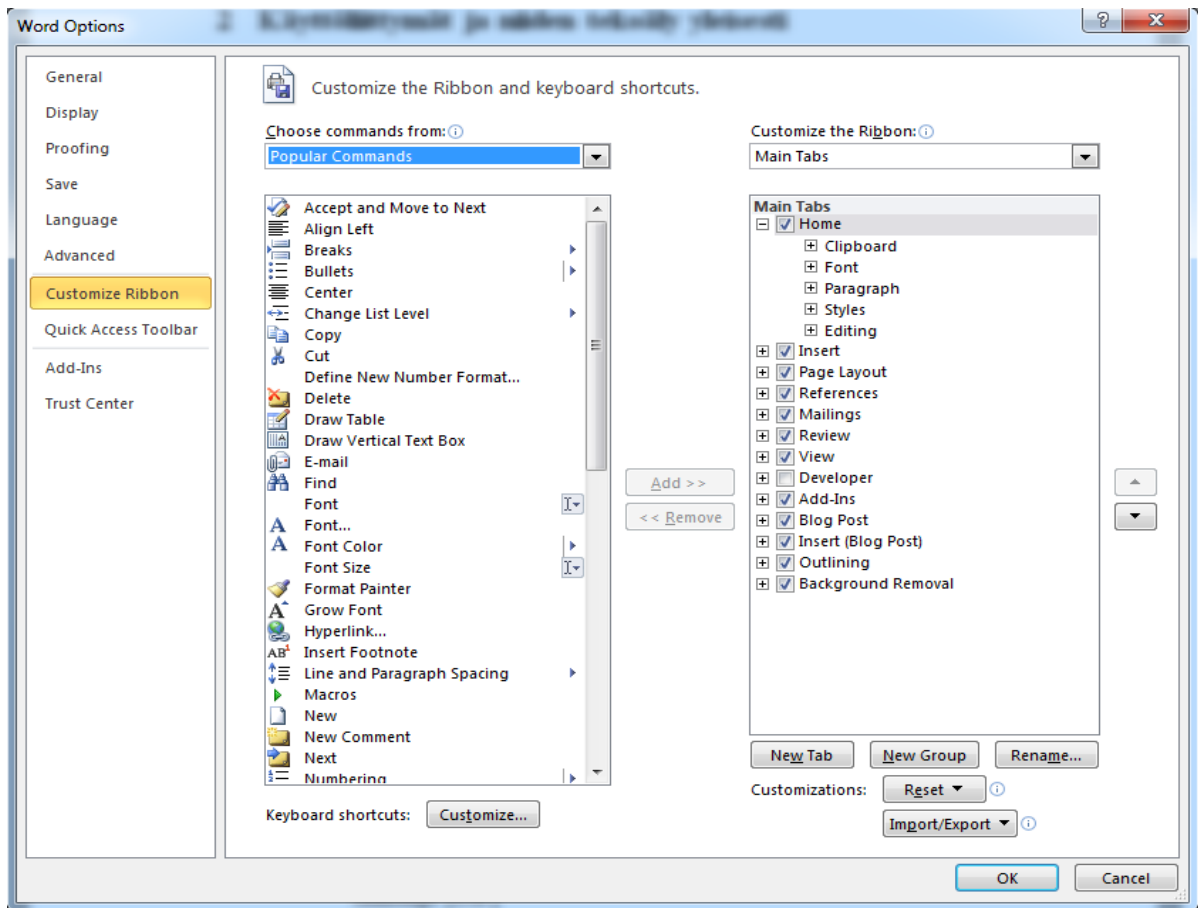
joiden avulla mitataan Netflixin elokuvien henkilökohtainen soveltuvuus katsojalle) tai vastaavien videopalveluiden suosittelulista, jossa personalisoitu etusivunäkymä muodostuu suurimmaksi osaksi vastaavien käyttäjien jo hyväksi havaituista muutoksista.

Víctor Álvarez-Cortés on tutkimuksessaan listannut neljä graafisten käyttöliittymien nykypäivän ongelmaa ja kuinka tekoäly voi auttaa ratkaisemaan niitä (Alvarez-Cortés V. 2007):

- **Personalisoitujen järjestelmien luominen:** Käyttäjillä on erilaiset tavat, mieltymykset ja työnkulut. Älykäs käyttöliittymä, joka osaa huomioida nämä eroavaisuudet pystyy luomaan personalisoidun tavan käydä interaktiota käyttäjän kanssa.
- **Yksinkertaisten tehtävien hoitaminen käyttäjän puolesta:** Älykäs käyttöliittymä voi tarkkailla käyttäjän tekemää työtä, tunnistaa sen päämäärä ja hoitaa osan työhön liittyvistä tehtävistä sallien käyttäjän keskittyä muihin tehtäviin.
- **Informaation määrän vähentäminen tai ongelmien suodatus ruudulla:** Koska nykyajan käyttöliittymät ovat niin täynnä graafisia elementtejä, käyttäjillä on vaikeuksia löytää tarvitsemiansa työkaluja. Älykäs käyttöliittymä voi parantaa käytettävyyttä poistamalla tarpeettomia työkaluja tai tarvittaessa tuoda esille uusia työkaluja.
- **Avun tarjoaminen uusien ja monimutkaisten järjestelmien käyttäjälle:** Tekoäly voi auttaa käyttäjiä toimimaan tehokkaammin kun kyseessä on uusi ohjelma tai esimerkiksi kun ohjelma päivittyy ja tarjoaa uusia toiminnallisuuksia, jotka ovat käyttäjälle vieraita. Älykäs käyttöliittymä voi tarjota käyttäjälle informaatiota näissä tapauksissa tai jopa korjata väärinkäsityksiä.

2.3 Kustomoitavat käyttöliittymät

Kustomoitavuus on yksi yksinkertaisimmista tavoista hyödyntää tekoälyä käyttöliittymässä. Käytännössä se tarkoittaa sitä, että käyttäjälle annetaan mahdollisuus muokata käyttöliittymän ulkoasua rajoitetussa määrin. Yleisimpiä kustomoinnin kohteita ovat erilaiset visuaaliset pikakuvakkeet, jotka sijaitsevat ohjelman näkyvimmillä paikalla ja tarjoavat käyttäjälle nopean pääsyn hänen työskentelynsä kannalta tärkeimpiin työkaluihin. Tällaisia kohteita ovat mm. työpöydän pikakuvakkeet tai ohjelmien työkalupalkit. Esimerkiksi Microsoft Word on jo pitkään tarjonnut käyttäjilleen mahdollisuuden kustomoida ohjelman yläosassa näkyviä työkaluja (kuva 3). Harvinaisempia ja enemmän käyttäjän asiantuntemusta vaativia kustomointikohteita ovat erilaiset makrot eli ennalta ohjelmoidut sarjat komentoja, joilla ohjataan pääsovellusta kuten Adobe Photoshopin kuvankäsittelymakrot ja ohjelmitava toiminnallisuus komentosarja- eli skriptikielellä kuten Visual Basic-kielellä Microsoft Excel-ohjelmassa. Muita käyttäjien suosiossa olevia kustomoinnin muotoja ovat ohjelmien ”skinit” ja ”teemat”, joilla käyttäjä voi muuttaa ohjelman ulkonäköä erilaiseksi.



Kuva 3 - Microsoft Excel kustomoitavuus (Mic15)

Vaikka kustomointi voi vaikuttaa kaiken kattavalta ratkaisulta käytettävyyden ongelmiin, sen toimiva implementointi osaksi käyttöliittymää ei ole aina täysin ongelmatonta. Yksi suurimpia ongelmia on se, että tämän toiminnallisuuden lisääminen ohjelmistoon vaatii huomattavan määrän lisätyötä kehittäjiltä. Valtaosa käyttöliittymätyökaluista ei suoraan tue kustomoitavuutta vaan se joudutaan ohjelmoimaan omana osanaan käyttöliittymän toiminnallisuutta. Muutamia poikkeuksiakin toki löytyy, kuten Applen Cocoa-ohjelmointirajapinta, mutta yleensä näissä poikkeustapauksissakin kustomointi-työkalut ovat kaukana täydellisestä ja sisältävät puutteita kuten mahdollisuuden palata taaksepäin peruen muutokset käyttöliittymään.

On myös syytä muistaa, että kustomoitavuuden toimivuus on paljolti kiinni käyttäjästä, jonka vastuulla käyttöliittymän muokkaus on. Valtaosa

normaalikäyttäjistä ei osaa käyttää kustomoitavuutta hyödyksi tehokkaasti. Makrot ja säännölliset lausekkeet (regular expressions) ovat vieraita monelle ja toisaalta käyttäjät, jotka osaisivat hyödyntää näitä mahdollisuuksia paremmin eivät jaksa tai pysty käyttämään työaikaan käyttöliittymän muokkaukseen, vaikka siitä olisi hyötyä tulevaisuudessa. Kustomoitavuuden suuri haaste onkin onnistua luomaan toiminnallisuudesta sellainen, että se on helposti ja nopeasti suoritettavissa (ja peruttavissa) käyttäjän osaamistasosta riippumatta. (Weld D. & al. 2003)

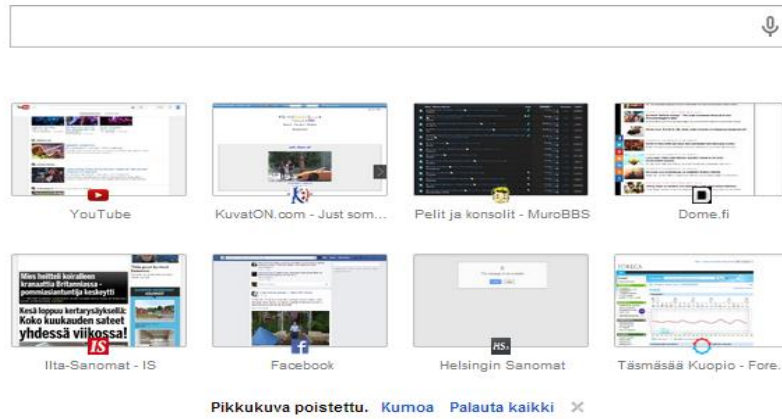
Wolfgang Stuerzlinger & al. ovat listanneet julkaisussaan neljä tärkeintä kriteeriä toimivalle kustomoitaville käyttöliittymälle (Stuerzlinger & al 2006):

1. **Nopea, yksinkertainen ja tehokkuutta parantava:** Käyttäjien on pystyttävä muokkaamaan käyttöliittymää ilman ennakkosuunnittelua, aina tarvittaessa ja nopeasti yksinkertaisia keinoja käyttäen niin, että se parantaa ohjelmalla suoritettavan työn laatua tai tehoa.
2. **Globaalit ja paikalliset muutokset:** Valtaosa kustomoitavista käyttöliittymistä tukee vain globaalia muutosta, joka voi pakottaa käyttäjät perumaan ne jossain vaiheessa. Paikallisella kustomoinnilla tarkoitetaan muutosta, jonka vaikutus pystytään rajaamaan tiettyjen parametrien (esim. tietty dokumentti) alle. Käyttäjien on pystyttävä määrittelemään kustomoinnin laajuus käyttäjäkohtaisesti, istuntokohtaisesti tai esim. dokumenttikohtaisesti.
3. **Kokonaisvaltainen kustomointi:** Käyttäjiä ei pitäisi rajoittaa ennalta määrättyihin kustomointimahdollisuuksiin vaan heidän pitää pystyä määrittelemään uusia. Rajoitteilla tarkoitetaan tässä tapauksessa esim. ruudulla olevien objektien sijaintia, ulkoasua tai toimintoa.
4. **Ohjelmien välinen kustomointi:** Käyttöliittymän kustomoinnin täytyy mahdollistaa eri ohjelmistojen yhdistäminen tai linkittäminen.

