

**Käsinkosketeltava käyttöliittymä tuotantolinjan ohjausjärjestelmän  
testaukseen**

Antti Nyman

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Tietojenkäsittelyoppi  
Pro gradu -tutkielma  
Ohjaaja: Roope Raisamo  
Huhtikuu 2007

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Tietojenkäsittelyoppi

Antti Nyman: Käsinkosketeltava käyttöliittymä tuotantolinjan ohjausjärjestelmän testaukseen

Pro gradu -tutkielma, 95 sivua, 1 liitesivu

Huhtikuu 2007

---

Käsinkosketeltavissa käyttöliittymissä digitaalista informaatiota käsitellään aineellisten objektien välityksellä. Graafisiin käyttöliittymiin verrattuna käsinkosketeltavien käyttöliittymien on havaittu lisäävän käyttäjien suorittamien kognitiivisten toimintojen määrää. Koska monimutkaisten automaatiojärjestelmien testaus on kognitiivisesti haastavaa, pyrin tässä tutkimuksessa selvittämään, kuinka käsinkosketeltava käyttöliittymä voisi tehdä tuotantolinjan ohjausjärjestelmän testauksesta helpompaa. Kirjallisuudesta löytämieni käsinkosketeltavien käyttöliittymien ominaisuuksien, arkkitehtuuriratkaisujen ja suunnitteluperiaatteiden perusteella laadin kaksi karkeaa prototyyppiä testauksen suorittamista varten. Toisessa prototyypissä kappaleita liikutetaan käsin siirtäen ja toisessa kourua pitkin liu'uttaen. Ihmisoperaattorin ohjaamassa Ozin velho -testauksessa testihenkilöt käyttivät kumpaakin käsinkosketeltavaa käyttöliittymää sekä hiirikäyttöistä graafista käyttöliittymää yksinkertaisen testustehtävän suorittamiseen. Tehtävänä oli havaita käyttöliittymän virheelliset tilat, joissa tila erosi näytöllä olevien numerokenttien tilasta. Erot ilmenivät näytöllä sekunnin mittaisina välähdyksinä ja huomaamiseensa asti säilyvinä pysyvinä muutoksina. Tehtävien suoritusajat, havaittujen virheiden määrät ja virheiden havaitsemisajat mitattiin. Vaikka testihenkilöt pitivät käyttöliittymän tilan tarkkailua helppona käsinkosketeltavia käyttöliittymiä käyttäessään, pitivät he hiirikäyttöliittymää helpoimpana käyttää ja sen avulla he havaitsivat virheet nopeimmin. Testin tulosten perusteella näyttäisi siltä, että käsinkosketeltavien käyttöliittymien täyden potentiaalin hyödyntäminen vaatii käyttöliittymältä kehittyneitä vuorovaikutteisia ominaisuuksia, kuten monipuolisia ja vaativia liikkeitä tai ryhmätyöskentelyä.

Avainsanat ja -sanonnat: Käsinkosketeltava käyttöliittymä, fyysinen käyttöliittymä, käyttöliittymäarkkitehtuuri, ohjelmistotestaus, ihmisen ja tietokoneen vuorovaikutus.

## Sisällys

1.	Johdanto.....	1
2.	Innoittavia sovelluksia.....	5
2.1.	Marble Answering Machine.....	5
2.2.	Real Reality.....	6
2.3.	SystemBlocks ja FlowBlocks.....	7
2.4.	Amphibian.....	9
2.5.	SiteView.....	9
2.6.	POUTS.....	10
2.7.	Tangible Task Modeling Demonstrator.....	11
2.8.	Yhteenveto.....	12
3.	Käsinkosketeltavan vuorovaikutuksen ominaispiirteitä.....	14
3.1.	Syöte- ja palautekanaviin liittyvät ominaisuudet.....	15
3.1.1.	Rikas kosketeltava suorakäyttöisyys.....	15
3.1.2.	Kaksikäisyys ja tilallinen limittyneisyys.....	20
3.2.	Kognitioon liittyvät ominaisuudet.....	21
3.2.1.	Tilallisuus.....	21
3.2.2.	Syötteen ja palautteen yhdentyminen.....	23
3.2.3.	Ryhmätyöskentely.....	24
3.3.	Yhteenveto.....	26
4.	Käsinkosketeltavien käyttöliittymien määritelmiä, kehyksiä ja malleja.....	27
4.1.	Mitä käsinkosketeltavat käyttöliittymät ovat?.....	28
4.1.1.	Terminologiaa.....	28
4.1.2.	Määritelmiä ja malleja.....	28
4.1.3.	Sovellusalueita.....	32
4.2.	Käsinkosketeltavien käyttöliittymien luokitteluja.....	32
4.2.1.	Terminologiaa.....	32
4.2.2.	Objektien merkityksiin perustuvat luokittelut.....	33
4.2.3.	Objektien suhteisiin perustuvat luokittelut.....	38
4.3.	Malli käsinkosketeltavien käyttöliittymien kehitykseen.....	41
4.4.	Yhteenveto.....	42
5.	Käsinkosketeltavien käyttöliittymien arkkitehtuureja.....	44
5.1.	Käsinkosketeltavien käyttöliittymien arkkitehtuuriratkaisuja.....	44
5.1.1.	DWARF.....	45
5.1.2.	metaDESK.....	47
5.1.3.	Papier-Mâché.....	49
5.1.4.	iStuff.....	50

5.1.5. Laitekomponentit .....	52
5.1.6. Shared Phidgets .....	54
5.2. Yhteenveto .....	55
6. Käsinkosketeltavien käyttöliittymien suunnittelu ja prototypointi .....	58
6.1. Suunnitteluperiaatteet .....	58
6.1.1. Tunnistus .....	59
6.1.2. Merkityksen esittäminen objektien suhteilla .....	61
6.1.3. Merkityksen esittäminen objektien ominaisuuksilla .....	64
6.1.4. Yritys ja erehtyminen .....	65
6.1.5. Ryhmätyöskentely .....	65
6.1.6. Minimalismi .....	66
6.2. Prototypointimetodit .....	67
6.2.1. Skenaariot ja videoprototyypit .....	68
6.2.2. Karkeat prototyypit .....	69
6.2.3. Täsmäprototyypit .....	70
6.3. Yhteenveto .....	72
7. Prototyypisovelluksen testaus .....	73
7.1. Sovelluksen kuvaus .....	73
7.1.1. Arkkitehtuuri .....	73
7.1.2. Vuorovaikutukset .....	74
7.2. Testin järjestelyt .....	76
7.2.1. Testihenkilöt .....	76
7.2.2. Laitteisto .....	76
7.2.3. Testin kulku .....	79
7.3. Tulokset .....	80
7.3.1. Suorituksen nopeus .....	80
7.3.2. Virheiden havaitseminen .....	81
7.3.3. Subjektiiiviset arvosanat .....	82
7.4. Pohdintaa .....	83
8. Yhteenveto .....	86
Viiteluettelo .....	87
Liitteet	

## 1. Johdanto

Kuluttajille suunnitellut tuotteet valmistetaan yhä enenevässä määrin koneistetuilla tuotantolinjoilla, joissa erilaiset robotit ja muu automaatioteknologia hoitavat tuotteiden valmistuksen ja joissa ihmisen rooliksi jää valvonta ja puuttuminen erilaisiin ongelmatilanteisiin. Koneellisesti valmistettavia tuotteita ovat esimerkiksi elektroniikka, pakattavat elintarvikkeet ja tekstiilit. Erilaisia tuotantolinjaan kuuluvia koneita taas ovat vaa'at, tuotteita pakkauksiin nostelevat robotit ja pakkaukset kääreisiin laittavat koneet. Tuotantolinjat koostuvat yleensä peräkkäisistä koneista, jotka kukin tekevät tuotteelle tietyn työvaiheen, jolloin viimeisen työvaiheen jälkeen tuote on valmis joko loppukäyttäjän käytettäväksi tai jatkokäsittelyyn muualla. Työvaiheiden välillä puolivalmiit tuotteet siirtyvät kuljettimia pitkin. Automatisoitujen tuotantolinjojen edut ovat kiistämättömät, sillä niiden avulla tuotantokapasiteettia pystytään kasvattamaan ilman henkilökustannusten merkittävää kasvua.

Eri työvaiheita suorittavissa koneissa on tavallisesti oma sisäinen ohjelmistonsa, joka ohjaa koneen toimintaa. Koska kaikki linjan koneet kuitenkin muodostavat yhtenäisen kokonaisuuden, pitää koneiden toimia hyvin myös yhdessä toistensa kanssa. Tämän takia tarvitaan järjestelmää, joka on yhteydessä yhtäaikaisesti useaan linjan koneeseen. Järjestelmän voi esimerkiksi pitää kirjaa linjalla kulkevista tuotteista sekä tuotteiden ominaisuuksista ja järjestyksestä, jotta tuotteisiin kiinnitettävien tarrojen tai muiden merkintöjen tiedot pitäisivät paikkansa. Jos yksi kone hylkää virheellisiä tuotteita, täytyy linjalla sitä myöhemmässä vaiheessa olevan pakkaukoneen tietää välistä puuttuvat tuotteet, jotta järjestys pysyy oikeana ja tarroihin tulee oikeat merkinnät. Toisaalta linjasta täytyy saada myös valvontatietoa, jotta sen toimintaa pystytään pitkällä aikavälillä seuraamaan ja kehittämään.

Tuotantolinjan virheetön toiminta on yrityksen toimialasta riippumatta edellytys yrityksen menestykselle. Toiminnalle aiheutuvat katkokset tulevat yleensä kalliiksi johtuen menetetyistä valmistuspotentiaalista, mutta myös mahdollisesti pilaantuvista tuotteista tai yrityksen asiakkailleen lupaamien toimitusehtojen pitämättömyydestä. Tämän takia on ensiarvoisen tärkeää testata kattavasti kaikki muutokset, jotka tuotantolinjaa ohjaavaan tai valvovaan järjestelmään tehdään.

Tuotantolinjan ohjausjärjestelmän testauksessa on kuitenkin monia vaikeuksia. Ohjausjärjestelmän olisi hyvä toimia heti sillä hetkellä, kun kaikki siihen kuuluvat koneet on saatu kytkettyä paikoilleen. Tällöin ei kuitenkaan ole vielä ollut aikaa testata järjestelmää kokonaisuutena. Vaikka kukin kone siis hoitaisikin oman tehtävänsä moitteetta, ei yhteistoiminnan laita välttämättä ole samalla tavalla. Usein linjan koneet on suunniteltu niin, että aikaisemmat vaiheet eivät vaikuta niiden toimintaan; ne saavat tietystä valmistusvaiheesta olevan tuotteen syötteenä ja muuntavat sen halutunlaiseksi kehittyneemmäksi tuotteeksi, joka sitten jatkaa matkaansa. Muista koneista johtuviin

häiriöihin kone voi varautua esimerkiksi ruuhkakannoilla, joiden avulla se voi hallita liian nopealla tahdilla tulevaa syötejonoa, tai muilla tunnistimilla, joiden avulla se voi poistaa virheelliset tuotteet jo ennen niiden saapumista koneelle. Tämän takia koneen toiminta voidaan melko suurelta osin kehittää ja testata vaikka linja ei kokonaisuudessa olisikaan vielä toiminnassa. Linjan kokonaistoimintaa valvovan järjestelmän kannalta asiat ovat kuitenkin toisin. Koska kaikki vaiheet ovat sidoksissa toisiinsa esimerkiksi saman tietokannan kautta, on järjestelmän testaus mielekkäintä silloin, kun linjan kaikki koneet ovat toiminnassa samanaikaisesti. Testaukselle ei normaalisti kuitenkaan enää tässä vaiheessa jää siihen vaadittavaa aikaa tuotantokapasiteetin nopean käyttöönototarpeen takia. Johtuen edellä mainituista seikoista, järjestelmän testaus on siis mahdollista täydessä laajuudessaan vain tuotantolinjan luona, vaikka järjestelmä muuten kehitettäisiinkin täysin muualla.