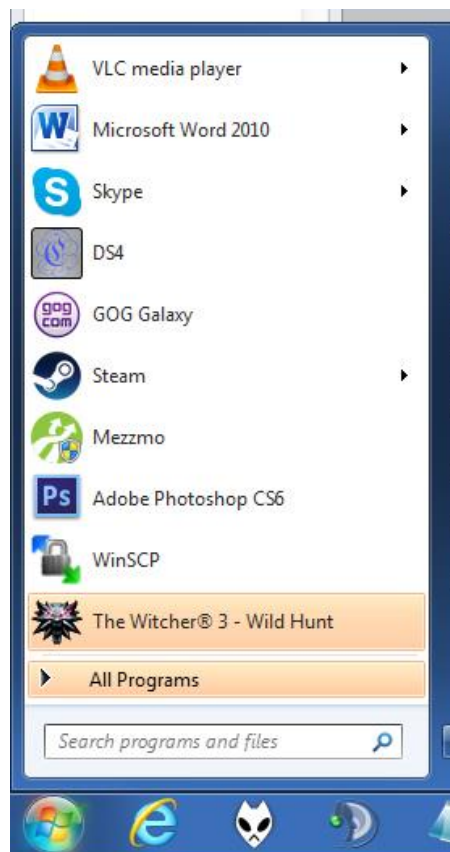
2.4 Adaptiiviset käyttöliittymät

Adaptiiviset käyttöliittymät ovat jo pitkään olleet tekoälytutkimuksen mielenkiinnon kohteena. Viimeaikaiset tutkimukset älykkäiden käyttöliittymien saralla ovat osoittaneet niiden edut perinteisiin käyttöliittymiin verrattuna, jotka ovat hyvin rajoittuneita mukautumaan vaihtelevaan käyttäjäkuntaan (Alvarez-Cortés V. 2007). Käyttöliittymän adaptiivisuudella tarkoitetaan ohjelman kykyä muokata käyttöliittymää ja sen toimintoja itsenäisesti tekoälyyn ohjelmoituja algoritmeja hyväksikäyttäen (Stuerzlinger & al. 2006). Adaptiivisilla järjestelmillä on myös etu perinteisiin staattisiin ja kustomoitaviin järjestelmiin verrattuna kun joudutaan olemaan vuorovaikutuksessa muiden järjestelmien kanssa, sillä ne pystyvät toimimaan monenlaisien eri interaktioiden kautta ilman erillisiä adaptoreita järjestelmien välillä.

Siinä missä kustomoitavuus perustuu käyttäjän omaan työhön käyttöliittymän ulkoasun muokkaamisen suhteen, adaptiivisuuden idea on tuottaa käyttöliittymässä tapahtuvat muutokset erilaisten algoritmien, toisin sanoen ohjelmaan sisäänrakennetun tekoälyn avulla. Algoritmien toimintaperusteena ja käyttöliittymän muokkaamisessa voidaan käyttää pohjana esim. käyttäjältä tulkittua työnkulkua tai muilta käyttäjiltä kerättyä dataa. Tuttuja esimerkkejä adaptiivisuudesta ovat mm. selainten etusivut (kuva 4) tai Windowsin käynnistä-valikko (kuva 5). Molemmissa näissä tapauksissa ohjelman käyttöliittymä mukautuu automaattisesti käyttäjän toimintoihin: selain tarjoaa uudelle sivulle kahdeksan käyttäjän useimmin vierailtua sivua nopeaa avausta varten ja Windowsin käynnistä-valikko tuo näkyviin käyttäjällä eniten käytössä olleiden ohjelmien pikakuvakkeet sekä näillä ohjelmilla viimeksi avatut tiedostot. Kuvan 5 esimerkissä käynnistä-valikko myös korostaa viimeisimpänä asennetun ohjelman.

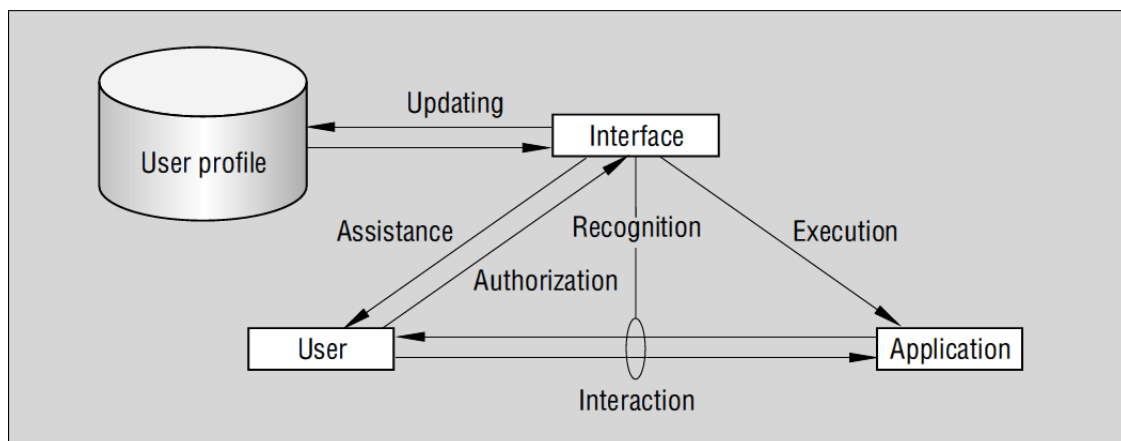


Kuva 3 - Google Chrome selaimen etusivun adaptiivisuus (Ggl15)



Kuva 4 - Windows 7 käynnistä-valikon adaptiivisuus (Mic15)

Kuva 6 havainnollistaa adaptiivisten käyttöliittymien toimintalogiikkaa melko yleisessä tapauksessa, jossa käyttäjäkohtaiset muutokset tapahtuvat käyttäjälle personalisoitua profiilia hyväksikäyttäen. Siinä käyttöliittymä tarjoaa ohjelmaa käyttävän rajapinnan lisäksi myös apua ohjelman käyttöön. Tekoäly lukee käyttäjän toiminnan käyttöliittymästä ja tekee sen pohjalta tulkinnan käyttäjän tavoitteista tai esimerkiksi mahdollisesta kokemustasosta. Sen pohjalta tekoäly tekee muutoksia käyttäjäprofiiliin ja voi jatkossa hyödyntää näitä muutoksia tarjoamalla tarkempaa apua käyttäjälle kun hän on sen tarpeessa.



Kuva 5 - Adaptive user interface toimintalogiikka (Liu J. & al. 2003)

Adaptiivisen käyttöliittymän käsite on sinänsä hyvin laaja ja adaptiivisuutta voidaan hyödyntää monella eri tasolla. Edellä mainitut esimerkit adaptiivisuudesta ovat älykkäiden käyttöliittymien yksinkertaisimmasta päästä, mutta niillä pystytään silti parantamaan eri käyttäjäryhmien käyttökokemusta ohjelmiston parissa. Tekoäly ja adaptiivisuus voivat kuitenkin olla myös vaatimuksia koko käyttöliittymän toiminnallisuuden kannalta. Tämän kaltaisia tapauksia ovat mm. puheen ymmärrys tai muut Natural Language Processing-järjestelmät, jossa käyttöliittymän on pystyttävä muuntumaan käyttäjän puhe- tai kirjoitustavan mukaan. (Benyon D. 1993)

Browne, Totterdell ja Norman (1990) ovat tutkimuksessaan määritelleet adaptiiviselle käyttöliittymälle neljä eri tasoa, jotka perustuvat järjestelmän ylläpitämien representaatioiden monimutkaisuudelle ja järjestelmän kyvyille käyttää näitä representaatioita (Browne D. & al. 1990):

1. **Yksinkertainen (simple):** Tekoälyllä ”kova koodatut” ärsyke-reaktio-toiminnot. Rajoittaa tekoälyn vaikutusmahdollisuudet yksinkertaisiin adaptioihin.
2. **Itse-säätöinen (self-regulating):** Tekoäly seuraa adaptaation jälkeen sen vaikutuksia käyttäjän työskentelyssä ja arvioi adaptaation onnistuneisuutta.
3. **Itse-sovitteleva (self-mediating):** Rakentaa malleja käyttäjän toimintojen pohjalta. Voi kehittää ja arvioida uusien mallien toimivuutta itsenäisesti ilman niiden käyttöönottoa.
4. **Itse-muuntautuva (self-modifying):** Pystyy muokkaamaan edellä mainittuja representaatioita tai muodostamaan kokonaan uusia.

Nykyään adaptiiviset käyttöliittymät perustuvat pääsääntöisesti johonkin seuraavista neljästä laskennallisista lähestymistavoista (Liu J. & al. 2003):

- **Bayesian network:** Todennäköisyysmalli, jossa esitetään joukko satunnaismuuttujia ja niiden väliset ehdolliset riippuvuudet suunnatun syklittömän verkon muodossa. Verkon solmukohdat esittävät satunnaismuuttujia ja kaaret niiden välisiä riippuvuuksia. Tätä mallia käytetään paljon adaptiivisissa käyttöliittymissä, kuten työpöytä-avustajissa, jotka pyrkivät ennustamaan käyttäjän lopullista päämäärää jo suoritettujen toimintojen perusteella. Tunnetuin järjestelmä, joka toteutti aikoinaan tätä laskentamallia oli surullisenkuuluisa Clippy.
- **Mixed-initiative:** Tekoäly ja käyttäjä käyvät yhteistä dialogia parantaakseen käyttöliittymää. Tutkimusta aiheesta on tehnyt paljon mm. Microsoft Research, jonka julkaisujen tuloksena on esitelty Microsoft Outlook-ohjelman LookOut-tekoäly. LookOut osaa tunnistaa sähköposteja sisällön perusteella ja ehdottaa sen perusteella esim. kokouksien sijaintia ja kestoja automaattisesti. (Horvitz E. 1999)

- **Model-based:** Pääsääntöisesti mallipohjainen tekoäly ei ole oppiva, vaan se perustuu valmiiksi kirjoitettuihin malleihin. Näiden mallien pohjalta tekoäly voi yrittää päätellä käyttäjän päämäärää ja toimia sen perusteella. Ymmärrettävästi tämä malli on rajoittunut toiminnallisuuden suhteen siinä mielessä, että sitä rajaa ohjelmoijan näkemys siitä, mitä käyttäjä ohjelmalta haluaa. (Liu J. & al. 2003)
- **Programming-by-example:** Tekoäly, jonka toiminnallisuus perustuu konkreettisten esimerkkien esittämiseen. Usein valtaosa opetuksesta tapahtuu loppukäyttäjän toimintoja seuraamalla. Normaali esimerkki järjestelmistä, jotka toimivat tällä tyylillä ovat mm. matkapuhelinten ennustavat tekstinsyötöt, jotka oppivat käyttäjän useimmin käyttämät sanat ja niiden käyttöjärjestyksen.

3 Nykypäivän tarpeita ja ongelmia

Perustana älykkäiden käyttöliittymien tarpeellisuudelle voidaan pitää yksinkertaista faktaa, että melkein minkä tahansa ohjelmiston loppukäyttäjien ryhmä on poikkeuksetta heterogeeninen, koska yksikään käyttäjä ei ole toisensa kopio ja jokaisella on oma tapansa työskennellä tai esimerkiksi fyysisiä rajoitteita (esim. värisokeus). Näissä tapauksissa suunnittelupäätökset tehdään yleensä kaikille ryhmän jäsenille sopivan mallin mukaisesti, jolloin lopullinen tuotos ei ole välttämättä kenellekään optimaalinen, eikä yksilölliseen käyttöliittymän räätälöintiin ole yleensä mahdollisuutta. Tämä pätee niin ohjelmistoteollisuudessa kuin kaikilla muillakin teollisuuden aloilla. Älykkäiden käyttöliittymien avulla pyritään saavuttamaan parempaa käytettävyyttä jokaiselle käyttäjälle. Tämä onnistuu tietysti vain jos pystymme kartoittamaan käyttäjien mahdolliset eroavaisuudet ja antamaan tekoälylle mahdollisuuden reagoida niiden perusteella. (Browne D. & al. 1990)

Ihmisten välisiä eroja, jotka vaikuttavat heidän tapaansa käyttää ohjelmistoja, on esitetty lukuisissa eri teoksissa. D. Browne & al. (Browne D. & al. 1990) listasivat heidän mielestään olennaisimmat erot joihin adaptiivisilla käyttöliittymillä pystytään vastaamaan: Psykologiset ja motoriset erot, kykeneväisyys, oppimiskyky, ymmärrys, odotukset, motiivit.