Koska testaus sijoittuu vasta myöhäiseen vaiheeseen järjestelmän toteutuksessa, tulevat siinä havaittavat suuren luokan virheet kalliiksi, sillä tällöin joudutaan muuttamaan lähes valmista järjestelmää. Mikäli virheet havaittaisiin aikaisemmin – ehkä jopa suunnitteluvaiheessa – olisi niiden korjaaminen helpompaa ja vähemmän altista uusien virheiden mukaantulolle. Monimutkaisten automaatiojärjestelmien testaus on myös kognitiivisesti haastavaa. Jos testaus tapahtuu simuloimalla koneilta tulevia syötteitä testaajan itse aiheuttamalla syötteillä, tapahtuu usein niin, että vain linjan optimaalinen toiminta tulee testattua. Tähän on syynä se, että monimutkaisten tapahtumasarjojen syöttäminen on työlästä ja järjestelmän tilan oikeellisuuden seuraaminen vaatii tarkkaa havainnointia. Normaaleja ohjelmistojen testauskeinoja voidaan käyttää aikaisemmassa vaiheessa ohjelmiston kehitystä, mutta ne eivät sovellu hyvin järjestelmän käyttöönototestaukseen. Koska koneet ovat kalliita ja linjoilla olevat konekokoonpanot yksilöllisiä, ei ole mahdollista, että ohjausjärjestelmää kehittäväällä organisaatiolla olisi käytössään oikeaa tilannetta vastaava linja. Koska testauksen suunnittelu olisi hyvä aloittaa jo järjestelmän suunnitteluvaiheessa, tukisi testausta varten rakennettava systeemi myös suunnitteluvaihetta.

Käsinkosketeltavilla käyttöliittymillä tarkoitetaan käyttöliittymiä, jotka mahdollistavat virtuaalisen informaation käsittelemisen fyysisen esityksen kautta [Ishii and Ullmer, 1997]. Ne koostuvat usein kappaleista, joita käyttäjä voi liikuttaa niiden fyysisten rajoitteiden puitteissa [Ullmer *et al.*, 2005]. Käsinkosketeltavia käyttöliittymiä on kehitetty esimerkiksi opettamaan lapsille systeemidynamiikkaa [Zuckerman and Resnick, 2003; Zuckerman and Resnick, 2004] ja helpottamaan työvaiheperustaisten prosessien suunnittelua työympäristössä [Oppl, 2006; Oppl *et al.*, 2006]. Myös automaatiojärjestelmien mallintamiseen on kehitetty käsinkosketeltava sovellus [Schäfer *et al.*, 1997], mutta niiden testaukseen ei käsinkosketeltavia käyttöliittymiä ole aikaisemmin käytetty. Käsinkosketeltaviin käyttöliittymiin usein liitettävät ominaisuudet – kuten tuki tilalliselle hahmottamiselle [Sharlin *et al.*, 2004],

kognitiivisesti monimutkaisille operaatioille [Fjeld *et al.*, 2002] ja ryhmätyöskentelylle [Hornecker and Buur, 2006] – kuitenkin vihjaavat siihen suuntaan, että käsinkosketeltava käyttöliittymä voisi olla hyödyllinen apuväline automaatiojärjestelmien suunnittelussa, simuloinnissa ja testauksessa.

Tutkimusongelmana tässä tutkimuksessa on selvittää, miten voitaisiin kehittää sellainen käsinkosketeltava käyttöliittymä, joka mahdollisimman hyvin avustaisi tuotantolinjan ohjausjärjestelmän testauksessa. Tutkimusongelma jakaantuu seuraaviin osaongelmiin: mitkä käsinkosketeltavan käyttöliittymän ominaisuudet helpottavat testausta, millaisen arkkitehtuurin järjestelmä tarvitsisi, millainen järjestelmän fyysisen rakenteen tulisi olla ja miten käyttäjien tulisi olla vuorovaikutuksessa järjestelmän kanssa.

Tutkimuksen alkuosan lähestymistavassa on piirteitä teoreettis-käsitteellisestä tutkimuksesta [Järvinen ja Järvinen, 2004, ss. 17–24]. Siinä pyrin johtamaan mallin, joka kuvaa käsinkosketeltavan käyttöliittymän rakennetta kehittämisenäkökulmasta ja tutkimaan, mitkä käsinkosketeltavien käyttöliittymien ominaisuudet ovat hyödyksi testausjärjestelmässä. Tutkimusraportin alkuosa käsittää luvut 2–6. Luvussa 2 esittelen sovelluksia, jotka ovat osaltaan innoittaneet testausjärjestelmän suunnittelussa ja joiden piirteitä olisi mahdollista soveltaa kehitettävässä järjestelmässä. Samalla tutustutan lukijan käsinkosketeltaviin käyttöliittymiin ennen niitä koskevien ominaisuuksien, mallien ja kehysten esittelyä. Luku 3 käsittelee sitten käsinkosketeltavien käyttöliittymien tärkeimpiä ominaispiirteitä ja luvussa 4 käyn läpi käsinkosketeltavia käyttöliittymiä kuvaavia malleja, joista esittelen tärkeimmät. Esittelen myös oman mallini, joka kuvaa erityisesti käsinkosketeltavaa käyttöliittymää suunnittelun tuloksena syntyvänä konkreettisena kokonaisuutena. Luvussa 5 esittelen arkkitehtuuriratkaisuja, joita on käytetty käsinkosketeltavien käyttöliittymien kehityksessä ja etsin niistä suunnittelumalleja, joita voisin käyttää testauskäyttöliittymässä. Luku 6 käsittelee käsinkosketeltavien käyttöliittymien prototyyppointia ja suunnittelua.

Kokonaisuudessaan tutkimus voidaan luokitella innovaation toteuttamista kuvaavaksi tutkimukseksi [Järvinen ja Järvinen, 2004, ss. 103–117] eli sen tarkoituksena on tutkia, onko mahdollista toteuttaa haluttu järjestelmä ja miten toteuttaa se niin, että siitä tulee mahdollisimman hyödyllinen. Toteutettavana järjestelmänä on tässä tapauksessa linjanohjausjärjestelmän testaukseen tarkoitettu käsinkosketeltava käyttöliittymä. Käytettävälle lähestymistavalla on tyypillistä se, että siinä pyritään hyödyntämään perustutkimuksen tuloksia [Järvinen ja Järvinen, 2004, ss. 103]. Niinpä tämänkin tutkimuksen jälkimmäisessä vaiheessa sovelletaan tutkimuksen alkuosassa esitettyä tietämystä. Samalla kuitenkin luodaan uutta tietämystä soveltamisongelman ratkaisemiseen. Tiedonkeruumenetelminä loppuosassa käytetään kokeiluja [Järvinen ja Järvinen, 2004, ss. 160-161], fyysisiä luomuksia [Järvinen ja Järvinen, 2004, ss. 154-156] ja havainnointia [Järvinen ja Järvinen, 2004, ss. 154-156]. Tutkimuksen loppuosaa

koostuu kahdesta luvusta. Luvussa 7 esittelen kehitettävän järjestelmän, toteutuneen suunnitteluprosessin ja arvioin suunnitteluprosessin tuloksia suorittamalla testauksen. Vertaan myös testauksen tuloksia aiempiin tutkimustuloksiin ja esittelemiini käsinkosketeltavien käyttöliittymien hyviin ominaisuuksiin. Luvussa 8 teen lopuksi yhteenvedon.



## 2. Innoittavia sovelluksia

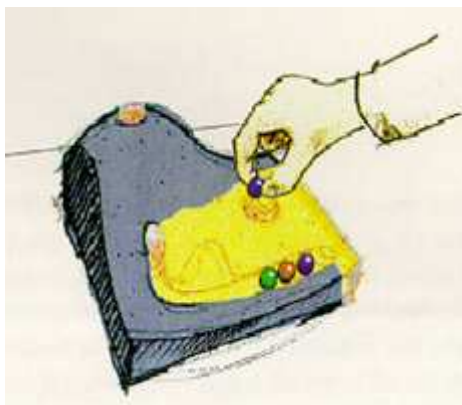
Tässä luvussa esittelen muutamia käsinkosketeltavia käyttöliittymiä, jotta lukijat saavat käsityksen siitä, mitä ne voivat käytännössä olla. Olen valinnut esittelyyn sellaisia sovelluksia, jotka ovat innoittaneet luvussa 7 esittelemäni linjanohjausjärjestelmän testikäyttöliittymän syntyä.

Vaikka käsinkosketeltavia käyttöliittymiä ei olekaan aiemmin käytetty automaatiojärjestelmien testaukseen, samankaltaisia sovelluksia on käytetty opettamaan lapsille systeemi-dynamiikkaa. Lisäksi aivan muihin tarkoituksiin suunnitellut käsinkosketeltavat käyttöliittymät sisältävät sellaisia piirteitä, joita voitaisiin hyvin käyttää myös uudenaikaisessa simulointisovelluksessa.

Systeemi-dynamiikkaa opettavat sovellukset koostuvat useimmiten erilaisista rakennuspalikoista, joita yhdistelemällä ja joiden yhteistoimintaa kokeilemalla lapset voivat rakentaa yksinkertaisia dynaamisia järjestelmiä. Lapsille on aikaisemminkin ollut käytössään erilaisia rakennussarjoja, joiden avulla abstraktit käsitteet kuten geometria ovat tulleet tutuksi. Uusien elektronisten rakennussarjojen avulla on kuitenkin mahdollista mallintaa myös alati muuttuvia järjestelmiä, jotka voivat simuloida esimerkiksi takaisinkytkentää.

### 2.1. Marble Answering Machine

Käsinkosketeltavista käyttöliittymistä kenties useimmiten käytetty esimerkki on Durrel Bishopin Marble Answering Machine [Poynor, 1995], joka mahdollistaa puhelinvastaajaviestien käsinkosketeltavan käsittelyn. Siinä pyöreät marmorikuulat edustavat puhelinvastaajaviestejä. Saapuneet viestit kerääntyvät niille tarkoitettuun kouruun (Kuva 1) ja ne voidaan kuunnella asettamalla kuula vastaanottajassa olevaan koloon. Viestin jättäneelle henkilölle voidaan soittaa asettamalla kuula puhelimessa olevaan koloon tai kuula voidaan jättää odottamaan jotakin toista henkilöä hänen omien viestien astiaansa.



---

Kuva 1. Durrel Bishopin Marble Answering Machine [Poynor, 1995].

Perimmäinen ajatus Bishopin puhelinvastaajassa ei ollut niinkään ehdottaa uutta, erityisen toimivaa lähestymistapaa puhelinvastaajien suunnittelijoilla, vaan osoittaa, kuinka tietokoneiden suorittama laskenta voidaan siirtää työpöytien ulkopuolelle antamalla ihmisten hyödyntää luonnollisia taitojaan käsitellä aineellisia objekteja [Poynor, 1995].

## 2.2. Real Reality

Schäfer ja muut [1997] ovat kehittäneet Real Reality -mallinnusjärjestelmän, joka käyttää hyväkseen sekä käsinkosketeltavia objekteja että niiden perusteella päivitettävää virtuaalista mallia. Sovelluksen ajatuksena on, että kummankin vaihtoehdon hyvät puolet saadaan yhdistettyä järjestelmäksi, jossa voidaan hyödyntää sekä vuorovaikutusta kosketeltavien objektien kanssa että tietokoneistettuja ominaisuuksia.

Järjestelmän käyttäjällä on kädessään datahansikas (data glove), joka konfiguroinnin jälkeen tunnistaa sekä asentonsa että sijaintinsa. Kun käyttäjä siirtää objekteja suunnittelupinnalle, muodostuu vastaava malli näytölle ja objektien kulkemat reitit jäävät järjestelmään talteen. Valmis malli ja sen luomiseksi suoritetut toimenpiteet voidaan tallentaa tulevia tarpeita varten.

Järjestelmälle on toteutettu sovellus, jossa voidaan simuloida kuljetuslinjoja (Kuva 2). Käyttäjä toimii simulaatiossa robottina, joka siirtelee pakkauksia kuljettimelta toiselle. Pakkausten kulkema reitti tallennetaan ja se voidaan esittää animoituna. Siitä saadaan myös jalostettua lähtökohta robottia ohjaavalle ohjelmalle.