Vaikka adaptiivisuus ja automaattisesti käyttäjän mukaan muokkautuva käyttöliittymä voivat kuulostaa kaikin puolin hyvältä idealta, adaptiiviset käyttöliittymät ovat kuitenkin keränneet paljon negatiivisia kommentteja varsinkin puhuttaessa hyvästä käytettävyydestä. Kuten aikaisemmassa luvussa kaksi todettiin, tekoäly ei välttämättä toimi yhdessä hyvän käyttöliittymäsuunnittelun kanssa. Adaptiivisuuden vaara on sen potentiaali luoda hämmentäviä ja epäjohdonmukaisia muutoksia sekä viedä käyttäjältä tunne siitä, että ohjelma on hänen kontrolloitavissaan. Onkin suuri haaste luoda adaptiota ylläpitävälle tekoälylle algoritmeja, jotka pystyvät ennustamaan käyttäjän toimintaa ja tekemään sen pohjalta johdetut muutokset niin, että ne tapahtuvat käyttäjää hämmentämättä tai vaihtoehtoisesti liian vähäisesti ja tehottomasti (Weld D. & al. 2003). On valitettavan

yleistä, että adaptiivisuus tuo mukanaan käyttäjän kannalta epämiellyttäviä sivuvaikutuksia, kuten pikakuvakkeiden liikuttaminen tai poistaminen vastoin käyttäjän suostumusta. Myös erilaiset adaptiiviset (työpöytä)avustajat voivat olla käyttäjän kannalta enemmän ärsyttäviä kuin hyödyllisiä, jos ne eivät onnistu päättämään käyttäjän päämäärää onnistuneesti tai ovat muuten tehottomia antamaan apua käyttäjän ollessa hädässä. Tämän lisäksi aikaisemmat tutkimukset ovat osoittaneet, että käyttäjille on tärkeää pystyä kontrolloimaan automatisoituja järjestelmiä sekä tarvittaessa kumoamaan niiden tekemät muutokset. Tutkimukseen osallistuneet myös suosivat kustomoitavaa käyttöliittymää enemmän, kuin staattista (muuttumatonta) tai adaptiivista käyttöliittymää (Stuerzlinger & al. 2006). Toisaalta, kuten Höök julkaisussaan mainitsee (Höök K. 1998), on mahdotonta tehdä täysin puolueetonta vertailua eri käyttöliittymätyyppien (staattinen, kustomoitava ja adaptiivinen) välillä.

Seuraavat luvut käsittelevät adaptiivisiin käyttöliittymiin liittyviä ongelmia ja tarpeita nykypäivän teknologisessa ympäristössä. Luvussa 3.1 pohditaan adaptiivisten käyttöliittymien tarjoamia etuja järjestelmän kehittäjän kannalta. Luvussa 3.2 käydään läpi tietoturva ja adaptiivisten järjestelmien positiivisia ja negatiivisia vaikutuksia niin käyttäjän kuin kehittäjän kannalta ja luvussa 3.3 pohditaan pikaisesti sosiologisia ongelmia, jotka liittyvät tekoälyyn ja adaptiivisuuteen.

3.1 Teknologiset ja taloudelliset edut

Vaikka käyttöliittymien adaptiivisuuden voi helposti mieltää vain erilaisia käyttäjiä hyödyttäväksi ominaisuudeksi ja valtaosa tutkimuksista tuntuu keskittyneen tähän puoleen, luonnollisesti adaptiivisuus voi myös tarjota suuria hyötyjä itse järjestelmän kehittäjälleen. Jos käyttöliittymän adaptiivisuus voidaan määritellä osaksi ohjelmistoa niin, että pystytään osoittamaan sen parantavan järjestelmän käytettävyyttä, voi kehittäjä hyötyä siitä mm. seuraavilla tavoilla (Browne D. & al. 1990):

- **Potentiaalisen käyttäjäkunnan laajentaminen:** Tietokoneohjelmistojen markkinat ovat nykyään käsittämättömän laajat ja jatkavat kasvamistaan. Myös asiakaskunnat laajenevat ja monimutkaisemmatkin ohjelmat kuten esimerkiksi videon- tai kuvankäsittelyohjelmat kiinnostavat nykyään yhä enemmän harrastelijoita alan ammattilaisten lisäksi. Adaptiiviset käyttöliittymät antavat mahdollisuuden tarjota hyvä käyttökokemus kaikenlaisille käyttäjille ja näin luoda uusia potentiaalisia ostajaryhmiä.
- **Järjestelmän eliniän pidentäminen:** Järjestelmät suunnitellaan ja toteutetaan niiden pitkäikäisyyttä silmällä pitäen. Ohjelmistokehittäjän termeillä voidaan puhua siitä, että ohjelmat suunnitellaan ylläpidettäväksi eli tiedostetaan ennakkoon se fakta, että ohjelmaan tulee muutoksia aikojen saatossa. Jos järjestelmä pystyy adaptoitumaan näihin muutoksiin automaattisesti, puoliautomaattisesti tai ulkoisen vaikuttajan kautta, se kasvattaa järjestelmän oletettua elinikää markkinoilla.
- **Käyttäjän tavoitteiden mahdollistaminen:** Yllättävän usein järjestelmän retrospektiivinen analyysi paljastaa, että käyttäjät eivät hyödynnä järjestelmän täyttä toiminnallisuutta tai eivät toimi järjestelmässä optimaalisesti vaikka onnistuisivatkin saavuttamaan tavoitteensa. Esimerkiksi Adobe Photoshop tarjoaa käyttäjille mahdollisuuden muokata kuvan tallennuskokoa, mutta tämä jää monelta harrastelijakäyttäjältä huomaamatta ja he käyttävät sen sijaan eri sosiaalisten medioiden palveluiden (esim. Facebook) sisäänrakennettua kuvanlaadun pienennystä, jossa kuvanlaatu kärsii huomattavasti tarvittua enemmän. Adaptiivinen käyttöliittymä voisi reagoida tämän kaltaisessa tapauksessa vaikkapa analysoimalla käyttäjän osaamisen tason ja korostamalla mahdollisuutta muokata kuvan laatua tai pikselimäärää ennen tallennusta.
- **Käyttäjän tarpeiden tyydyttäminen:** Käyttäjät voivat usein turhautua työskennellessään tietokoneohjelmien kanssa. Tietämättömyys ja kokemattomuus ohjelmistoista saattaa synnyttää liian suuria odotuksia mm. toiminnallisuuden ja joustavuuden suhteen. Älykkäillä käyttöliittymillä

pystytään yhä paremmin vastaamaan näihin yksilöllisiin odotuksiin tekemällä ohjelmistoista entistä helpommin lähestyttäviä.

- **Kommunikaation tarkkuuden parantaminen:** Joissain tapauksissa olemassa olevan järjestelmän muokkaamisen päämääräisenä tavoitteena on parantaa ohjelman ja käyttäjän välisen kommunikaation tarkkuutta. Työnkulun aikana käydyn kommunikaation tarkkuus onkin yksi olennaisimmista hyvän käytettävyyden vaatimuksista. Tällainen tarkkuuden parantaminen voikin olla vahva peruste adaptiivisuuden toteuttamiseen.
- **Nopeuttaa optimaaliseen työskentelynopeuteen pääsemistä:** Ennen kuin tuottelias työnkulku saavutetaan, käyttäjät joutuvat käyttämään osan ajastaan ohjelman käytön oppimiseen. Tämä voi olla työn kannalta kallis ja aikaa hukkaava ajanjakso, joka voi vaatia muitakin resursseja kuten koulutusta. Myös yleinen työn laatu voi kärsiä. Adaptiivisuudella on mahdollista vähentää tämän ajanjakson kestoa ja mahdollista koulutuksen tarvetta, jos adaptiivisuutta pystytään käyttämään tehokkaasti hyväksi helpottamaan työntekijän oppimista.
- **Nopeuttaa ohjelman sisäistämistä:** Vaikka käyttäjä osaisi käyttää ohjelman komentoja sujuvasti, hän ei ole välttämättä sisäistänyt kaikkia komentoja tai niiden yksityiskohtia. Esimerkiksi komentorivillä käyttäjä voi osata lauseet, joilla poistaa tai kopioida tiedostoja, mutta ei ymmärrä kaikkien operaattoreiden tarkoitusta lauseissa. Joissain tapauksissa käyttäjät voivat saada hyötyä siitä, että ymmärtävät ja osaavat käyttää näitä operaattoreita eri tavoin parantaen tehokkuuttaan. On mahdollista toteuttaa adaptiivisuutta käyttöliittymässä niin, että käyttäjä oppii sisäistämään eri työkalujen pienimmätkin vivahteet tarkemmin siltä varalta, että niistä on jatkossa enemmän hyötyä kuin käyttäjä alun perin aavistaakaan.

Sen lisäksi, että ohjelmat voivat adaptiivisuuden avulla reagoida käyttäjän yksilöllisiin toimintamalleihin tai rajoitteisiin, voidaan nykyään käyttää saatavilla olevaa teknologiaa myös apuna reagoimaan tilannekohtaisiin muutoksiin. Erilaisilla sensoreilla ja muilla apuvälineillä kuten kameralla ohjelmat voivat seurata käyttötilannetta ja vaikuttaa niistä kerätyn informaation avulla. Esimerkiksi puhetta

tulkitsevat ohjelmat voivat havaita käyttäjän stressitilat ja yrittää operoida paremmin niiden puitteissa jos ohjelma havaitsee, että käyttäjä alkaa stressaantua tietynlaisista ohjeista (Sebe N. 2009). Samalla tavoin Nintendon uusi käsikonsoli New 3DS, joka tarjoaa käyttäjälle kolmiulotteista stereokuvaa osaa seurata kameran avulla käyttäjän sen hetkistä peliasentoa ja korjata kuvaruutuun lähetettävän 3D-videon tulokulmaa niin, että kuvanlaatu ei kärsi katselukulmaa vaihtaessa. On selvää, että tällainen adaptaatio mahdollistaa järjestelmien käyttämisen useammassa tilanteessa ja on varmasti tarpeellista varsinkin tulevaisuudessa, kun erilaiset tietokonejärjestelmät kuten älykellot ja älylasit, jotka kulkevat ihmisten matkassa paikasta toiseen yleistyvät.

Adaptiivisuus voi myös tarjota taloudellisesti paremman vaihtoehdon järjestelmän päivittämiselle uuden kiinteän käyttöliittymän laatimisen sijaan. Usein järjestelmän elinkaaren aikana huomataan tarve muutokselle. Tämän muutoksen toteuttaminen vaatii yleensä kattavan järjestelmäanalyysin, jolla kartoitetaan nykyisen järjestelmän ongelmat sekä vaatimukset uusille toiminnoille. Jos järjestelmäanalyysin lopputulos sallii adaptiivisen muutoksen toteuttamisen, voi siitä olla suuri hyöty tuotannolliselta näkökannalta mikäli adaptiivisuudella pystytään vastaamaan käytettävyyksivaatimuksiin paremmin kaikkien käyttäjäryhmien kannalta. (Benyon D. 1993)

Vaikka käyttöliittymän adaptiivisuudella voidaan saavuttaa suuria etuja taloudellisesti, niin täytyy myös muistaa, että adaptiivisuuden toteuttaminen vaatii suurta lisätyötä ohjelmoijilta, kuten myös yksinkertaisemmin toteutettava kustomoitavuus (Stuerzlinger & al. 2006). Adaptiivista käyttöliittymää ja algoritmeja joihin adaptiivisuus perustuu on välillä vaikea toteuttaa niin, etteivät muutokset tapahdu käyttäjää häiritsevästi ja hyvinkin toteutettu adaptiivisuus ei aina kulje käsi kädessä vaatimustenmäärittelyjen kanssa. Myös ohjelmaan jälkeensä lisättävä adaptiivisuus vaatii enemmän työtä sekä tarkemman analyysin kuin ei-adaptiivisen muutoksen toteutus (Benyon D. 1993).