Kuva 2. Real Reality -järjestelmä liukuhihnojen mallintamiseen ja sääntöjen luomiseen [Schäfer *et al.*, 1997].

Toisen sovelluksen avulla voidaan suorittaa monimutkaista optimointia kuljetinjärjestelmille. Tavoitteena voi olla esimerkiksi minimoida koneiden seisonta-aika tai tuotteiden läpimenoaika. Tämänkaltaisia tehtäviä voidaan ratkoa kuljettimia, koneita ja tuotteita simuloivien aineellisten kappaleiden avulla. Kappaleista voidaan rakentaa oikeata järjestelmää vastaavia pienoismalleja ja luoda sääntöjä simulointia varten (Kuva 2). Pakkauksia linjalla siirtelemällä voidaan esimerkiksi luoda sääntö,

joka määrää, että yhden koneen ollessa käytössä pakkaus menee linjan haaraumakohdassa toiselle koneelle.

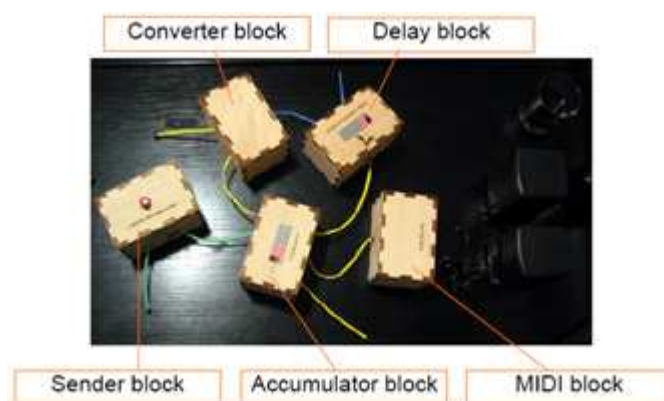
### 2.3. SystemBlocks ja FlowBlocks

SystemBlocks on kokoelma elektronisia palikoita, jotka on kehitetty opettamaan lapsille dynaamisten järjestelmien alkeita. Niiden avulla lapset voivat oppia monimutkaisten järjestelmien toimintaa sekä syy- ja seuraussuhteita järjestelmien rakenteissa.

Alkuperäinen SystemBlocks-sarja (Kuva 3) sisälsi kuuden tyyppisiä palikoita [Zuckerman and Resnick, 2003]:

- Lähettäjä (sender) syöttää käskystä numeron yksi tulostuskanavaansa.
- Kerryttäjällä (accumulator) on plus- ja miinussyötekanavat, joista saamansa arvot se lisää aikaisempaan arvoonsa. Sitten se lähettää arvon eteenpäin.
- Viivytyin (delay) hidastelee käsketyin ajan ja antaa saamansa arvon eteenpäin.
- Kertoja (multiplier) kertoo saamansa luvun käsketyllä vakiolla ja antaa tuloksen eteenpäin.
- Muunnin (converter) jättää huomioimatta saamansa arvon ja antaa aina luvun yksi eteenpäin.
- MIDI-palikka soittaa saamaansa lukua vastaavan sävelen kaiuttimestaan.

Kukin palikka voidaan yhdistää useaan muuhun palikkaan tai takaisin itseensä, jolloin yksinkertaisista ominaisuuksista saadaan muodostettua monimutkaisia järjestelmiä.



Kuva 3. Ensimmäinen SystemBlocks-rakennussarja [Zuckerman and Resnick, 2003].

Uudemmassa SystemBlocks-versiossa [Zuckerman and Resnick, 2003] palikoihin voidaan liittää kuvakortteja sitomaan palikat tosimaailmaan (Kuva 4) ja äänen lisäksi käytetään LED-valoja ilmaisemaan järjestelmän tilaa. Palikkatyyppejä on uudemmassa versiossa vain neljä. Ne on nimetty kuvaavammin, mutta niiden toimintaperiaatteet ovat pääasiassa samat kuin aiemmassakin versiossa [Zuckerman *et al.*, 2005].

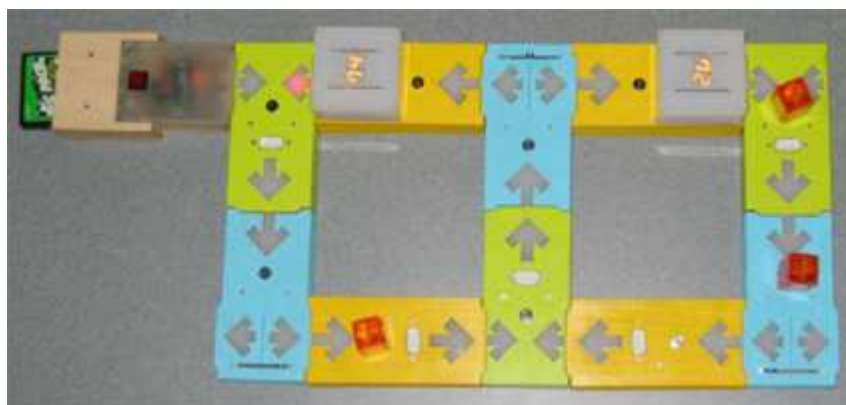


---

Kuva 4. Uudistettu SystemBlocks-rakennussarja [Zuckerman and Resnick, 2004].

FlowBlocks-palikat [Zuckerman *et al.*, 2005] (Kuva 5) muodostavat samankaltaisen rakennussarjan kuin SystemBlocks, mutta niiden tarkoitus on opettaa laskentaa, todennäköisyyksiä sekä toisto- ja haarautumisrakenteita. FlowBlocks-sarjassa on neljän tyyppisiä palikoita:

- Tuottaja (generator) antaa virtaa muille palikoille ja syöttää käskystä ensimmäisen valon, jolloin prosessi lähtee liikkeelle.
- Polku (path) yhdistää palikat toisiinsa ja näyttää prosessin etenemisen valoilla. Polku voi myös haarautua, jolloin suunta määräytyy halutun todennäköisyyden mukaisesti.
- Polkuihin voidaan kiinnittää sääntöjä (rules), jotka muuttavat isäntäpalikkansa käyttäytymistä esimerkiksi nopeuttamalla tai hidastamalla valon etenemistä.
- Anturi (probe) voi esimerkiksi mitata, kuinka monta kertaa valo on sen kohdalta kulkenut ja esittää sen näytöllään.



---

Kuva 5. FlowBlocks-rakennussarja [Zuckerman *et al.*, 2005].

Vaikka esitellyt järjestelmät on suunniteltu lapsille ja näin ollen niillä suoritettut käyttäjätestaukset ja evaluoinnit eivät välttämättä päde aikuisiin, on niissä paljon piirteitä, jotka tekisivät niistä käyttökelpoisia myös aikuisille. Mitä yksinkertaisemmista

palasista jokin järjestelmä pystytään toteuttamaan, sitä helpompaa on myös aikuisten ymmärtää sen toiminta.

#### 2.4. Amphibian

Käsinkosketeltavissa käyttöliittymissä on käytetty monenlaisia teknologioita aineellisten objektien tunnistamiseen (esimerkiksi RFID-tageja ja viivakoodeja). Amphibian-ohjelmiston [Carvey *et al.*, 2006] tarkoituksena on esitellä halpa ja yleisesti käytettävissä oleva tekniikka tunnistuksen tekemiseen. Ohjelmisto keskustelee USB-porttiin kiinnitettävän elektronisen vaa'an (Kuva 6) kanssa ja toteuttaa objekteihin liitettyjä käskyjä.

Käyttäjän tulee ensin asettaa jokin aineellinen objekti vaa'alle, jonka jälkeen hänen tulee valita graafista käyttöliittymää käyttäen tiedosto tai makro, johon kyseinen objekti liitetään. Kun sama objekti myöhemmin asetetaan vaa'alle, suoritetaan siihen liittyvä tiedosto.



Kuva 6. Vaaka Amphibian-järjestelmässä [Carvey *et al.*, 2006].

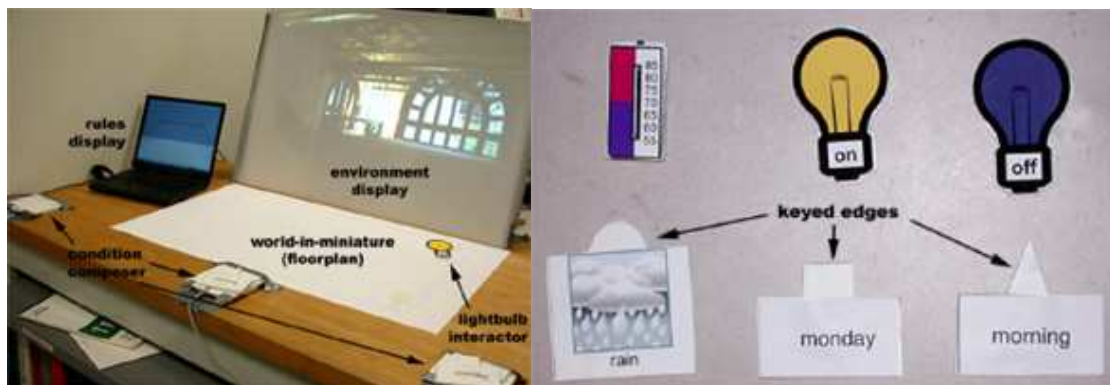
Vaikka Amphibianin yhteydessä tutkittiinkin henkilökohtaisten objektien käyttämistä käyttöliittymän osana, voi samaa tunnistustapaa käyttää muissa yhteyksissä. Tutkijat havaitsivat, että paino tunnistustapana on käyttökelpoinen ja että oletettu ongelma samanpainoisista objekteista on helposti korjattavissa, kunhan järjestelmä vain osaa huomauttaa tilanteesta.

#### 2.5. SiteView

SiteView [Beckmann and Dey, 2003] on sovellus, jolla voidaan hallita kodin automaatiota. Sen tarkoitus on tehdä automaation ohjelmoinnista niin helppoa, että loppukäyttäjät pystyvät hallitsemaan sen itse. Sovelluksen käyttöliittymään kuuluvia aineellisia ikoneita käyttämällä muodostetaan sääntöjä, joilla kontrolloidaan koneiden toimintaa. Käyttäjä käsittelee ikoneita pienoismaailmassa (joka kuvaa oikeaa maailmaa) ja muodostaa haluamiaan sääntöjä loogisia operaatioita käyttäen. Järjestelmä antaa käyttäjälle välittömästi palautetta osoittamalla, mitkä ikonit kulloinkin ovat

käytettävissä, ja näyttämällä tietokoneen ruudulla, miltä oikean järjestelmän tila muutosten jälkeen näyttäisi. Järjestelmässä on viisi osaa (Kuva 7):

- Ikoneista (interactors) muodostetaan sääntöjä ja ne tunnistetaan RFID-tagien avulla.
- Miniatyyrimalli (world-in-miniature) kuvastaa oikean järjestelmän tilaa ja oikeita tilallisia suhteita.
- Sääntöjenmuokkausalue (condition composer) tunnistaa ikonit ja muodostaa säännöt. Alue on toteutettu kolmella RFID-lukijalla.
- Näyttö (environment display) esittää järjestelmän tilan sääntöjen toteuduttua.
- Sääntönäyttö (rules display) näyttää editoitavan säännön sekä järjestelmässä jo ennestään olevat säännöt.



Kuva 7. SiteView-järjestelmä ja siihen kuuluvia ikoneja [Beckmann and Dey, 2003].

Järjestelmää kehittäessään Beckmann ja Dey erottelivat ikonit kahteen luokkaan:

- Tiukasti yhdistetyillä (tightly-coupled) ikoneilla on vastine todellisessa maailmassa. Tällainen on esimerkiksi ikoni, joka esittää laitetta ja vastaa oikeaa laitetta.
- Löyhästi yhdistetyt (loosely-coupled) ikonit ovat abstraktimpia ja niillä ei ole suoranaista vastinetta reaali maailmassa. Tällaisia ovat esimerkiksi viikonpäivät tai säätilat.

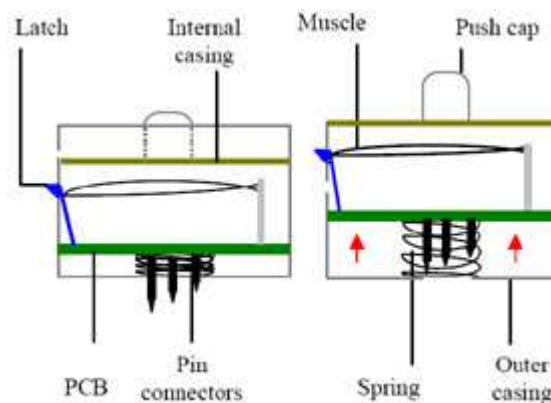
Järjestelmä ei tue monimutkaisten sääntöjen luomista, sillä RFID-lukijoiden määrä rajoittaa yhdessä säännössä käytettävien ikoneiden määrän kolmeen.

## 2.6. POUTS

Vaikka käsinkosketeltavat käyttöliittymät mahdollistavat järjestelmän tilan ohjaamisen aineellisia objekteja käsittelemällä, niin lähes poikkeuksetta niistä kuitenkin puuttuu käsinkosketeltava palaute. Tämä johtuu siitä, että aineellisia objekteja on hyvin vaikea muokata tai liikuttaa tietokoneohjatusti. Olisi kuitenkin oleellista, että

käsinkosketeltavat käyttöliittymät pystyisivät muuttamaan tilaansa, mikäli mallissa tapahtuu muutoksia, jotka sitä vaativat.

POUTit (vrt. PIN) [Ng *et al.*, 2005] ovat nastoja, jotka pystytään irrottamaan ilmoitustaulusta tietokoneohjatusti. Ne voidaan kiinnittää aivan kuten tavallisetkin nastat, mutta irrottaa ilman käyttäjän toimintaa. POUTit sisältävät tarvittavan tekniikan (Kuva 8) irrottamaan itsensä taulusta, mutta ottavat virtansa taulusta ja kommunikoivat sen kautta. POUTeilla on kaksi tilaa: ne voivat olla joko tilassa, jossa ne voidaan kiinnittää tauluun tai tilassa, jossa ne irtoavat taulusta. Kun POUTille annetaan irrotuskäsky, kulkee sähkövirta läpi POUTin sisällä olevan vaijerin. Vaijeri lyhenee sähkövirran vaikutuksesta ja avaa säpen, joka päästää irrotusjousen laukeamaan.



Kuva 8. POUTin rakenne [Ng *et al.*, 2005].

## 2.7. Tangible Task Modeling Demonstrator

Oppl ja muut [2006] ovat kehittämässä käsinkosketeltavaa käyttöliittymää, joka tukee työvaiheperustaisten järjestelmien suunnittelua. He perustelevat järjestelmän tarpeellisuutta sillä, että se tukee suunnittelussa mukana olevien ihmisten kommunikointia sekä parantaa ymmärtämystä eri kiinnostusryhmiin (kuten asiakas, ohjelmistokehittäjä tai loppukäyttäjä) kuuluvien ihmisten välillä.

Työvaiheiden suunnittelu alkaa tyhjästä pöydästä, jolle suunnitteluryhmä alkaa lisätä nimeämiänsä tehtäviä, rooleja, tavoitteita, dataa ja niiden välisiä yhteyksiä (Kuva 9). Eri elementtejä voidaan lisätä myös sisäkkäin (Kuva 9). Järjestelmä tarkkailee koko ajan taustalla ja täydentää vastaavaa virtuaalista mallia (Kuva 9), joka voidaan tallentaa tulevia tarpeita varten ja jonka kehityshistoriaa voidaan tutkia. Jotkin toiminnot, kuten esimerkiksi data-artefaktin sisällyttäminen johonkin työvaiheeseen, vaativat artefaktin tunnistamista järjestelmään. Käyttäjät voivat kehittää uusia symboleita, jotka liittyvät kiinteästi kulloiseenkin kontekstiin.





Kuva 9. Tangible Task Modeling -järjestelmä [Oppl *et al.*, 2006; Oppl, 2006].

Oppl ja muut ovat tunnistaneet kaksi vaihtoehtoa käsinkosketeltavien käyttöliittymien fyysiselle rakenteelle. Älykkäät objektit (smart things) sisältävät itsessään kaiken tarvitsemansa teknologian, kun taas älykkäissä ympäristöissä (smart spaces) teknologia sijaitsee muualla ympäristössä kuin objekteissa. Oppl ja muut ovat valinneet toteutustavaksi näiden yhdistelmän, sillä se mahdollistaa enemmän vaihtoehtoja kehittyneistä teknologioista.

## 2.8. Yhteenveto

Tämän luvun myötä lukijoille on toivottavasti syntynyt yleiskuva siitä, minkälaisia järjestelmiä käsinkosketeltavilla käyttöliittymillä yleisesti tarkoitetaan. Samalla olen esitellyt esimerkkijärjestelmissä olevia piirteitä, jotka ovat vaikuttaneet luvussa 7 esiteltävän linjanohjauksen testausjärjestelmän suunnitteluun.

Bishopin puhelinvastaajassa olennaista on koko uudenlainen lähestymistapa. Kun verrataan vastaajaviestien käsittelyä kosketeltavien kuulien ja perinteisessä vastaajassa olevien painikkeiden avulla, on helppo huomata, mikä potentiaali käsinkosketeltavilla käyttöliittymillä hyvin toteutettuna on. Real Reality -järjestelmä tuotantolinjan simuloimiseen mielestäni osoittaa, kuinka hyvin aineelliset objektit sovellusalueelle sopivat ja kuinka raaka-aineiden ja valmisteiden kulkua linjalla voidaan simuloida. SystemBlocks- ja FlowBlocks-sarjat puolestaan näyttävät, kuinka yksinkertaisista osasista saadaan rakennettua monimutkaista toiminnallisuutta sisältäviä kokonaisuuksia, joiden toiminta on silti helposti ymmärrettävissä ja muokattavissa halutunlaiseksi. Uusin SystemBlocks-versio lisäksi osoittaa halvan tavan personoida yleiskäyttöisiä palikoita liittämällä ne kuvien avulla vastaamaan jotakin reaali maailman kohdetta. Amphibian esittelee halvan ja monikäyttöisen tavan tunnistaa objekteja ilman, että niihin liitetään mitään niihin normaalisti kuulumatonta tunnistetta. Tuotantolinjoja simuloivassa järjestelmässä vaa'an käyttäminen tuntuu tämän jälkeen melko luonnolliselta.

SiteView esittelee, kuinka reaali maailman laitteita voidaan ohjata muodostamalla pienoismaailmassa esiintyvistä ikonisista objekteista sääntöjä. Samalla se esittelee hyviä



tapoja, joilla palautetta sääntöjen muodostamisesta saadaan esitettyä käyttäjälle. Aineellisten objektien tietokoneohjaus ilman kallista robotiikkaa on yhä haaste käsinkosketeltaville käyttöliittymille. POUTit osoittavat yhden tavan, jossa yksinkertaisella aineellisiin objekteihin kohdistuvalla toimenpiteellä saadaan toteutettua käyttäjälle merkityksellisiä muutoksia järjestelmän tilassa. Tangible Task Modeling Demonstrator taas osoittaa, kuinka ryhmätyönä tapahtuvaa työvaiheiden suunnittelua saadaan tuettua käsinkosketeltavuudella, silti kuitenkin menettämättä tietokoneella tapahtuvan suunnittelun hyviä puolia. Kenties jo käsinkosketeltavien käyttöliittymien suunnitteluvaihe voisi olla osa itse käyttöliittymää, jota sitten myöhemmin valmiina käytetään varsinaiseen testaustarkoitukseen.

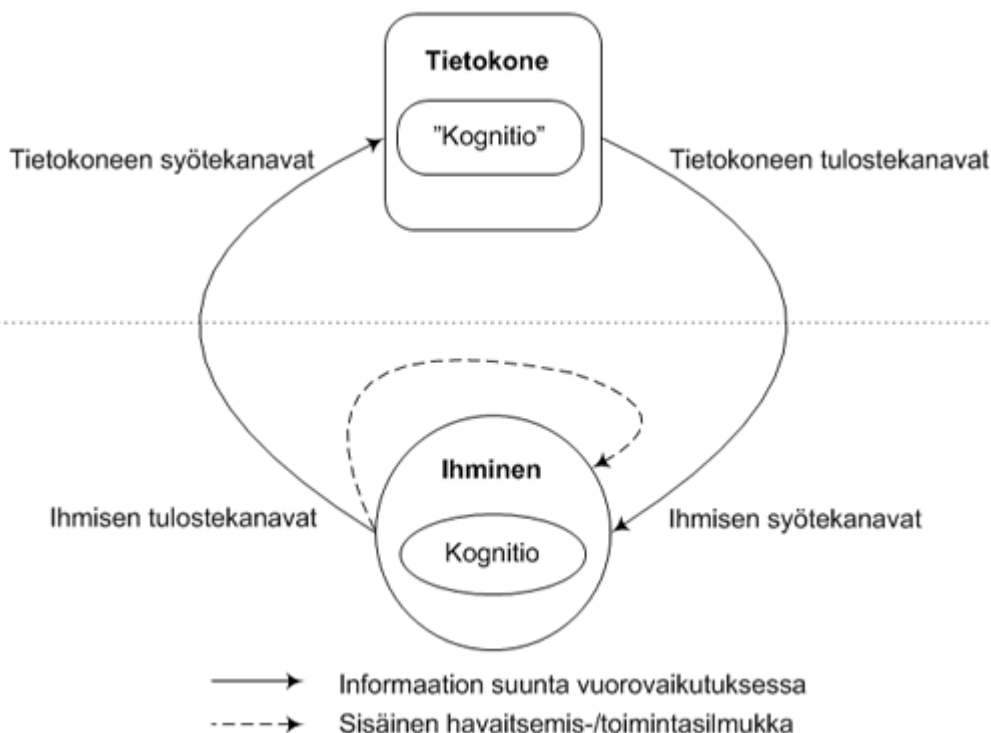
Monet tässä luvussa esitellyistä käyttöliittymistä tuntuvat tekevän järjestelmien simuloinnista jollakin tapaa intuitiivisempaa ja ehkä jopa tehokkaampaa kuin normaaleita graafisia työkaluja käyttäen. Jotta asiaa ei tarvitsisi perustella pelkällä tunteella, pureudutaan seuraavassa luvussa niihin konkreettisiin syihin, jotka tuon tunteen taustalla vaikuttavat.

### 3. Käsinkosketeltavan vuorovaikutuksen ominaispiirteitä

Käsinkosketeltavat käyttöliittymät ovat vaihtoehto ihmisen ja tietokoneen väliseksi vuorovaikutusmalliksi graafisten käyttöliittymien (graphical user interface, GUI) rinnalle. Koska kyseessä on kaksi hyvin erilaista vuorovaikutustapaa, on niiden välillä paljon eroavaisuuksia.

Tässä luvussa vertailen käsinkosketeltavan käyttöliittymän käyttämistä graafisen käyttöliittymän käyttämiseen ja esittelen tutkimuksia, joissa eroavaisuuksia on testattu. Graafisilla käyttöliittymillä tarkoitan tässä ikkuna, ikoni, valikko ja osoitin-vuorovaikutusmalliin (window, icon, menu and pointer, WIMP) perustuvia käyttöliittymiä, joissa käytetään hiirtä graafisella näytöllä olevan osoittimen ohjaamiseen. Esiteltävissä testauksissa on käsinkosketeltavia käyttöliittymiä verrattu myös muunlaisiin käyttöliittymiin.

Vertailtavat ominaisuudet olen jakanut kahteen luokkaan käyttäen apunani ihmisen ja tietokoneen vuorovaikutuksen perusmallia [MIAMI, 1995] (Kuva 10). Tässä luvussa tarkastelen ominaisuuksia vuorovaikutuksessa olevien ihmisten kannalta, joten tietokoneen syöte- ja tulostekanavia koskevat ominaisuudet käsitellään ihmisen näkökulmasta.