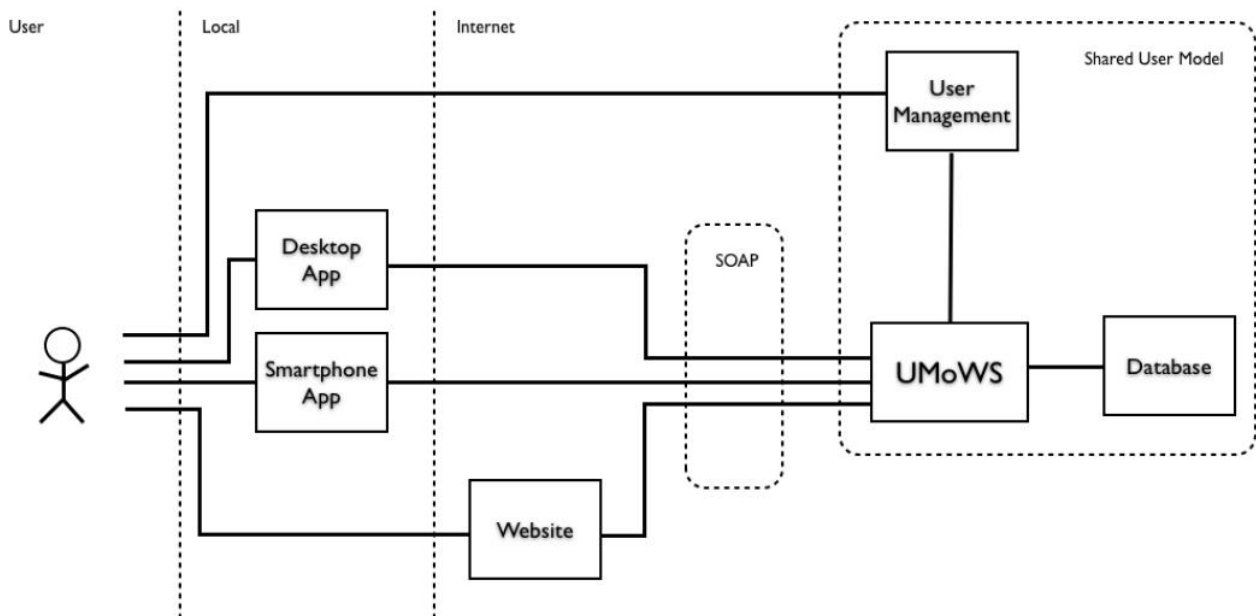
3.2 Käyttäjäprofiilit ja tietoturva

Eräs suosituimmista ja yleisimmistä älykkäiden käyttöliittymien ilmentymistä ovat personalisoidut sovellukset ja sivustot, jotka keräävät jatkuvasti lisää suosiota ja tänä päivänä monet käyttäjät ovatkin vuorovaikutuksessa jonkinlaisen adaptiivisen järjestelmän kanssa jo päivittäin. Nykyaikaiset adaptiiviset järjestelmät perustuvat yleensä kahteen tekijään: käyttäjän tietämykseen ja käyttäjän aikaisemmin osoittamiin mielenkiinnon kohteisiin. Normaalisti varsinkin viihdekäyttöön suunnitellut sovellukset painottavat enemmän käyttäjän mieltymyksiin mukautuvaa adaptiivisuutta kun taas koulutus- ja osaamispainotteiset ohjelmistot käyttävät hyväkseen tietoa käyttäjän kokemustasosta. (Montague K. & al. 2010)

Jotta adaptiiviset järjestelmät pystyvät operoimaan käyttäjän kokemuksen tai intressien pohjalta, täytyy järjestelmällä olla toimintalogiikkansa takana dataa, jonka avulla tehdä arvionsa käyttäjästä. Tästä datasta tehdyistä päätelmästä voidaan käyttää termiä käyttäjämalli, *user model*, tai tutumpaa nimeä profiili (Montague K. & al. 2010). Yksinkertaisimmillaan käyttäjämallit voivat perustua mm. käyttäjän nimeen ja työnkuvaan tai hänelle määriteltyihin käyttöoikeuksiin. Tämän kaltaiset järjestelmät ovat yleisimmin käytössä luomaan käyttäjille omat ja yksinkertaiseen tekoälyyn pohjautuvat profiilit, joiden avulla pystytään mm. parantamaan järjestelmien tietoturvaa rajaamalla tietyn käyttäjäryhmän oikeudet dataan. Esimerkiksi yhtiön sisäisissä ohjelmistoissa projektien hinta-, tarjous- tai vaikkapa palkkatiedot on normaalia rajata vain tiettyjen käyttäjäryhmien saataville tietoturvan parantamiseksi. On kuitenkin usein tapauksia, jossa eri käyttäjäryhmät työskentelevät samojen asioiden parissa tarvitsematta kaikkea tarjolla olevaa informaatiota, joten erillisten ohjelmistojen ostaminen tai kehittäminen ei ole kannattavaa. Tällaisissa tapauksissa yhteen järjestelmään liitetty älykäs adaptaatio voi toimia tietoturvaa parantavana tekijänä rajoittaen käyttäjille tarjolla olevaa informaatiota.

Käyttäjien mieltymyksiin perustuvat järjestelmät voivat olla parhaimmillaan todella monimutkaisia tekoälynä puolesta ja pystyvät yhdistelemään useiden eri käyttäjämallien tietoja saadakseen muodostettua tarkempia ennustuksia käyttäjien

tulevista mieltymyksistä. Yleinen toimintamalli tämän tyyppisille järjestelmille ja siitä kuinka käyttäjämalleja voidaan käyttää hyväksi on esitetään kuvassa 7. Kun adaptiivisuus perustuu jaettuihin käyttäjämalleihin, jokaisen sovelluksen täytyy kyetä päivittymään ja tallettamaan muutoksia. Tämä tapahtuu muodostamalla yhteys verkon yli erilaisia rajapintoja, kuten SOAP (Simple Object Access Protocol), hyväksi käyttäen tietokantaan, jossa tiedot eri käyttäjistä ovat tallessa. Konkreettinen adaptiivisuus ei muodostu automaattisesti näistä tiedoista, vaan tietoihin pohjautuvat muutokset tehdään erillisen palvelun avulla, jota kuvassa 7 havainnollistaa UMoWS (User Model Web Service).



Kuva 6 - Käyttäjämallien toiminta verkossa (Montague K. & al. 2010)

Siinä missä käyttäjän käyttöoikeuksien mukaan adaptoituvaa käyttöliittymää voidaan käyttää hyväksi parantamaan yleistä tietoturvaa, voivat nämä ”julkisiin” profiileihin perustuvat järjestelmät olla uhka käyttäjän omalle tietoturvalle, kuten Kyle Montague & al. (Montague K. & al. 2010) tutkimuksessaan huomauttavat. Harvalla käyttäjälle on täysin selvää, kuinka suuri osa heidän toiminnoistaan lisätään profiiliin ja kuinka

sitä käytetään hyväksi järjestelmän toiminnoissa tai jaetaan muiden käyttäjien kesken. Esimerkiksi Netflix on ollut hyvin avoin siitä, että käyttäjien luomia profiileja hyväksikäyttäen tehdään analyyskejä, joiden pohjalta Netflix suosittelee ja jopa käsikirjoittaa sekä kuvaa uusia tv-sarjoja ja elokuvia käyttäjille. Samoin Google, Amazon, Facebook jne. käyttävät käyttäjäprofiilien tarjoamaa informaatiota muokatakseen sivustojensa mainokset käyttäjilleen persoonallisesti houkutteleviksi. Tällaisissa tapauksissa käyttäjän tiedot ovat täysin niitä käyttävän yrityksen päätäntävällän alla ja pahimmassa tapauksessa voivat päätyä väriin käsiin. Vaikka esimerkeissä käytetyt yhtiöt edustavatkin tietoturvasa tasolta tämän kaltaisten palvelujen huipputasoa on syytä muistaa, että vastaavia palveluita on lukemattomia ja kaikissa niissä ei ole panostettu tietoturvaan samoissa määrin.

3.3 Sosiologiset ongelmat

Vaikka adaptiivisuudella pystyttäisiin vastaamaan sujuvasti kaikkiin edellä mainittuihin käyttäjien eroavaisuuksiin, törmätään tekoälyn ohjelmoinnissa usein sosiologisiin ilmiöihin, johon Lieberman viittaa artikkelissaan (Lieberman H. 2009) nimillä ”pieni määrä loputtomuutta” (*small size of infinity*) ja ”katoava tekoäly” (*disappearing AI*). Kun puhutaan tekoälyn ongelmista kuten puheen tai kirjoituksen ymmärtämisestä voi helposti tuntua, että mahdollisia variaatioita on loputon määrä ja ongelma on mahdoton ratkaista. Monet aikaisemmista epäonnistumisista tekoälyn parissa voidaankin laskea sen syyksi, että tämä loputon variaatioiden määrä yritettiin laittaa liian harvoihin kategorioihin sen ajan teknologisten rajoitteiden puitteissa. Kuitenkin pikkuhiljaa tietokoneiden laskentatehon, muistin määrän jne. kehittyessä näissä ongelmissa joita aikoinaan pidettiin mahdottomana ratkaista (ainakaan toimivissa määrin) on tehty hyvää edistystä ja edistyneemmät tekoälyn ongelmat kuten puheen ymmärtäminen alkavat olla arkipäivää ja luotettavasti toimivia jokapäiväisessä käytössä.

Valitettavan usein tämä arkipäiväistyminen johtaa siihen, että onnistuneet tekoälysovellukset, jotka ovat täysin tekoälytutkimuksen aikaansaannosta sulautuvat osaksi niihin liitettyjä tuotteita ja tekoäly tutkimuskohteena jää ilman kiitosta. Harva

edes miettii Googlea tekoälynä vaikka he käyttävät siihen liittyvää tekstinprosessointia päivittäin ja monelle robotti-imurit ovat vain normaaleja kodinkoneita. Näin myös käyttöliittymiin liitetty tekoäly unohtuu monelta, silloin kun se oikeasti parantaa käytettävyyttä. Toisaalta vastakkaisessa tilanteessa epäonnistunut tekoäly korostuu helposti tuotteissa ja kerää negatiivista mainetta kaikille saman tutkimuskohteen projekteille. (Lieberman H. 2009)

Myös ihmisten odotukset tai käsitykset siitä, millä tavalla tekoälyn kuuluisi näkyä käyttöliittymissä, voi olla välillä vääristynyt. On hyvin normaalia ajatella, että on ohjelmien ja järjestelmien työ mukautua ihmisten mukaan eikä toisinpäin. Tämä ajattelu voi kuitenkin johtaa niin käyttäjät kuin suunnittelijatkin harhaan saaden heidät uskomaan, että älykkään käyttöliittymän tulisi käyttäytyä kuin kanssaihminen; muuttaen käytöstään luontevasti käyttäjän tietämystä, kykyjä ja mieltymyksiä mukailevaksi ja mielellään kehittyntä dialogia hyväksi käyttäen. Käytännössä tämä kuitenkin harvoin toteutuu ja valtaosa menestyneimmistä älykkäiden käyttöliittymien sovelluksista ovatkin vain pieniä adaptaatioita yksinkertaisesta tiedosta käyttäjän mieltymyksistä tai poimien tätä samaa tietoa muilta käyttäjiltä. (Höök K. 2000)

4 Tunnettuja älykkäitä käyttöliittymäsovelluksia

Adaptiivisuus ja tekoäly käyttöliittymissä ei ole nykypäivänä lainkaan harvinaista ja erilaisia kaupallisesti menestyneitä käyttöliittymäsovelluksia, jotka pohjautuvat näille teknologioille on olemassa lukemattomia. Adaptiivisuus on myös levinnyt alkuaikeiden ammattilaiskäyttöön tarkoitetuista järjestelmistä nettisivuihin, peleihin sekä muihin arkipäiväisiin sovelluksiin. Myös adaptiivisuuden ilmeneminen ja mahdolliset käyttötarkoitukset vaihtelevat järjestelmästä toiseen. Tässä luvussa käydään läpi muutamia tunnettuja tekoälyjä sekä järjestelmiä, joissa nämä tekoälyt toimivat.

Luvussa 4.1 käydään läpi ehkä kuuluisinta esimerkkiä adaptiivisten käyttöliittymäavustajien saralla eli Microsoftin Office apuria Clippyä ja Lumiere-projektia johon Clippy sekä Microsoftin monet muut tulevat älykkäät käyttöliittymäsovellukset perustuvat. Luku 4.2 käsittelee Microsoftin sähköpostiohjelma Outlookin taustalla toimivaa tekoälyä LookOut sekä Microsoft-agentteja. Tuoreempia esimerkkejä adaptiivisuudesta esitellään luvussa 4.3, jossa käydään läpi mobiilipuolen puheentunnistukseen perustuvia tekoälyavustajia: Applen Siri, Microsoftin Cortana ja Googlen Google Now. Luku 4.4 käsittelee adaptiivisten käyttöliittymien kaupallisempaa puolta verkkokauppa Amazonin esimerkkien kautta.