Kuva 10. Ihmisen ja tietokoneen vuorovaikutuksen perusmalli [MIAMI, 1995].

### 3.1. Syöte- ja palautekanaviin liittyvät ominaisuudet

Tässä kohdassa esittelen tutkimuksia, joiden koeasetelmat ja tulokset liittyvät pääasiassa ihmisten syöte- ja palautekanaviin. Ne mittaavat siis pääasiallisesti motorisen suorituksen tehokkuutta.

#### 3.1.1. Rikas kosketeltava suorakäyttöisyys

Kenties ensimmäinen ero, joka tulee mieleen ajatellessa käsinkosketeltavia käyttöliittymiä verrattuna graafisiin käyttöliittymiin, on aineellisesta objekteista saatava haptinen palaute (haptic feedback). Tämä piirre mainitaan usein puhuttaessa käsinkosketeltavista käyttöliittymistä. Vaikka haptiikkaa on tutkittu omana tutkimussuuntauksenaan paljonkin, ei tutkittua tietoa sen vaikutuksista käsinkosketeltaviin käyttöliittymiin juurikaan ole.

Haptinen palaute käsinkosketeltavissa käyttöliittymissä voidaan jakaa passiiviseen haptiseen palautteeseen ja aktiiviseen haptiseen palautteeseen. Passiivisessa haptisessa palautteessa on kyse objektien luonnostaan antamasta, yleensä taktiilista (tactile) palautteesta, jota käyttäjä voi saada esimerkiksi objektien muotoa tai pintamateriaalia tunnustelemalla. Samoin voidaan puhua myös passiivisesta kinesteettisestä (kinesthetic) palautteesta, jota saamme lihastemme liikkeistä ja asennoista niitä liikuttaessamme. Nämä palautekanavat jo yksistään mahdollistavat usein sen, että käyttäjän ei tarvitse jatkuvasti katsoa käsittelemiinsä objekteihin, vaan hän voi pitää katsettaan jossakin muualla. Passiivinen haptinen palaute on myös siinä mielessä pysyvää, että vaikka tietokone ei olisikaan päällä, ovat käsinkosketeltavaan käyttöliittymään kuuluvat aineelliset objektit yhä tunnusteltavissa ja liikuteltavissa.

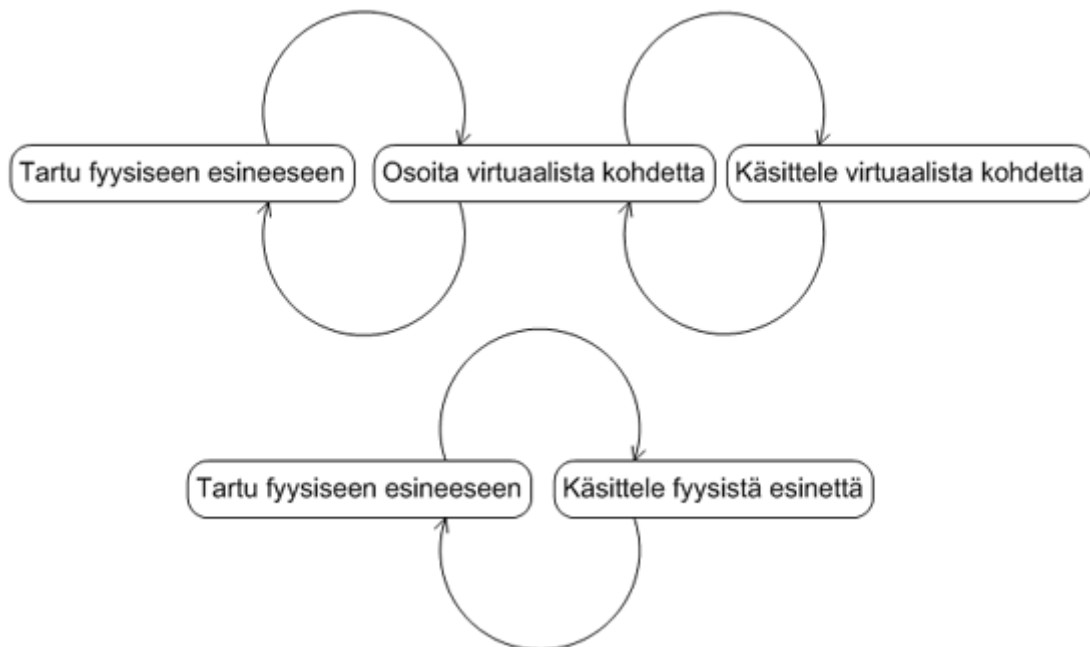
Aktiivisella haptisella palautteella tarkoitetaan sitä, että tietokone pystyy ohjaamaan sellaista palautetta, jota voidaan aistia tuntemalla. Tietokone voi esimerkiksi liikuttaa aineellisia kappaleita magneettikenttien avulla [Pangaro *et al.*, 2002], muotoilla objektien pintoja ilmalla täyttyvien taskujen avulla [Mazzone *et al.*, 2004], tulostaa kosketeltavia malleja kolmiulotteisista objekteista [Weghorst, 2003] tai muuntaa objektien muotoa ja tuntua [Michelitsch *et al.*, 2004]. Tuovatpa Poupyrev ja muut [2007] jopa esiin mahdollisuuden käsinkosketeltavien, kolmiulotteisten käyttöliittymäkontrollien dynaamisesta luomisesta muotonäyttöjen (shape display) avulla. Vaikka aktiivista haptista palautetta pystytään tällä hetkellä jo jonkin verran tuottamaan, ovat laitteet usein isokokoisia ja vielä prototyyppiasteella. Siksi laajamittainen aktiivisen haptisen palautteen käyttö käsinkosketeltavissa käyttöliittymissä antaa vielä odottaa itseään.

Käsinkosketeltavia käyttöliittymiä kuvaillaan monesti sanoilla ”intuitiivinen” tai ”luonnollinen”, jolloin niiden hyvyys katsotaan jo osoitetuksi. Nämä melko abstraktit termit eivät kuitenkaan yksinään riitä todistamaan minkään vuorovaikutustavan tehokkuutta, ellei niiden tueksi tarjota luotettavien koejärjestelyjen tuottamia tuloksia –

tai edes esitellä käytettyjen termien takana olevia syitä. Hurtienne ja Israel [2007] määrittelevät käsinkosketeltavan käyttöliittymän intuitiiviseksi, jos sen tehokas käyttö onnistuu pelkästään aikaisempaa tietämystä tiedostamattomasti hyväksikäyttäen. Tätä tutkimuksissa kuitenkin harvoin todistetaan. Usein luonnollisuuden todistamisessa vedotaan ihmisen historian saatossa oppimiin taitoihin käsitellä aineellisia objekteja ja asioita. Tästä johtuen oletetaan, että käsinkosketeltavien käyttöliittymien käyttäminen on helpommin opittavaa ja tehokkaampaa kuin graafisten käyttöliittymien käyttäminen.

Myös suoraikäyttöisyyden on pitkään tiedetty tehostavan ihmisen ja tietokoneen vuorovaikutusta. Shneiderman [1983; 1997] listaa tärkeimmiksi suoraikäyttöisyyden ominaisuuksiksi kiinnostuksenkohteiden näkyvyyden, kiinnostuksenkohteiden suoran käsittelyn fyysisillä toiminnoilla sekä komentojen nopeuden ja helpon perumisen. Hyvin suunniteltu käsinkosketeltava käyttöliittymä toteuttaa kaikki nämä ominaisuudet: normaalisti aineelliset objektit ovat koko ajan käyttäjän nähtävissä sekä käsiteltävissä ja objekteilla suoritettavat toiminnot ovat nopeita tehdä sekä helppoja perua, esimerkiksi siirtämällä objekti takaisin alkuperäiseen paikkaansa.

Käsinkosketeltavien käyttöliittymien tehokkuutta yrittää selittää myös Buxtonin [1990] esittämä malli, jolla voidaan kuvata millaisista tiloista ja siirtymistä syötelaitteen avulla tapahtuva vuorovaikutus koostuu. Myöhemmin Fitzmaurice ja Buxton [1997] johtivat eri vaihtoehdot graafisille ja käsinkosketeltaville käyttöliittymille (Kuva 11).



Kuva 11. Vuorovaikutuksen tilakaaviot graafisessa, hiirellä käytettävässä käyttöliittymässä (ylhäällä) ja käsinkosketeltavassa käyttöliittymässä (alhaalla).

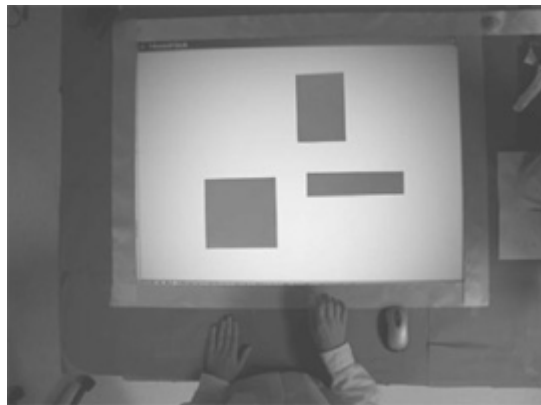
Muokattu lähteestä [Fitzmaurice and Buxton, 1997].

Graafisessa käyttöliittymässä käyttäjän on ensin tartuttava hiireen, sitten ohjattava kohdistin graafisen kohteen päälle ja lopuksi suoritettava haluamansa toimenpide, kuten

esimerkiksi kaksoisklikkaus. Käsinkosketeltavan käyttöliittymän tapauksessa tarvitaan yksi vaihe vähemmän, sillä käyttäjän tarvitsee vain tarttua haluamaansa aineelliseen ohjaimeen ja suorittaa sillä haluamansa manipulaatiot. Vaikka tämä yksinkertaistaakin vuorovaikutusta, täytyy silti huomata, että ei hiirtäkään tarvitse jokaisen muokkauksen välillä etsiä uudestaan, vaan useimmiten se on kädessä jo edellisen toimenpiteen jäljiltä. Käsinkosketeltaviin käyttöliittymiin taas kuuluu yleensä lukuisia objekteja, joista täytyy aluksi etsiä oikea.

Käsinkosketeltavien käyttöliittymien tehokkuutta on onneksi pyritty mittaamaan myös kontrolloitujen kokeiden avulla. Huolimatta siitä, että erilaisten vuorovaikutustapojen vertailu on hankalaa, voidaan ainakin suuntaa antavia tuloksia saavuttaa, mikäli testaus suoritetaan hyvin.

Huangin [2004] raportoimassa kaksiulotteisen mallin jäljennystehtävässä koehenkilöille esitettiin yksinkertaisista kuvioista koostuva malli, johon he saivat tutustua haluamansa ajan. Tämän jälkeen heitä pyydettiin toistamaan malli käyttäen joko käsinkosketeltavaa paperikäyttöliittymää (Kuva 12) tai graafista käyttöliittymää. Tehtävän suorituksesta mitattiin suoritus aika ja kappaleiden sijoitusvirheet. Koehenkilöiltä myös kysyttiin mielipiteitä kummastakin käyttötavasta.



---

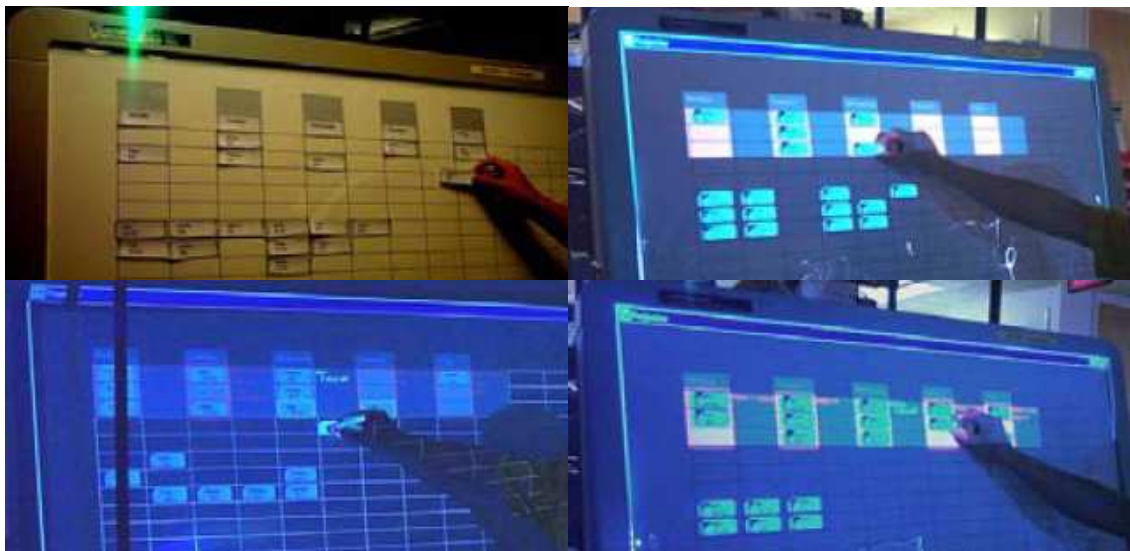
Kuva 12. Paperikäyttöliittymä Huangin kokeessa [Huang, 2004].

Testiin osallistui 12 koehenkilöä. Kokeen tulokset kertoivat, että ero koehenkilöiden suoritusajoissa eri käyttöliittymien välillä oli tilastollisesti merkitsevä. Suoritus aika oli lyhempi käsinkosketeltavaa käyttöliittymää käytettäessä. Myös käyttäjien subjektiiviset mielipiteet puolsivat käsinkosketeltavaa käyttöliittymää. Kuvioiden opettelemiseen käytetty aika ei kuitenkaan eronnut eri käyttöliittymävaihtoehtojen välillä.

Mahdolliseksi syyksi käsinkosketeltavien käyttöliittymien hyvällä menestykselle Huang [2004] ehdottaa sitä, että käyttäjien käsittelemät objektit (paperia) olivat heille ennestään tuttuja eivätkä varta vasten koetta varten suunniteltuja. Arkisten tai jokapäiväisten esineiden käyttö käsinkosketeltavissa käyttöliittymissä saattaakin edistää niiden opittavuutta ja käyttöä. Testi ei myöskään vaatinut monimutkaista päättelyä koehenkilöiltä ja näin ollen toi erot hyvin esiin.

Jacob ja muut [2002] toteuttivat kokeen, jossa koehenkilöiden tehtävänä oli järjestellä informaatiota sisältävät objektit – normaalisti esimerkiksi tarralaput – ennalta määritettyjä sääntöjä noudattavaan järjestykseen. Koeasetelmassa käytettiin Senseboard-järjestelmää, joka on pystyssä oleva 1,1m leveä ja 0,8m korkea taulu. Se on jaettu 12 sarakkeeseen ja 20 riviin. Siihen kuuluu lisäksi muovisia palikoita, jotka voidaan kiinnittää taulukon soluihin jääkaappimagneettien tavoin. Palikoiden sijainti taululla tunnistetaan tietokoneistetusti ja palikoihin voidaan heijastaa videoprojektorilla muuttuvaa informaatiota. Järjestelmä osaa myös heijastaa palikoiden välillä tapahtuvat sääntökonfliktit taululle.

Koejärjestelyssä verrattiin käsinkosketeltavia käyttöliittymiä sekä graafiseen käyttöliittymään että täysin tietokonetuettomaan käyttötapaan (Kuva 13). Koehenkilön tehtävänä oli toimia työnjohtajana ja järjestää työntekijät aikatauluun siten, että päällekkäisyyksiä ei satu ja kaikilla päivillä on tasaisesti eri taitoja osaavia tekijöitä.



Kuva 13. Eri koeasetelmat Jacobin ja muiden kokeessa: paperi (ylhällä vasemmalla), rajoitettu Senseboard (ylhällä oikealla), kynää käyttävä GUI (alhaalla vasemmalla) ja Senseboard (alhaalla oikealla) [Jacob *et al.*, 2002].

Kaikki neljä koeasetelmaa oli suunniteltu fyysisesti mahdollisimman samankaltaisiksi:

- Paperikäyttöliittymässä koehenkilöllä oli käytössään tarralappuja, jotka sisälsivät saman informaation kuin Senseboard-palikat, mutta jotka muuten olivat aivan tavallisia.
- Rajoitettu Senseboard sisälsi palikoita joihin informaatio heijastettiin, mutta ei tarjonnut tietoa rajoitusten rikkomisesta.
- Kynäkäyttöisessä graafisessa käyttöliittymässä informaatio oli heijastettu näytölle ja virtuaalisia lappuja pystyi raahaamaan kynän avulla. Myös tiedot rajoitusten rikkomisesta olivat käytössä.

- Senseboardissa käytettiin palikoita ja näytölle heijastettiin tiedot rajoitusten rikkomisesta.

Kolmesta koehenkilöä osallistui kokeeseen. Järjestelmiseen käytetty aika, palikoiden liikkeet ja palikoiden loppujärjestys mitattiin ja tallennettiin. Lopuksi koehenkilöt haastateltiin ja heiltä kysyttiin subjektiivista mielipidettä eri asetelmista. Lähes kaikki tehtävät onnistuivat, joten suoritusaikaa käytettiin mittarina. Tulokset olivat heikosti tilastollisesti merkitseviä ja Senseboard-järjestelmä osoittautui tehokkaimmaksi käyttää. Koehenkilöt myös pitivät Senseboardista eniten. Kirjoittajien mukaan ero graafiseen käyttöliittymään olisi ollut vielä suurempi, jos käytössä olisi ollut normaali, hiirtä ja näyttöä hyödyntävä asetelma. Heidän mielestään käsinkosketeltavan käyttöliittymän paremmuus johtui sen kyvystä säilyttää joitakin paperille luontaisia ominaisuuksista yhdistettynä kykyyn tuoda laskennallisia ominaisuuksia mukaan järjestelmään. Tutkimuksen johtopäätelmien perusteella voidaan siis olettaa, että käsinkosketeltavien käyttöliittymien parhaat puolet seuraavat sopivasta yhdistelmästä luonnollista vuorovaikutusta ja tietokonetuettuja ominaisuuksia.

Data Tiles -järjestelmässä [Waldner *et al.*, 2006] pöytäpinnalle heijastettuja digitaalisia piirustuksia voidaan muokata asettamalla niiden päälle läpinäkyviä litteitä palikoita. Kunkin palikan sijaintia seurataan konenäön avulla. Kun palikan havaitaan olevan kuvan päällä, liitetään kuva palikkaan, jolloin sitä voidaan muokata. Esimerkiksi kopiointitiilellä voidaan luoda kuvasta uusia ilmentymiä pöydälle ja suurennuslasitiilellä saadaan kasvatettua kuvan kokoa.

Käyttäjäkokeessa Data Tiles -järjestelmää verrattiin sekä paperikäyttöliittymään että kosketusnäyttöä käyttävään järjestelmään (Kuva 14). Kaksikymmentä henkilöä käsittäneissä testeissä käyttäjien tehtävänä oli löytää pöydällä olevista kuvista niissä olevat piilokuvat.



Kuva 14. Koeasetelmat DataTiles-testissä: kosketusnäyttö (vasemmalla), käsinkosketeltava käyttöliittymä (keskellä) ja paperikäyttöliittymä (oikealla) [Waldner *et al.*, 2006].

Käyttäjiltä kysyttiin heidän mielipiteitään eri käyttöliittymistä. Paperiversio osoittautui käyttäjien mielestä parhaaksi tähän tehtävään. Tutkijat kuitenkin huomauttavat, että erityisesti kohteiden luonnolliseen manipulointiin liittyvissä kysymyksissä käsinkosketeltavat tiilet osoittautuivat käyttäjien mielestä kosketusnäyttöä

paremmaksi vaihtoehdoksi. Kosketusnäyttöä pidettiin kuitenkin helpompana oppia. Tätä tutkijat selittivät osittain käsinkosketeltavien käyttöliittymien uutuudella sekä kyseisen järjestelmän suunnitteluongelmissa. Tässä kokeessa käsinkosketeltavien käyttöliittymien luonnollisuus havaittiin hyväksi ominaisuudeksi, jota voidaan vielä kehittää eteenpäin paremmalla suunnittelulla.

### 3.1.2. Kaksikäisyys ja tilallinen limittyneisyys

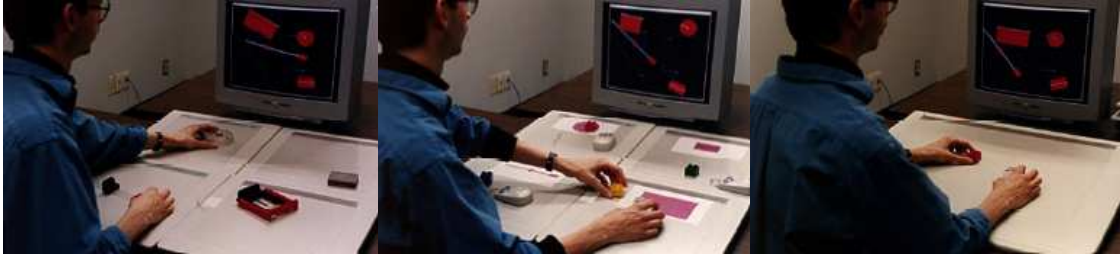
Kaksikäisen vuorovaikutuksen on havaittu olevan monissa tilanteissa yhden käden käyttöä tehokkaampaa. Buxton ja Myers [1986] esimerkiksi havaitsivat, että vuorovaikutuksen jako sellaisiin osiin, joita voidaan suorittaa samaan aikaan kahdella kädellä, lyhentää suoritusajoja. He myös huomasivat, että noviisit käyttäjät oppivat nopeasti kahden käden käyttöä. Usein käsinkosketeltavat käyttöliittymät tukevat luonnostaan kahden käden käyttämistä, mutta erityistä huomiota kannattaa silti kiinnittää siihen, että kumpaakin kättä käytetään tehokkaasti, ilman että eri käsien tehtävät sotkeentuvat.

Myös diskreetin ja jatkuvan kontrollin yhdistäminen (kts. [MacLean *et al.*, 2000]) on tekijä, joka mahdollistaa sujuvan vuorovaikutuksen käsinkosketeltavissa käyttöliittymissä. Yhdistämisellä tarkoitetaan sitä, että aineellinen objekti liitetään laskennalliseen funktioon, jonka arvoa voidaan muuttaa objektia liikuttamalla. Tässä esimerkissä objektin liittäminen funktioon on diskreetti toiminto ja objektin vapaa liikuttaminen on jatkuvaa kontrollia.

Fitzmaurice ja Buxton [1997] puolestaan esittelivät jo aikaisessa vaiheessa käsinkosketeltavien käyttöliittymien kehitystä ajatuksen tilallisesti limitetyistä syötteistä (space-multiplexed input). Siinä ideana on, että jokaisella toiminnolla on oma käsinkosketeltava ohjaimensa. Kukin ohjain on myös mahdollista suunnitella kuvastamaan sitä digitaalista funktiota, jota se vastaa. Vertailukohtana tilalliselle limittyneisyydelle on perinteinen hiiriohjaus, joka on ajallisesti limittynyttä (time-multiplexed). Siinä yhdellä ohjaimelle voidaan yhtäaikaaisesti käsitellä vain yhtä toimintoa. Toimintoja voidaan suorittaa peräkkäin, mutta ohjain ei kuvasta niitä toimintoja joita sillä ohjataan.

Fitzmaurice ja Buxton [1997] suorittivat myös kokeen, jossa he testasivat tilallisen limittyneisyyden ja kohdettaan muistuttavien ohjainten vaikutusta kohteenjäljitystehtävässä. Jäljityksessä vaadittiin kohteiden liikuttamista, kiertämistä ja skaalaamista. Tutkijat käyttivät kolmea eri koasetelmaa (Kuva 15). Ensimmäisessä asetelmassa kutakin neljästä seurattavasta kohteesta vastasi oma, kohdetta ulkoisesti muistuttava ohjain. Toisessa asetelmassa kutakin kohdetta vastasi toistensa kanssa samankaltaiset tiili-kiekko-parit, joiden avulla seuraus, kierto ja skaalaus tapahtuivat. Kolmannessa asetelmassa tiili-kiekko-pareja oli vain yksi ja se piti sitoa vuorollaan kuhunkin näytöllä olevista kohteista. Kahta kättä sai käyttää vapaasti hyväkseen.