4.1 Lumiere Project – Clippy

Eräs mielenkiintoisimmista ja varmasti tunnetuimmista yrityksistä soveltaa tekoälyä käyttöliittymään graafisen avustajan muodossa on Microsoftin Clippy – Office Assistant. Tämä, ehkä jopa huonon maineen omaava tekoälyapuri ei alun alkaen ollut vain pieni sivuprojekti Office-tuotteiden kylkiäisenä, vaan tutkimusprojekti Lumierin aikaansaannosta. Lumiere-projekti, jonka tulokset Microsoft julkaisi vuonna 1998 (Horvitz E. & al. 1998), käsitteli ja tutki adaptiivisen käyttöliittymän perusteita eli käyttäjämallien selvittämistä Bayesian-todennäköisyysmallien avulla.

Tutkimus itsessään oli onnistunut tulkinta siitä, kuinka tekoälyn avulla voidaan luoda käyttäjämalli käytännössä minkä tahansa ohjelmiston käytön perusteella ja tarpeen tullen helpottaa tai nopeuttaa käyttäjän pääsyä päämääräänsä. Siitä huolimatta kaupallisessa mielessä Clippy ei saavuttanut yhtä suurta menestystä. Voidaan ehkä jopa sanoa, että suuren yleisön silmissä usko oikeasti toimiviin tekoäly-kumppaneihin avustamassa käyttäjiä ohjelmistojen käytössä romahti vuosiksi.

Vaikka Lumiere-projektin tutkimusta voidaan pitää käyttöliittymätekoälyn ja adaptiivisuuden uranuurtajana, Clippyn kaupallisesta versiosta tuli mittava epäonnistuminen. Tutkimuksessa esiteltiin kaikin puolin teoreettisella tasolla toimiva versio hyödyllisestä tekoälystä, käytännössä teknologiset ja taloudelliset rajat tulivat vastaan. Alla muutamia vakavia ominaisuuksia (tai puutteita), jotka Microsoftin tutkimusryhmä on esitellyt ja laatinut tutkimuksessaan, mutta eivät päässeet Clippyn tekoälyyn saakka:

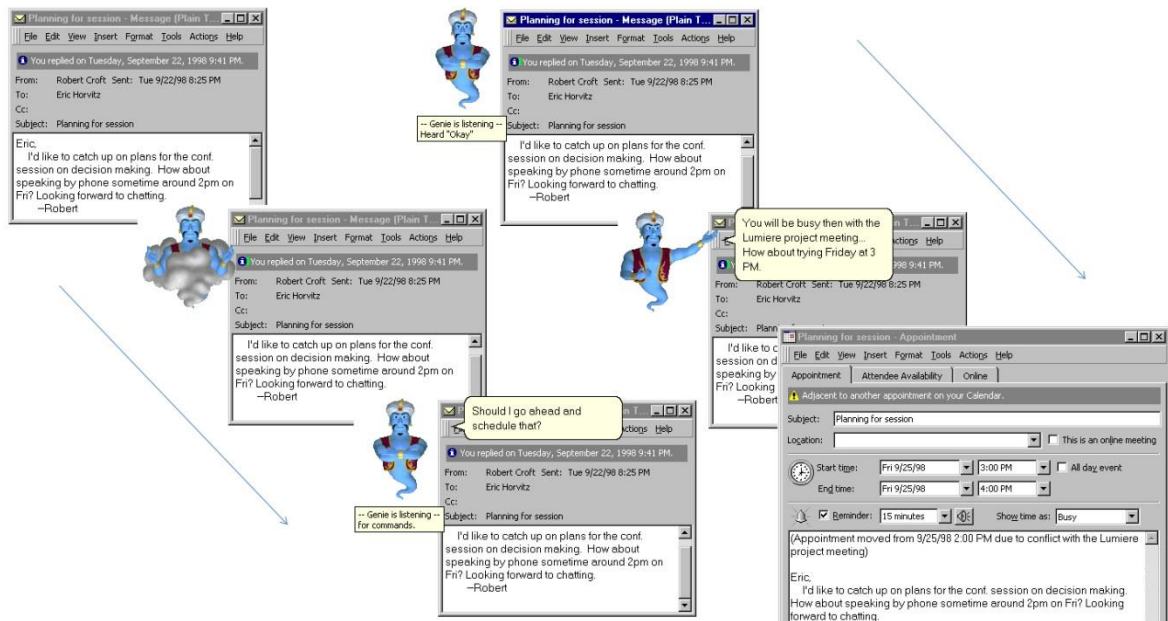
- Ei käyttäjäprofiileja.
- Ei päättelyä käyttäjän kokeneisuudesta (vain viimeisin ja hyvin pieni määrä käyttäjän tekemistä toiminnoista osana tulkintaa, kaikki aikaisemmat työt jäivät siis tekoälyn ”kantaman ulkopuolelle”).
- Tekoälyllä ei yhteyttä graafisen käyttöliittymän toimintojen ja kirjoitettujen komentojen välillä.
- Ei tekoälyn tulkintaa siihen, milloin käyttäjä tarvitsee apua tai vinkkejä. Clippy ei käyttänyt tutkimuksessa kehitettyä metodologiaa, jolla tekoäly pystyi tulkitsemaan käyttäjän tarpeen avulla juuri sillä hetkellä. Sen sijaan käytössä oli yksinkertainen päälle liimattu matemaattinen kaava, jonka takia Clippy tarjosi apua aivan liian usein.

Microsoft ei ole antanut virallista lausuntoa siitä, miksi Clippy jäi vaille näin oleellisia tekoälyn toimintoja. Lumiere-projektissa osallisena ollut Eric Horvitz on kuitenkin antanut epäviralliseksi syyksi niinkin yksinkertaisen syyn, kuin tilan puute myyntiin lähteneillä levykkeillä.

4.2 LookOut ja Microsoft-agentit

Microsoft ei antanut periksi aktiivisten käyttöliittymäagenttien kanssa Clippyn jälkeen vaan jatkoi kehitystä. Clippyn jälkeen kehitettiin Microsoft-agentit, eli tekoäly joka käyttää hyväksi animoituja hahmoja vuorovaikutuksessa käyttäjän kanssa. Clippystä poiketen mukana oli paljon kehittyneempää tekoälyteknologiaa kuten puheen ymmärtämistä, jonka avulla käyttäjä pystyi antamaan komentoja agentille tai muuntamaan puhetta tekstiksi. Käyttämällä hyväksi näihin agentteihin ohjelmoituja rajapintoja pystytään luomaan graafinen ilmentymä eli tässä tapauksessa animoitu työpöytäavustaja ilmentämään ohjelman taustalla toimivan tekoälyn toimintoja.

Microsoft Outlook oli yksi ensimmäisiä ohjelmia, joka käytti hyväksi Microsoft-agenttia. Tekoälyprojekti Outlook-ohjelman takana kulki nimellä LookOut ja sen avulla pyrittiin tehostamaan sähköpostiohjelman käyttöä mm. tulkitsemalla ennakkoon saapuneiden viestien sisältöä ja tarjoamaan käyttäjälle valmiiksi vastausvaihtoehtoja tai erilaisia toimintoja vastausviestiin (kuva 8). Esimerkiksi LookOut pystyy tekemään johtopäätöksiä tekstin sisällön ja lähetetyn päivämäärän välillä: jos viesti on lähetetty eilen ja teksti sisältää sanan ”huomenna”, ohjelma ymmärtää sen tarkoittavan sen hetkistä päivää. LookOut kykenee myös tulkitsemaan tekstistä tietyt viittaukset kuten ”päivällinen” tai ”kokous” ja ehdottamaan mahdollisia vapaita aikoja käyttäjän kalenterista näille tapahtumille sekä päättämään tapahtuman mahdollisen keston mm. käyttäjän tapahtumahistoriaa hyväksikäyttäen.



Kuva 7 - Outlook käyttöliittymäagentti Genie (Mic15)

Microsoftin visuaaliset apurit eivät koskaan nousseet huimaan suosioon, mutta LookOut-projektin mukana kehitetty tekoäly toimii edelleen Outlook-ohjelman taustalla, mutta ilman visuaalista representaatiota. Microsoft-agenttien tuki loppui Windows XP:n jälkeisissä käyttöjärjestelmissä ja agenttien toimintoihin liitetyt ohjelmat eivät ole enää saatavilla Windows 7 tai uudemmissa tuotteissa. Vaikka animoidut agentit ovat olleet vähässä käytössä viimeisen vuosikymmenen, ne alkavat saamaan uutta muotoa mm. Applen Siri-tekoälyn ja tulevan Microsoftin Cortanan myötä.

4.3 Siri, Cortana ja Google Now

Muutaman 90-luvun lopulla tapahtuneen valitettavan epäonnistumisen jälkeen personalisoidut visuaaliset käyttöliittymätekoälyt katosivat hetkeksi järjestelmien käyttöliittymistä. Vasta viime vuosina tekoälystä ja sen avulla toteutettavasta adaptiivisuudesta on pyritty jälleen tekemään käyttäjälle näkyvämpää varsinkin mobiilipuolen eri käyttöliittymäassistenttien avulla. Niin Apple, Microsoft kuin Googlekin ovat luoneet omat puhettunnistukseen perustuvan tekoälyn avustamaan käyttäjiä käyttöjärjestelmiensä parissa. Siri on Applen tuottama järjestelmä, joka

toimii iOS 5 tai uudemmissa tuotteissa. Cortana taas puolestaan Microsoftin luoma tekoäly joka löytyy Windows Phone 8.1, Microsoft Band ja Windows 10 tuotteista. Nämä molemmat ovat vahvasti personalisoituja systeemejä, joille molemmat yhtiöt ovat pyrkineet luomaan omaa persoonaa ja ihmismäisiä ”luonteenpiirteitä” siinä missä Googlen puheentunnistusta käyttävä tekoäly Google Now on pyritty pitämään enemmän konemaisena. Tätä ilmentää suoraan esimerkiksi se, että Cortana on nimetty Microsoftin suosituissa Halo-pelisarjassa esiintyvän naishahmoisen tekoälyn mukaan (kuva 9).



Kuva 8 - Cortana (Nelson F. 2014)

Puheentunnistusta käyttävät assistentit ovat löytäneet uuden kotinsa mobiililaitteista muutamasta luonnollisesta syystä. Näiden laitteiden käyttöpinta-ala on huomattavasti pienempi kuin pöytätietokoneen tai kannettavan, jolloin tilaa pikakuvakkeille tai muulle informaatiolle on vähemmän ja erilaisten komentojen kirjoittaminen on hitaampaa ja vaivalloisempaa. Käyttämällä hyväksi puheentunnistusta, pystytään kiertämään helposti nämä ongelmat ja esimerkiksi käynnistämään ohjelmia, joiden pikakuvakkeet eivät ole suoraan näkyvillä tai muuntamaan käyttäjän puhetta suoraan tekstiksi. Vaikka puhetta ymmärtävää tekoälyä ei helposti miellä adaptiivisuuden muodoksi, tarkemman toiminnallisuuteen perehtymisen jälkeen huomataan helposti,

että käyttäjien toimintaan mukautuminen on yksi tärkeimpiä valttikortteja tällaisen tekoälyn menestymistä ajatellen.

Sen sijaan, että Cortana ja Siri käyttäisivät puheentunnistusta vain ottaakseen vastaan yksinkertaisia komentoja tai muuntaisivat käyttäjän puhetta tekstiksi, niihin ohjelmoitu tekoäly kykenee yhdistelemään käyttäjien komentoja ja oppimaan käyttäjän aksenttia, kieltä ja ilmaisua pikkuhiljaa käytön aikana. Kaikki kolme tekoälyä pystyvät mm. asettamaan hälytyksiä, soittamaan puheluita, muokkaamaan asetuksia, hakemaan reittejä tai vaikkapa vastaamaan käyttäjän esittämiin pyyntöihin tai kysymyksiin käyttämällä hyväksi eri hakupalveluita. Uusimpana ja mahdollisesti kehittyneimpänä tekoälynä Microsoftin Cortana kykenee myös käsittelemään käyttäjien puhetta kontekstissa, esimerkiksi osaten yhdistää sanan ”sisko” tai ”äiti” tiettyihin henkilöihin tai asettamaan muistutukset toimimaan tietyn kellonajan sijaan vaikkapa seuraavaan kertaan kun käyttäjä on tietyssä sijainnissa tai puhuu tietyn henkilön kanssa.