Kuva 15. Koeasetelmat Fitzmauricen ja Buxtonin testissä: tilallisesti limitetty erikoistunein laittein (vasemmalla), tilallisesti limitetty yleisin laittein (keskellä) ja ajallisesti limitetty (oikealla) [Fitzmaurice and Buxton, 1997].

Tuloksia analysoidakseen Fitzmaurice ja Buxton [1997] laskivat koehenkilön jäljitysvirheelle arvon, joka yhdisti sijainnissa, asennossa ja koossa havaitut eroavaisuudet kaikille neljälle seurattavalle kohteelle. Tuloksista selvisi, että tilallinen limittyneisyys antoi koehenkilöille parhaat eväät suoriutua tehtävästä hyvin, sillä virheistä laskettu keskiarvo oli sille pienin. Myös kohdettaan muistuttavat ohjaimet osoittautuivat paremmiksi kuin yleispätevät. Käyttäjien mielestä tilallisesti limitetyt ja kohdettaan muistuttavat ohjaimet olivat sekä miellyttävimpiä että helpoimpia käyttää.

### 3.2. Kognitioon liittyvät ominaisuudet

Tässä kohdassa esittelemissäni tutkimuksissa ei käsinkosketeltavia käyttöliittymiä ole testattu niinkään motorisen suorituksen näkökulmasta, vaan testeissä on kiinnitetty erityistä huomiota siihen, kuinka järjestelmät tukevat käyttäjän ajatteluprosesseja. Ne siis mittaavat käyttäjän kognitiivisia prosesseja.

Eräs kognitiivisiin prosesseihin liittyvä ominaisuus on oppiminen. Huolimatta siitä, että käsinkosketeltavia käyttöliittymiä pidetään hyvin soveltuvina opetustarkoituksiin, ovat muodolliset tutkimukset käsinkosketeltavien käyttöliittymien oppimista tukevista vaikutuksista vähäisiä ja niidenkin tulokset korkeintaan suuntaa-antavia [Marshall, 2007].

#### 3.2.1. Tilallisuus

Kaikki käsinkosketeltavat käyttöliittymät ovat luonteeltaan tilallisia. Niihin kuuluvat objektit ovat kolmiulotteisia ja objekteja voidaan yleensä liikuttaa kolmiulotteisessa tilassa. Vaikka kaksiulotteisella grafiikalla pystytäänkin melko hyvin esittämään kolmiulotteisia objekteja, ovat nykyiset graafiset käyttöliittymät todellisuudessa aina kaksiulotteisia: hiiri liikkuu aina vain yhdellä tasolla, samoin kuin sen näytöllä ohjaama kursori.

Sharlinin ja muiden [2004] mukaan tilallisuus on käsinkosketeltavissa käyttöliittymissä jopa niin tärkeä ominaisuus, että se määrää koko käyttöliittymän onnistuneisuuden. Patten ja Ishii [2000] testasivat, mitä eroja on ihmisten tilankäytössä käsinkosketeltavan ja graafisen käyttöliittymän välillä. Testi perustui Kirshin [1995]

tutkimukselle siitä, kuinka ihmiset käyttävät hyväksi käsiään ja objekteja helpottaakseen omaa kognitiivista kuormaansa. Tällaisia täydentäviä strategioita (complementary strategies) voivat olla esimerkiksi objektien järjesteleminen tai muistiinpanojen tekeminen. Taustalla oli myös Kirshin ja Maglion [1994] käsite episteeminen toiminta (epistemic action). Sillä he tarkoittavat toimintaa, joka

- vähentää tarvittavan muistikapasiteetin määrää,
- vähentää tarvittavien työvaiheiden määrää, tai
- pienentää virheiden todennäköisyyttä.

Toisin sanottuna, episteeminen toiminta ei pyri suoraan ongelman ratkaisemiseen, vaan monimutkaisessa ongelmassa avustamaan etenemistä.

Käyttäjien tehtävänä Pattenin ja Ishiin [2000] testissä oli tutustua kymmeneen lehtiartikkelin tiivistelmään. Käsinkosketeltavassa käyttöliittymässä kutakin lehtiartikkelia edusti puinen palikka, kun taas graafisessa käyttöliittymässä graafinen ikoni (Kuva 16).



Kuva 16. Koeasetelmat Pattenin ja Ishiin testissä: käsinkosketeltava käyttöliittymä (vasemmalla) ja graafinen käyttöliittymä (oikealla) [Patten and Ishii, 2000].

Käsinkosketeltavassa käyttöliittymässä koehenkilöiden tuli asettaa kukin palikka vuorollaan lukijaan, jolloin he näkivät ruudulla artikkelin tiivistelmän. Graafinen käyttöliittymä toimi samalla logiikalla, mutta siinä koehenkilön täytyi raahata artikkelin ikoni lukijaikonin päälle saadakseen esiin tiivistelmän. Kummassakin tapauksessa artikkelit olivat aluksi ruudukkomaisessa muodostelmassa, mutta tiivistelmän lukemisen jälkeen koehenkilö sai asettaa ne minne halusi. Palikoiden ja ikonien ulkonäössä ei ollut keskinäisiä eroja ja muiltakin piirteiltään käyttöliittymät oli suunniteltu mahdollisimman samankaltaisiksi. Vasta lehtiartikkeleihin tutustumisen jälkeen koehenkilöille kerrottiin heidän todellinen tehtävänsä, joka oli osoittaa mikä lehtiartikkeli kuhunkin palikkaan tai ikoniin liittyy.

Testiin osallistui kolmekymmentäkuusi koehenkilöä. Testissä selvisi, että käsinkosketeltavan käyttöliittymän käyttäjät muistivat paremmin artikkeleiden paikat ja käyttivät useammin strategioita järjestelyssä. Lisäksi strategioita käyttäneet muistivat

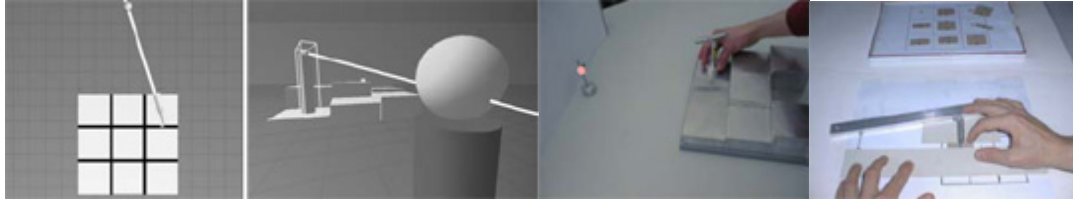
artikkelien sijainnit muita paremmin. Useimmiten käytettyjä strategioita olivat ryhmittely, järjestäminen nousevaan tai laskevaan järjestykseen ja kappaleen asettaminen johonkin merkitykselliseen paikkaan – esimerkiksi uutinen sydänvaivoista asetettiin lähelle sydäntä. Merkityksellisiin paikkoihin kappaleita sijoittaneet muistivat paremmin sijainteja kuin muut.

Patten ja Ishii [2000] selittävät tuloksia sillä, että käsinkosketeltavan käyttöliittymän palikoita on oletettavasti helpompi siirrellä kuin graafisen käyttöliittymän ikoneja. Palikoiden siirtelystä saa myös enemmän kinesteettistä palautetta. Käsinkosketeltavat käyttöliittymät näyttäisivät siis tukevan paremmin episteemistä toimintaa. Vastaavien tutkimusten tulkinnassa täytyy myös ottaa huomioon Zhangin [1997] tutkimus siitä, kuinka käytössä olevien objektien laatu voi vaikuttaa siihen, kuinka hyvin ne tukevat ongelmanratkaisua.

### **3.2.2. Syötteen ja palautteen yhdentyminen**

Olenainen ominaisuus käsinkosketeltavissa käyttöliittymissä on syötteen ja palautteen yhdentyminen. Sharlinin ja muiden [2004] mukaan käsinkosketeltavissa käyttöliittymissä voidaan syötteitä ja palautteita yhdentämällä ohittaa monia ongelmia, joita graafisissa käyttöliittymissä ilmenee. Normaalisti esimerkiksi hiiren käyttö (syöte) tapahtuu toisaalla kuin sen aiheuttamat muutokset näytöllä (palaute). Käsinkosketeltaviin käyttöliittymiin sen sijaan voidaan suunnitella toimintoja, joissa syöte ja palaute yhdentyvät. Esimerkiksi kääntäessämme kosketeltavaa vipua, tunnemme kuinka se muuttaa tilaansa ja voimme nähdä saman tapahtuvan siellä, missä kätemme toimintaa suorittaa. Syötteen ja palautteen yhdentyminen vaatii parhaalla tavalla onnistuakseen myös aktiivista palautetta käsinkosketeltavalta käyttöliittymältä (kts. [Pangaro *et al.*, 2002]).

Fjeld ja muut [2002] raportoivat käyttäjätestin, jossa he vertasivat ongelmanratkaisutehtävässä kolmea erilaista käyttöliittymää (Kuva 17), joista yksi oli käsinkosketeltava käyttöliittymä. Kokeessa oli tarkoitus mitata sitä kognitiivista tukea, jonka järjestelmä suunnittelutehtävälle antaa. Kokeessa testattu käsinkosketeltava käyttöliittymä, BUILD-IT [Fjeld *et al.*, 1998], on tarkoitettu kolmiulotteisiin suunnittelutehtäviin ja sitä käytetään pöydällä olevaa palikkaa liikuttamalla. Lisäksi pöydälle heijastetaan pohjapiirros suunnitelmasta ja vieressä olevalle kankaalle heijastetaan kolmiulotteinen sivunäkymä rakennuksesta. Vertailukohteina testissä oli kolmiulotteisista palikoista ja kaksikulotteisista pahvinpaloista koottavat mallit. Pahvin tapauksessa täytyi käsitellä pohjapiirrosta ja sivunäkymää erikseen.



Kuva 17. Koeasetelmat Fjeldin ja muiden testissä: BUILD-IT (kaksi kuvaa vasemmalla), palikkakäyttöliittymä (keskellä) ja pahvikäyttöliittymä (oikealla) [Fjeld *et al.*, 2002].

Tutkimuksessa nimitettiin vain BUILD-IT-järjestelmää käsinkosketeltavaksi käyttöliittymäksi, mutta mielestäni kaikkia testattuja vaihtoehtoja voidaan kutsua käsinkosketeltaviksi. Kolmiulotteisia palikoita käyttävä järjestelmä on Fishkinin [2004] asteikolla jopa käsinkosketeltavampi kuin BUILD-IT, sillä siinä syöte ja palaute ovat samassa paikassa.

Kolmekymmentä koehenkilöä osallistui testiin ja heiltä mitattiin tehtävän suoritus aika, suoritettujen operaatioiden määrä, oppimisen vaikutuksen edellä mainittuihin muuttujiin ja tyytyväisyys. Kolmiulotteiset palikat ja BUILD-IT mahdollistivat lyhyimmät suoritusajat ja suurimman määrän kokeiluja ja siten niiden voidaan katsoa antaneen eniten tukea kognitiiviselle toiminnalle. Käyttäjät kuitenkin pitivät kolmiulotteisia palikoita helpoimpina käyttää. Vaikka testaajat eivät tätä kommentoineetkaan, oletan tämän johtuvan juuri siitä, että syöte ja palaute olivat samassa paikassa.