Vaikka teknologia puheenymmärryksen takana on kehittynyt paljon, kaikki kolme tässä luvussa esiteltyä tekoälyä ovat vielä kaukana täydellisestä. Jokaisella niistä on omat pienet ongelmansa puheen tulkinnassa, esimerkiksi Cortanalla on vaikeuksia tulkita välimerkkejä käyttäjän puheessa. Myös luonnollinen keskustelu käyttäjän ja tekoälyn välillä on olematonta ja näistä kolmesta tekoälystä vain Cortana kykenee yhdistämään kahden esitetyn kysymyksen välillä olevan yhteyden kun taas Siri ja Google Now käsittelevät jokaisen käyttäjän lauseen yksittäisenä kyselyinä (Dolcourt J. 2014). Siitä huolimatta kaikki kolme tekoälyä toimivat loistavina esimerkkeinä mobiililaitteiden adaptiivisuudesta, jossa tekoäly pyrkii parantamaan käyttäjän tehokkuutta järjestelmän käytössä muokkaamalla omaa käytöstään käyttäjän tapoja mukailevaksi.

4.4 Adaptiiviset verkkokaupat ja muut verkkopalvelut

Internetin erilaiset kauppasivustot ja muut kaupalliset verkkopalvelut tuntuvat olleen viime vuosikymmenenä adaptiivisuuden tienraivaajia, mutta usein näissä esiintyvä

adaptiivisuus jää käyttäjältä huomaamatta. Syynä tähän on se, että usein mm. verkkokauppojen sivuilla esiintyvä adaptiivinen ulkoasu ei ole suoraan tarkoitettu käyttäjän etua palvelevaksi vaan kauppa pyrkii mainostamaan käyttäjille sopivia tuotteita. Tuotteet, joista käyttäjä voisi mahdollisesti olla kiinnostunut tulkitaan yleensä käyttäjän ostos- tai katseluhistorian pohjalta tai tekemällä viittauksia muiden käyttäjien profiileihin, jotka ovat mahdollisesti ostaneet samoja tuotteita tai osoittaneet kiinnostusta vastaaviin tuotteisiin. Monesti tämä tuotteiden suosittelu voi tarkkuudestaan huolimatta olla käyttäjän kannalta kovin tungettelevaa tai ei palvele käyttäjän tehokkuutta, mutta kaupan kannalta tällainen adaptiivinen mainostaminen ratkaisee tehokkaasti ongelman rajallisen mainostilan sekä lukemattomien myytävien tuotteiden kanssa. Optimaalisessa tilanteessa sekä kauppa että ostaja hyötyvät.

Tämä alunperin verkkokaupoista kuten Amazon ja eBay liikkeelle lähtenyt adaptiivisuuden trendi on levinnyt hyväksi todetun toimivuutensa takia lukemattomiin eri verkkopalveluihin, kuten Netflix, YouTube, Spotify ja muut vastaavat. Suuret käyttäjämäärät ja tarkat kuvaukset eri tuotteiden sisällöstä sekä viittaukset vastaaviin tuotteisiin ovat luoneet mahdollisuuden tarjota käyttäjille yhä tarkemmin heidän makuihinsa sopivia tuotteita. Mitä suurempi määrä käyttäjiä, sitä tarkempia ennustuksia myös uusien käyttäjien mieltymyksistä pysytään tekemään jo vähäisellä alkudatalla.

Koska tällaiset palvelut perustuvat hyvin tarkkoihin käyttäjäprofiileihin on käyttäjien kuitenkin syytä muistaa erilaiset tietoturvariskit, jotka liittyvät omien henkilökohtaisten tietojen säilyttämiseen ulkopuolisissa palveluissa. Erilaiset jättimäisiin verkkokauppoihin ja -palveluihin iskeneet verkkohyökkäykset ovatkin olleet viime vuosina paljon uutisissa.

5 Pohdinta ja jatkotutkimusaiheet

Adaptiivisuus on ollut käyttöliittymäsuunnittelijoiden mielenkiinnon kohde jo 1980-luvulta alkaen. Samoin kuin käyttöliittymät yleensä ottaen, adaptiivisuus sekä käyttöliittymiin kytketyt tekoälyt ovat kehittyneet huimasti viimeisen vuosikymmenen aikana. Myös yleinen käsitys niin käyttäjien kuin ohjelmistotuottajien keskuudessa siitä millaisia hyötyjä järjestelmissä pystytään saavuttamaan adaptiivisuuden avulla on parantunut. Vaikka tekoälyllä ja älykkäillä käyttöliittymillä on ollut aikojen saatossa omat vastoinkäymisensä markkinoilla sekä kohtaamastaan kritiikistä huolimatta, adaptiivisuudesta on selvästi tullut pysyvä osa moderneja käyttöliittymiä.

Kun järjestelmien toimintaperiaatteisiin perehtyy tarkemmin ja alkaa kiinnittämään tarkemmin huomiota adaptiivisuuteen sen eri muodoissa, tajuaa kuinka monet arkipäiväisistä ohjelmistoista ja palveluista kykenevät mukautumaan käyttäjän toimintamallien mukaisesti tai tarjoaa käyttäjälle apua kun hän sitä tarvitsee. Välillä on myös helppo unohtaa, että varsinkin internetissä erilaisilla nettikaupoissa, sosiaalisessa mediassa jne. adaptiivisuus voi olla ratkaiseva tekijä esim. onnistuneessa mainonnassa, jossa mainokset tai tuotekuvaukset suunnataan käyttäjille personalisoidusti. Vaikka tämä adaptiivisuuden muoto ei ehkä olekaan aina käyttäjiä palvelevaa, on se siitä huolimatta yksi adaptiivisten käyttöliittymien ehkä yleisimmistä malleista ja käytössä miltei jokaisella sivustolla mm. Googlen tarjoamien mainosten muodossa.

Arkipäiväisten sovellusten käytettävyyden parantamisen lisäksi adaptiivisuus on myös mahdollistanut uusien teknologioiden käyttöä (kuten Nintendon 3DS-konsolin 3D-stereokuvan) sekä onnistunut laajentamaan tietokonejärjestelmien käyttöympäristöjen monimuotoisuutta. Voitaisiin väittää, että ilman adaptiivisia käyttöliittymiä elämä älypuhelinien, tablettien ja vaikkapa tulevien lisätyn todellisuuden (augmented reality) järjestelmien kanssa olisi mahdotonta tai ainakin hyvin rajoittunutta.

Monimuotoisuutensa ansiosta adaptiiviset käyttöliittymät tarjoavat myös monipuolisen tutkimuskohteen.

Seuraavissa luvuissa pohditaan muutamia mahdollisia tulevia tutkimuskohteita adaptiivisten käyttöliittymien parista ja käydään läpi miksi kyseinen tutkimus olisi tarpeellista. Luvussa 5.1 pohditaan adaptiivisten käyttöliittymien kannattavuutta kaupallisesti järjestelmän toimittajan kannalta. Tämän jälkeen luvussa 5.2 pohditaan mahdollisia olemassa olevia universaaleja adaptiivisia käyttöliittymätekoälyjä ja ongelmia todellisen universaalien käyttöliittymätekoälyn luomisessa. Luvuissa 5.3 ja 5.4 pohditaan mahdollisia tutkimuskohteita adaptiivisten käyttöliittymien ja varsinkin tekoälyn sosiologisten ongelmien parista. Viimeiseksi luvussa 5.5 käsitellään mahdollisia tulevaisuuden teknologioita, niiden vaikutusta järjestelmien käyttöympäristöihin ja mahdollisia uusia tutkimuskohteita, joita nämä uudet teknologiat voivat avata.

5.1 Adaptiivisuuden kannattavuus kaupallisesti ja tuotannollisesti

Luvussa 3 esiteltyjen hyötyjen kannalta on selvää, että adaptiivisuudella voidaan saavuttaa suuria hyötyjä varsinkin käyttäjän kannalta, mutta myös järjestelmän tuottaja pystyy saavuttamaan hyötyjä. Todetuista eduista huolimatta on kuitenkin syytä muistaa, että adaptiivisuuden toteuttaminen järjestelmään vaatii aina ylimääräistä työtä ja resursseja, kuten lukuisissa edeltävissä tutkimuksissa on todettu (Stuerzlinger & al. 2006 & Benyon D. 1993). Eikä adaptiivisten käyttöliittymän toiminnallisuuden onnistuminen ole aina täysin varmaa niin, että se tukisi onnistuneesti käytettävyyttä vaan lopputuloksena voi joskus olla käyttöliittymä, jonka käytettävyys tai ylläpitokustannukset voivat olla staattista käyttöliittymää huonompi.

Koska adaptiivisilla käyttöliittymällä pyritään vastaamaan käyttäjien eroavaisuuksiin sekä parantamaan yleistä käyttäjäkokemusta käyttäjäryhmien välillä on luonnollista, että valtaosa älykkäiden käyttöliittymien tutkimuksesta keskittyy erilaisiin hyötyihin ja haittoihin käyttäjien näkökulmasta käsin. Tutkimukset järjestelmän toteuttajan

näkökulmasta taas ovat melko vähäiset ja esimerkiksi taloudellisen hyödyn saavuttamista luvussa 3 mainituilla tavoilla on tutkittu melko vähäisesti.

Mielenkiintoisen tutkimuskohteen tekisikin sen selvittäminen, kuinka järjestelmän toteuttava osapuoli kykenisi arvioimaan adaptiivisen käyttöliittymän kehittämisen taloudellisen kannattavuuden jo ennen varsinaista kehitysvaihetta (eli vaatimustenmäärittelyä laatiessa). Lukuisissa tutkimuksissa on otettu huomioon jo se, kuinka käyttäjä hyötyy adaptiivisuudesta ja se kuinka siitä saavutettu hyöty pystytään mittaamaan. Voidaan kuitenkin olettaa, että jossain tapauksissa tätä hyötyä ei saavuteta jokaiselle käyttäjäryhmälle siinä määrin, että ero staattisen ja adaptiivisen käyttöliittymän välillä olisi käytettävyyden kannalta olisi merkittävä. Siinä tapauksessa jää kuitenkin vielä harkittavaksi ja mitattavaksi mahdolliset saavutettavat hyödyt järjestelmän toimittajan kannalta.

Todennäköisesti näiden hyötyjen tutkiminen tai saavutettavien hyötyjen mittaaminen ei tule olemaan kovin yksinkertaista. Siinä missä käyttäjähyötyjä, käytettävyyttä ja työskentelytehoa pystyttäisiin mittaamaan mm. luomalla adaptiivisia prototyyppisiä vaatimustenmäärittelyä varten, taloudellisten hyötyjen mittaamiseen ei riitä pelkkä visio tuotteesta sen valmistumisen hetkenä. Kuten luvussa 3 mainittiin, suurimpia taloudellisia etuja adaptiivisuudesta pystytään saavuttamaan hallitsemalla järjestelmään kohdistuvia tulevia muutoksia adaptiivisuuden avulla. Eli sen sijaan, että järjestelmän käyttöliittymän toimintaan tehtäisiin täydellinen muutos, pystytään adaptiivisuudella vastaamaan tähän muutosvaatimukseen automaattisesti vähentäen järjestelmän tuottamiseen vaadittuja resursseja. Jotta tällaisiin tuleviin muutoksiin pystyttäisiin varautumaan jo vaatimuksia määriteltäessä, joudutaan pelkästään sen hetkisten vaatimusten lisäksi analysoimaan mahdollisia tulevia vaatimuksia, joita järjestelmän käyttäjillä voi tulla. Tämän lisäksi analyysiin voidaan liittää lukemattomia mahdollisia muuttujia ohjelman lähivuosien elinkaaresta, kuten käyttöympäristössä tapahtuvat muutokset tai uusien kehitteillä olevien teknologioiden vaikutukset tulevaisuudessa.

Tästä syystä adaptiivisten käyttöliittymien kannattavuus kaupallisesti voisi olla hyvin mielenkiintoinen ja tarpeellinen tutkimuksen kohde. Järjestelmään sen elinkaaren

aikana vaikuttavien tekijöiden määrä voi olla todella suuri, joten vaadittaisiin laaja-alainen tutkimus/pohdinta kaikista mahdollisista aihepiireistä, jotka vaikuttavat ohjelmiston elinkaaren aikana niin, että adaptiivisuudella pystyttäisiin vastaamaan näihin muutoksiin. Sen lisäksi olisi tarpeen tutkia voidaanko edellä mainitun analyysin tuloksista mitata adaptiivisuudella saavutettavien taloudellisten hyötyjen ja niihin vaadittavan työn suhdetta. Toisin sanoen olisi tarvetta tutkia mahdollisia mittareita, joiden avulla toteuttava organisaatio pystyisi tulkitsemaan mitkä tulevaisuuden muutokset pystytään hoitamaan adaptiivisuuden avulla kannattavasti.

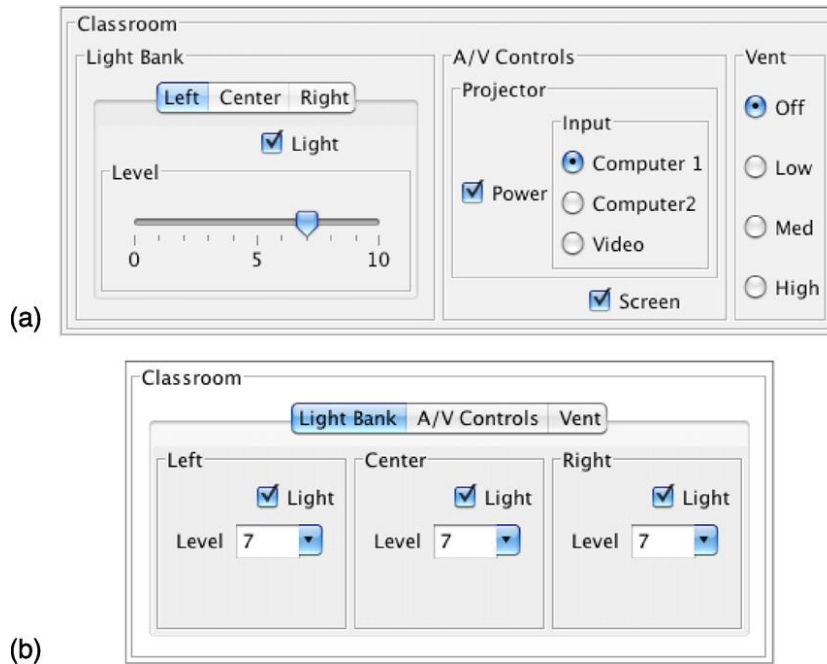
5.2 Universaali käyttöliittymätekoäly

On totta, että adaptiivisuus on hyvin monimuotoista ja sen takia pelkkä ajatus universaalista adaptiivisuudesta voi tuntua käytännössä mahdottomalta jo lukemattomien eri käyttöympäristöjen takia. Kuitenkin tarkemmin asiaan perehtyessä huomaa, että monet samoissa käyttöjärjestelmissä tai vaikkapa mobiililaitteissa toimivista ohjelmistoista omaavat paljon yhteisiä ominaisuuksia ja toimintamalleja toisten samassa käyttöympäristössä toimivien ohjelmistojen kanssa. Esimerkiksi Windows-käyttöjärjestelmän ohjelmistoissa ohjelman yläpalkeissa olevat avautuvat valikot sekä oikeassa yläkulmassa olevat napit ovat melkein poikkeuksetta samanlaisia kaikissa ohjelmistoissa järjestelmän toteuttajasta riippumatta, koska nämä ovat käyttäjille tuttuja sijainteja ja parantavat yleistä käytettävyyttä sijaitessaan käyttäjille ennestään tutuissa paikoissa.

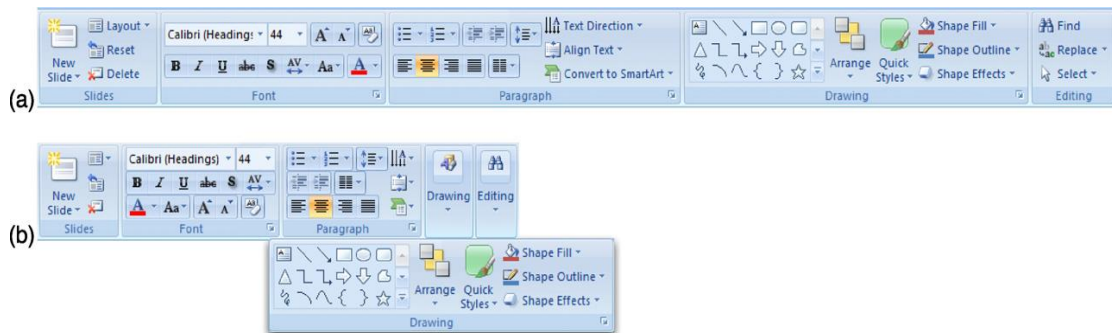
Tätä samankaltaisuutta hyväksikäyttäen on jo suunniteltu muutamia rajoitetussa käyttöympäristöjoukossa toimivia adaptiivisia tekoälyjä, kuten luvussa 4.2 esitellyt Microsoft-agentit. Agentit hyväksikäyttävät näitä tuttuja Windows-käyttöjärjestelmissä toimivien ohjelmistojen perusominaisuuksia ja niitä on nopea muokata toimimaan uusissa ohjelmistoissa käyttäjän apuna. On silti ehkä hieman liioiteltua puhua niistä universaalina käyttöliittymätekoälynä, ainakaan kovin hyvin toimivina. Myös älypuhelin ja muiden mobiililaitteiden tekoälyjä kuten Cortanaa voidaan periaatteessa mieltää tietyn käyttöjärjestelmän piirissä toimivana universaalina tekoälynä, koska niiden käyttöä ei rajoita laitteen koko, sijainti tai

käyttötarkoitus vaan ne on rajattu (tarkoituksellisesti) vain tietyn käyttöjärjestelmän omaaviin laitteisiin.

Myös Krzysztof Z. Gajos tutkijaryhmineen on perehtynyt tällaisen universaalien adaptiivisuuden toimintaan luomansa Supple-järjestelmän avulla (Gajos K. & al. 2010). Supplen tarkoitus on pystyä luomaan eri kokoisia ja eri näköisiä käyttöliittymiä, jotka soveltuvat parhaiten käyttäjän toimintamalleihin niin, että järjestelmän perustoiminnot pysyvät entisellään. Supple pystyy mm. tunnistamaan käyttäjäprofiilin perusteella yleisimmät toiminnot, joihin käyttäjä on käyttöhistoriansa aikana osoittanut kiinnostusta ja muokkaamaan näitä toimintoja säätelevät pikakuvakkeet tarkemmiksi ja helpommin saavutettaviksi. Esimerkiksi kuvassa 10 vertaillaan Supple-järjestelmän luomaa käyttöliittymää luokkahuoneen valo- ja videojärjestelmän hallinnointia varten, jossa kohta A esittää käyttöliittymää neutraalissa tilassa ilman käyttäjän historiatietoja ja kohta B esittää samaa käyttöliittymää muuntautuneena käyttäjään, jonka työskentelyhistoriasta korostuu mielenkiinto eri valojen kirkkautta kohtaan. Näin ollen Supple korostaa käyttäjälle valojen säätöpaneeleita ja sen sijaan sijoittaa käyttäjää vähemmän kiinnostaneet paneelit erillisten välilehtien taakse. Käyttäjän lisäksi Supple pystyy mukautumaan myös käyttöympäristön muutoksiin kuten toimimaan alkuperäistä kapeammissa ikkunoissa (kuva 11).



Kuva 9 - Supplen luoma käyttäjään mukautuva näkymä luokkahuoneen laitteiden hallintaan (Gajos K. & al. 2010)



Kuva 10 - Supple muokkaa MS Word ohjelman yläpalkin sopimaan kapeampaan ikkunaan (Gajos K. & al. 2010)

Gajos & al. ovat myöskin Supplen avulla saaneet aikaiseksi kiistattomia tutkimustuloksia siitä, että jopa tällainen ”yksinkertainen” adaptiivisuus, joka vaikuttaa lähinnä painikkeiden sijaintiin sekä muotoon pystyy tehostamaan käyttäjien työskentelyä erilaisista muutoksista tai rajoitteista huolimatta (Gajos K. & al. 2010).

Vaikka Supple onkin määritelmällisestä huomattavasti lähempänä aitoa universaalia adaptiivista käyttöliittymätekoälyä kuin Microsoft-agentit, sen käyttö on silti hyvin rajoitettua eri käyttöympäristöjen kuten kosketusnäyttöjen suhteen.

Olisikin mielenkiintoista tutkia, kuinka haastavaa olisi toteuttaa oikeasti universaali adaptiivinen tekoäly, joka pystyisi operoimaan toimivasti eri ohjelmistoissa ja käyttöjärjestelmästä riippumatta. Pysyttäisiinkö esimerkiksi tutkimaan käyttäjän valintoja ruudulla ja tulkitsemaan niistä eri painikkeiden sijainnit ja tulkitsemaan niistä seuraavien tapahtumien perusteella käyttäjän työskentelyn päämäärä. Yhdistämällä tähän dataan vaikkapa muilta käyttäjiltä kerättyä dataa vastaavanlaisista ohjelmista ja vastaavista toimintamalleista voitaisiin luoda käyttöjärjestelmästä riippumatta jonkinlaista adaptiivisuutta esimerkiksi älykkään käyttöliittymäassistentin muodossa.

Vaikka tällaisen universaalin adaptiivisen tekoälyn luominen voisi teoriassa olla mahdollista, joudutaan sen parissa varmasti tekemään monenlaisia kompromisseja toimintojensa sekä adaptaation syvyyden suhteen käyttöympäristöjen monimuotoisuuden takia. Pelkästään pöytätietokoneissa käytettävät käyttöjärjestelmät kuten Windows, Linux ja Mac OS eroavat perustoiminnoiltaan siinä määrin, että ilman jokaisessa käyttöliittymässä erikseen toimimaan opetettua käyttöliittymää on vaikea kuvitella toimivan kovin monimutkaisesti tai osaavan neuvoa käyttäjiä kovinkaan tarkalla tasolla, ainakaan alussa ennen kuin saadaan kasattua tarpeeksi dataa eri käyttäjiltä muodostamaan monimutkaisempia käyttäjämalleja.

5.3 Adaptiivisuuden hyödyntäminen

Myös adaptiivisuuden sosiologiset ongelmat tarjoavat paljon mielenkiintoisia tutkimuskohteita ja kysymyksiä. Kuten luvussa 3.3 mainittiin, eräs sosiologisista ongelmista, joka piinaa adaptiivisia käyttöliittymiä sekä niiden taustalla toimivia tekoälyjä on se, että tekoälyllä on tapana kadota osaksi järjestelmää eivätkä sitä käyttävät ihmiset välttämättä ymmärrä toimivansa adaptiivisen käyttöjärjestelmän

parissa. He ennemmin mieltävät hyvin toimivan käyttöliittymän osaksi järjestelmää eivätkä näin ollen välttämättä tiedosta tekoälyn läsnäoloa. Kun käyttäjät eivät tiedosta toimivansa adaptiivisessa tai kustomoitavassa ympäristössä he luonnollisesti menettävät mahdollisuuden hyödyntää adaptiivisuutta sen täydessä potentiaalissa. Tällöin käyttäjä ei pysty saavuttamaan järjestelmän tarjoamaa täyttä työskentelytehokkuutta. On myös täysin mahdollista, että osa käyttäjistä ei ole kiinnostunut käyttämään adaptiivisuutta esimerkiksi huonojen menneiden kokemusten takia tai he voivat käsittää käyttöliittymän muokkaamiseen käytetyn ajan ”hukkaan heitettyinä” aikana, jonka he ennemmin käyttävät suoraan hyödykkääseen työskentelyyn.

Todennäköisesti erot erilaisten käyttäjäryhmien kesken siinä, kuinka hyvin adaptiivisuutta käytetään hyväksi vaihtelee runsaasti esimerkiksi kokemuksen, iän tai toimenkuvan mukaan. Myös erilaiset käyttäjäkohtaiset rajoitukset voivat luultavasti vaikuttaa todella vahvasti siihen, kuinka halukas käyttäjä on oppimaan ja hyödyntämään käyttöliittymän tarjoamaa muokattavuutta. Esimerkiksi ihmiset, jotka kärsivät värisokeudesta ovat tottuneet etsimään järjestelmistä ja peleistä erilaisia asetuksia, jotka helpottavat järjestelmän käyttöä. Voidaan olettaa, että tällaisissa tapauksissa kun käyttäjä joutuu olosuhteiden pakosta käyttämään usein aikaansa käyttöliittymän muokkaamiseen, myös muiden asetusten muokkaaminen on luontevampaa ja näin adaptiivisuuden hyödyntämistä tapahtuu enemmän kuin keskivertokäyttäjillä.

Tämän aiheen parissa löytyisi vielä paljon tutkittavaa ja se varmasti tarjoaisi paljon lisätietoa järjestelmien kehittäjille siitä, kuinka paljon ihmiset keskimäärin käyttävät adaptiivisuutta hyödykseen ja kuinka tehdä siitä helpommin lähestyttävän tapauksissa, jossa käyttäjät vierastavat sitä. Erilaisia tutkimuksia siitä kuinka paljon adaptiivisuus pystyy teoriassa parantamaan työskentelyn tehokkuutta löytyy runsaasti mm. vuosittaisen älykkäiden käyttöliittymien konferenssin (Intelligent User Interfaces, IUI) materiaaleista, mutta käytännön tutkimuksia esimerkiksi siitä kuinka erilaiset käyttäjäryhmät eroavat adaptiivisuuden hyödyntämisessä tai mistä erilaiset

ennakkoasenteet adaptiivisuutta ja käyttöliittymätekoälyjä kohtaan johtuu on vielä vähänlaisesti.

5.4 Adaptiivisuuden ymmärtäminen

On selvää, että valtaosa tietokoneen käyttäjistä on tänä päivänä jonkinlaisessa vuorovaikutuksessa adaptiivisen järjestelmän kanssa, olipa kyse käyttöjärjestelmästä tai verkkokaupasta. Vaikka adaptiivisuus on osa jokaisen ihmisen arkipäivää, kuinka suuri osa lopulta ymmärtää sen, että he ovat vuorovaikutuksessa jonkin sortin tekoälyn kanssa? Ja vaikka kaikilla käyttäjillä olisi ymmärrys siitä, että järjestelmän käytöstä ohjaa tekoäly niin ymmärrys siitä mitä tekoäly on, kuinka sen kuuluisi toimia ja kuinka se toteuttaa adaptiivisuutta voivat erota huomasti eri käyttäjien välillä.

Olisikin erittäin tarpeellista selvittää kuinka erilaisten järjestelmien käyttäjät ymmärtävät adaptiivisuuden ja tekoälyn osallisuuden järjestelmän toimintaan. On todennäköistä, että esimerkiksi valtaosa mobiililaitteiden käyttäjistä ei osaa yhdistää puheen ymmärrystä adaptiiviseksi tekoälyksi ja voivat helposti lopettaa sen käyttämisen jos se ei toimi moitteettomasti muutaman ensimmäisen käyttökerran jälkeen, vaikka varsinkin näin monimutkaisten järjestelmien toimiva adaptiivisuus on usein pitkän työn takana. Samanlaisia tapauksia löytyy takuulla monista eri tyyppisistä adaptiivisista järjestelmistä.

Varsinkin verkkokaupat ja muut kaupalliset sovellukset, joista puhuttiin luvussa 4.4 voisivat hyötyä siitä, että käyttäjillä olisi parempi ymmärrys siitä kuinka heidän toimintansa vaikuttaa järjestelmän adaptiivisuuteen. Monet suosituimmista ja kaupallisesti menestyneimmistä palveluista osaavat kyllä tarjota tarkkoja suositteluja ja mahdollisesti jopa perustella miksi se ajattelee ko. suosituksen olevan käyttäjälle mieluisa, mutta todella harvat järjestelmät antavat käyttäjälle mahdollisuutta antaa palautetta suosituksen onnistuneisuudesta. Vasta viime aikoina tällaiset ominaisuudet ovat alkaneet ilmestyä mm. YouTubeen ja Googlen mainoksiin, mutta erittäin rajoittuneessa määrin.

Adaptiivisuuden sosiologiset ongelmat, kuten kuinka käyttäjät suhtautuvat tekoölyyn ja ymmärtävät sen tarjoamat mahdollisuudet tarjoavatkin laajasti potentiaalisia tutkimusaiheita. Varsinkin kun otetaan huomioon vielä viime vuosina hallitseva trendi adaptiivisuuden toteuttamisessa on ollut tekoölyn toimintalogiikan piilottaminen käyttäjältä. Tämän takia varsinaiset kaupalliset menestystarinat ns. näkyvästä adaptiivisuudesta ovat harvinaisia, vaikkakin mobiilipuolen persoonalliset tekoölyt antavatkin pientä viittausta siihen, että tämän kaltainen tekoöly on tekemässä paluuta käyttöliittymiin.

5.5 Uudet teknologiat ja adaptiivisuus

Kuten edellisessä luvussa sanottiin, mobiiliteknologiat ovat ajaneet vahvaa muutosta adaptiivisten käyttöliittymien sekä niitä ohjaavien tekoölyjen toimintaan. Tuntuukin, että adaptiivisuus on uusien teknologioiden myötä aina vain tärkeämpi osa käyttöliittymää kun laitteet muuttuvat yhä enemmän osaksi liikkuvan ihmisen elämää. Viimeisimpänä ilmiönä ja loistavana osoituksena tästä ovat esimerkiksi älykellot, joissa adaptiivisuus on todella tärkeässä roolissa erittäin rajoitetun näyttötilan ja vaihtelevien käyttöympäristöjen takia.

Älykello on kuitenkin vielä siinä mielessä perinteinen tietokonejärjestelmä, että se toimii kosketusnäytön avulla kuten normaalit mobiililaitteet. Sijaintinsa takia se voi kuitenkin hyödyntää muuta informaatiota, jota normaalit laitteet eivät voi kuten käden liikettä tai sydämen sykettä. Nämäkin uudet informaation lähteet ovat jo adaptiivisten sovellusten arkipäivää teknologian tuoreudesta huolimatta ja pystytään käyttämään hyväksi esimerkiksi lukuisissa erilaisissa terveyttä ja hyvinvointia mittaavissa sovelluksissa. Varmasti tutkittavaa tälläkin saralla riittäisi vaikkapa erilaisten tietokonejärjestelmien yhteistoiminnan kautta. Pysyttäisiinkö älykello valjastamaan osaksi pöytäkoneen käyttöä ja mittaamaan käyttäjän erilaisia tunnetiloja tai reaktioita ohjelmaan? Olisiko mahdollista käyttää älykelloa tulkitsemaan käyttäjän kädenliikkeitä ja käyttää sitä tukemaan jonkun muun mobiililaitteen käyttöä, esimerkiksi vähentämään tabletti-laitteella painettavien virhelyöntien määrää? Koska kyseessä on tuore teknologia, jonka käyttötyyli eroaa

huomattavasti määrin perinteisestä tietokoneesta sijaintinsa takia olisikin mielenkiintoista tutkia kuinka älykelloja käytetään jo hyväksi tänään ja kuinka paljon enemmän niitä voitaisiin vielä hyödyntää.

Vielä älykellojakin mielenkiintoisempi tutkimuksen kohde adaptiivisten käyttöliittymien saralla on kuitenkin vasta tuloillaan markkinoille eli erilaiset virtuaalitodellisuuden ja lisätyn todellisuuden järjestelmät. Näitä ovat esimerkiksi Oculus Rift virtuaalitodellisuuslasit tai Microsoftin hiljalleen julkistamat HoloLens lisätyn todellisuuden lasit.

Varsinkin lisätty todellisuus on adaptiivisuuden kannalta erittäin mielenkiintoinen tutkimuskohde, koska sen toiminta on pohjimmiltaan täysin adaptiivisuudesta riippuvaista. Koska käyttäjä ja käyttöympäristö ovat käytännössä osa järjestelmän toimintaa, järjestelmä joutuu sopeutumaan käyttäjän liikkeeseen, ympärillä oleviin esineisiin sekä käyttäjän komentoihin (kuva 12). Käyttöympäristöä on siis täysin mahdotonta rajata, mutta käyttöliittymän eri palaset on silti pystyttävä upottamaan osaksi käyttäjän näkökenttää luonnollisesti. Vielä jää nähtäväksi, kuinka hyvin tätä teknologiaa onnistutaan toteuttamaan käytännössä, mutta tavat hyödyntää tätä teknologiaa ovat varmasti rajattomat. Eroavaisuuksiensa takia perinteiseen tietotekniikkaan verrattuna myös tutkittavaa tällaisen adaptiivisuuden parista löytyy niin teknologiselta, sosiologiselta ja miksei jopa psykologiselta kannalta.

Adaptiivisuus ja sen eri muodot ovatkin tutkimuskohteena käytännössä katsoen eräänlainen ”uusiutuva luonnonvara”, johon uudet teknologiat synnyttävät uusia tutkimuskohteita aina luodessaan uudenlaisia tapoja käyttää tietotekniikkaa tai mahdollistavat parempia tapoja hyväksikäyttää tekoälyä nykyisissä käyttöympäristöissä.



Kuva 12 - Microsoft Hololens adaptoituu olohuoneeseen (MicHL)

Viitteet

Stuerzlinger W., Chapuis O., Phillips D., Roussel N. (2006): *User Interface Facades: Towards Fully Adaptable User Interfaces*. Comp. Science & Engineering, York University. Toronto, Canada.

Tidwel J.(2005): *Designing Interfaces*. O'Reilly Media, Inc.

Horvitz E., Breese J., Heckerman D., Hovel D., Rommelse K. (1998): The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Microsoft Research. Redmond, USA.

Lieberman H. (2009): *User Interface Goals, AI Opportunities*.Massachusetts Institute of Technology. Cambridge, USA.

Höök K. (1998): *Designing and Evaluating Intelligent User Interfaces*. Swedish Institute of Computer Science. Kista, Sweden.

Höök K. (2000): *Steps to take before intelligent user interfaces become real*. Swedish Institute of Computer Science. Kista, Sweden.

Weld D., Anderson C., Domingos P., Etzioni O. etc. (2003): *Automatically Personalizing User Interfaces*. Department of Computer Science & Engineering University of Washington. Seattle, USA.

Benyon D. (1993): *Adaptive Systems*. Computing Department, Open University, Milton Keynes.

Alvarez-Cortés V. (2007): *Current Trends in Adaptive User Interfaces*. Electronics, Robotics and Automotive Mechanics Conference. Morelos, Mexico.

Liu J., Wong C., Hui K. (2003): *Adaptive User Interfaces Based on Personalized Learning*. Hong Kong Baptist University. Hong Kong.

Horvitz E. (1999): *Principles of Mixed-Initiative User Interfaces*. Microsoft Research. Redmond, USA.

Niemi V. (2013): *Käyttöliittymän suunnittelu ja toteutus*. Turun ammattikorkeakoulu. Turku, Finland.

Browne D., Totterdell P., Normal M. (1990) *Adaptive User Interfaces*. Academic Press LTD.

Sebe N. (2009): *Multimodal interfaces: Challenges and perspectives*. Journal of Ambient Intelligence and Smart Enviroments, Volume 1, Issue 1. University of Amsterdam. Amsterdam, Netherlands.

Montague K., Hanson V., Cobley A. (2010) : *Adaptive Interfaces: A Little Learning is a Dangerous Thing*. School of Computing, University of Dundee. Dundee, UK.

Shackel B. (1990): *Human Factors and Usability*. Prentice Hall. Hemel Hempstead, UK.

Gajos K., Weld D., Wobbrock J. (2010) : *Automatically generating personalized user interfaces with Supple*. Artificial Intelligence, Volume 174 p.910-950.

Dolcourt J.: *Cortana vs. Siri vs. Google Now: An early look at how Cortana stacks up.* www.cnet.com/news/cortana-vs-siri-vs-google-now (14.4.2014)

Nelson F.: *How Microsoft Cortana Improves Upon Siri and Google Now.* www.tomshardware.com/news/microsoft-cortana-unique-features,26506.html (9.4.2014)

Apple: Macintosh Finder. The Mac 512, 9.6.2015.
www.mac512.com/macwebpages/system10.htm

Microsoft: Microsoft HoloLens. Microsoft, 9.6.2015.
www.microsoft.com/microsoft-hololens/en-us/experience

Microsoft: Microsoft Word 2010. Microsoft, 9.6.2015.

Google: Google Chrome. Google, 9.6.2015.