### 3.2.3. Ryhmäyöskentely

Käsinkosketeltavista käyttöliittymistä sanotaan usein, että ne soveltuvat hyvin ryhmäyöskentelyyn ja useamman ihmisen käytettäväksi samanaikaisesti. Luonnollisen vuorovaikutuksen tavoin tämäkin väite jää tutkimusraporteissa usein perustelematta. Käsinkosketeltavia käyttöliittymiä ei ole juurikaan testattu ryhmäolosuhteissa. Käyttöliittymien ominaisuuksien arviointi on jo itsessään vaikeaa ja ryhmäyöskentelyn tuoma monimutkaisuus lisää entisestään kokeeseen vaikuttavien muuttujien määrää.

Maher ja Kim [2005] ovat kuitenkin tarkkailleet sarjaa suunnitteluistuntoja, joissa eri istunnoissa käytettiin erilaisia käyttöliittymiä (Kuva 18). Yhdessä koeasetelmassa normaalia näyttöä, hiirtä ja näppäimistöä käytettiin kolmiulotteiseen mallintamiseen tarkoitettua tietokoneavusteisen suunnitteluohjelman ohjaamiseen. Toinen käyttöliittymä taas oli eräänlainen käsinkosketeltava käyttöliittymä, joka koostui horisontaalisesta tasosta, jolla käsinkosketeltavia kappaleita liikuteltiin ja näytöstä, jolla mallinnuksen tulos oli reaalisajassa seurattavissa. Tilanne, jossa käyttöliittymiä käytettiin, muistutti normaalia suunnittelukatsausta, jossa ryhmä suunnittelijoita kokoontuu saman suunnitelman ympärille muokkaamaan sitä. Tutkijat tarkkailivat koehenkilöiden toimintaa ja mittasivat löytämiään koehenkilöiden suorittamia kognitiivisia toimintoja.



Kuva 18. Koeasetelmat Maherin ja Kimin testissä: käsinkosketeltava käyttöliittymä (kuvat vasemmalla) ja graafinen käyttöliittymä (kuvat oikealla) [Maher ja Kim, 2005].

Alustavassa aineiston analyysissään Maher ja Kim [2005] huomasivat joitakin seikkoja, jotka erottivat käsinkosketeltavan käyttöliittymän käytön graafisen käyttöliittymän käytöstä. Graafisen käyttöliittymän käyttäjät jakoivat ideoitaan verbaalisesti, kunnes päätyivät ratkaisuun, ja vasta tämän jälkeen suorittivat muutoksen malliin. Käsinkosketeltavan käyttöliittymän käyttäjät puolestaan käyttivät palikoita ideoidensa esittämiseen ja samalla keskustelivat päätöksistä. Tällä tavalla käsinkosketeltavan käyttöliittymän käyttäjät saivat kokeiltua useampia vaihtoehtoisia asetelmia ennen päätymistään johonkin lopputulokseen. Syiksi tähän tutkijat ehdottavat sitä, että vain yksi henkilö pääsi käsiksi hiireen kerrallaan, kun taas palikat mahdollistivat yhtäaikaisen muokkaamisen usean käyttäjän toimesta. Käsinkosketeltavien käyttöliittymien käyttäjät myös suorittivat enemmän kognitiivisia toimintoja ja tekivät enemmän yllätyksellisiä havaintoja objekteista ja niiden suhteista. Havainnoistaan tutkijat myös huomasivat, että graafisen käyttöliittymän käyttäjät keskittyivät enemmän yksittäisiin tiloihin ja objekteihin, kun taas käsinkosketeltavan käyttöliittymän käyttäjät huomioivat myös objektien ja tilojen suhteita.

Käsinkosketeltavien käyttöliittymien tukea ryhmätyöskentelylle on tutkittu myös Billinghurstin ja muiden [2002] tutkimuksessa. Siinä käytetty käyttöliittymä ei ole käsinkosketeltava käyttöliittymä puhtaimmillaan, vaan lähestymistapa on peräisin lisätyn todellisuuden (augmented reality) tutkimusalueelta. Tutkimuksessa vertailtiin kolmea erilaista käyttöliittymää (Kuva 19). Normaalissa kasvokkain tapahtuvassa vuorovaikutustavassa käyttäjät liikuttelivat pöydällä paperista valmistettuja aitoja rakennuksia muistuttavia malleja. Lisätyn todellisuuden sovelluksessa käyttäjät puolestaan liikuttelivat vastaavan kokoisia pahvinpaloja ja heillä oli päässään lasit, joiden läpi he näkivät pahvinpalojen päällä virtuaaliset rakennukset. Kolmannessa asetelmassa virtuaaliset rakennukset oli heijastettu seinällä olevalle kankaalle ja käyttäjät siirtelivät rakennuksia ohjaimella, jonka pystyi sitomaan kerrallaan vain yhteen rakennukseen. Tehtävänä kokeessa oli yhteistyötä tehden siirtää kukin yhdeksästä rakennuksesta sellaiseen asemaan, että rakennusten paikat vastasivat koehenkilöille paperilla annettuja sääntöjä.



Kuva 19. Koeasetelmat Billinghurstin ja muiden testissä: kasvokkain (vasemmalla), lisätty todellisuus (keskellä) ja heijastuskangas (oikealla) [Billinghurst *et al.*, 2002].

Itse testistä käyttäjät suoriutuivat nopeimmin käyttäen oikeita rakennusten malleja ja toiseksi nopein oli asetelma, jossa näkymä oli heijastettu seinällä. Lisätyn todellisuuden sovellusta käyttäen koehenkilöt kuitenkin tekivät ryhmätyötä paremmin kuin seinälle heijastetun näkymän tapauksessa. Oleellista tämän tutkimuksen kannalta kuitenkin on se, että käyttäjät pitivät helpompana kohteiden käsinkosketeltavaa liikuttelemista kuin erillisellä ohjaimella tapahtuvaa liikuttelua. Samoin käyttäjät pitivät hyvänä vaihtoehtoja, joissa he pystyivät olemaan katsekontaktissa ryhmänsä kanssa ilman, että heidän täytyi jatkuvasti katsoa näytölle.

### 3.3. Yhteenveto

Käyttöliittymätyyppien suuresta eroavaisuudesta johtuen myös ominaisuuksien vertailu on hankalaa. Miten esimerkiksi voidaan verrata graafisen käyttöliittymän tarjoamaa muokkauksen helppoutta käsinkosketeltavien käyttöliittymien intuitiivisuuteen? Luotettavien testausjärjestelyjen suunnittelu on hankalaa ja testien tulokset aina korkeintaankin suuntaa antavia.

Jonkin verran luotettavia testejä on kuitenkin suoritettu ja näyttäisi siltä, että ainakin tietyissä tehtävissä käsinkosketeltavat käyttöliittymät ovat helppoja oppia, miellyttäviä käyttää, säästävät kognitiivisia resursseja ja tehostavat työskentelyä. Tämä osaltaan antaa syyn jatkaa käsinkosketeltavien käyttöliittymien tutkimusta.

Käsinkosketeltavien käyttöliittymien suurimmat hyödyt ovat samalla myös niiden suurimpia heikkouksia. Erikoistuneet työvälineet soveltuvat vain yhteen tarkoitukseen ja intuitiivinen käyttö ei helposti tue eksperttien oikoteitä. Jotta vahvuudet kuitenkin päihittäisivät heikkoudet, täytyy suunnitteluvaiheessa tuntea ominaisuudet hyvin ja osata korostaa kussakin tilanteessa oikeita ominaisuuksia.

#### 4. Käsinkosketeltavien käyttöliittymien määritelmiä, kehyksiä ja malleja

Visio kaikkialla läsnäolevasta tietotekniikasta, joka katoaa taustalle käyttäjän suorittaessa jokapäiväisiä tehtäviään [Weiser, 1991], on inspiroinut laajasti tutkimusta ihmisen ja tietokoneen vuorovaikutuksessa. Jokapaikan tietotekniikassa (ubiquitous computing) tietokoneet häviävät käytöstä sellaisina kuin olemme ne aikaisemmin tunteneet, ja sulautuvat osaksi ympäristöämme. Tällöin voimme käyttää niitä työkaluina suorittaessamme jokapäiväisiä tehtäviämme, ilman että ne vaativat erityistä huomiota ja irrottavat keskittymisemme varsinaisesta tehtävästä. Samalla koko nykyinen käsityksemme tietokoneenkäyttötaidosta voidaan unohtaa. Vaikka vision tärkeimpänä viestinä pidetään monesti verkottuneiden tietokoneiden leviämistä kaikkialle, oli siinä heti alusta alkaen tärkeää myös tietokoneiden mahdollistaman laskennan yhdistyminen huomaamattomaksi osaksi jokapäiväistä aineellista ympäristöämme.

Eräs aikaisimmista ja vaikuttavimmista yrityksistä yhdistää jokapäiväisissä toimissa käyttämämme esineet tietotekniikan kanssa on DigitalDesk-ympäristö [Wellner, 1993], joka on hyvä esimerkki lisätyn todellisuuden sovelluksesta. Se mahdollistaa käyttäjälle totutun vuorovaikutuksen paperin kanssa, samalla kuitenkin helpottaen paperilla esitetyn informaation laskennallista käsittelyä. Videokameran ja -projektorin avulla toteutetulla työpöydällä käyttäjä pystyy käsittelemään tavallisia papereita ja samalla hyödyntämään digitaalisten dokumenttien hyviä puolia, kuten esimerkiksi kopiointia ja liittämistä. Jo tässä järjestelmässä käyttäjän ja tietokoneen välinen vuorovaikutus perustui aineellisiin objekteihin, vaikka siitä suuri osa tapahtuikin eleiden avulla.

Aikaisempaa kiinteämmän liitoksen aineellisten objektien ja digitaalisen informaation välille toteuttivat tartuttavat käyttöliittymät (graspable user interface) [Fitzmaurice *et al.*, 1995], jotka mahdollistivat digitaalisen datan muokkaamisen aineellisten objektien avulla. Ne toivat käyttöliittymäkomponentit näytöltä fyysisesti kosketeltaviksi. Eräässä sovelluksessa käyttäjä pystyy asettamaan tasolle heijastettujen piirustusohjelman elementtien päälle aineellisia palikoita ja näitä liikuttamalla muuttamaan graafisten elementtien ominaisuuksia, kuten esimerkiksi sijaintia, suuntausta tai kokoa.

Tartuttavat käyttöliittymät ovat vaikuttaneet vahvasti myöhempään tutkimukseen digitaalisen informaation ja aineellisten objektien yhdistämisessä. Inspiraatiota siitä on saanut myös visio käsinkosketeltavista käyttöliittymistä [Ishii and Ullmer, 1997], joka antoi nimen tutkimussuuntaukselle. Esitettyä visiota laajasti tulkiten käsinkosketeltaviksi käyttöliittymiksi voidaan käsittää kaikki käyttöliittymät, joissa käyttäjä käsittelee aineellisia objekteja muuttaakseen niihin sidottua digitaalista

informaatiota. Kuten jatkossa käy ilmi, ei täyttä yksimielisyyttä käsinkosketeltavien käyttöliittymien määritelmästä tai peruselementeistä ole kuitenkaan vielä saavutettu.

#### **4.1. Mitä käsinkosketeltavat käyttöliittymät ovat?**

Käsinkosketeltava käyttöliittymä on nuori käsite ja siksi sen merkitys on monille tuntematon. Laajan tulkinnan mukaan käsinkosketeltaviksi käyttöliittymiksi voidaan luokitella kaikki käyttöliittymät, joissa käyttäjä käsittelee aineellisia objekteja, joihin on sidottu digitaalista informaatiota. Kuten myöhemmin esiteltävistä määritelmistä kuitenkin käy ilmi, eroavia mielipiteitä käsinkosketeltavan käyttöliittymän ominaispiirteistä on olemassa. Tässä kohdassa yritän selventää käsinkosketeltaviin käyttöliittymiin liittyvää käsitteistöä sekä tärkeimpiä periaatteita ja yleisimpiä käyttötarkoituksia. Vertailen aluksi käsinkosketeltavista käyttöliittymistä käytettyjä termejä, jonka jälkeen käsittelen niille esitettyjä määritelmiä. Lopuksi esittelen käsinkosketeltavien käyttöliittymien yleisimpiä sovellusalueita.

##### **4.1.1. Terminologiaa**

Käsinkosketeltavuutta kuvaamaan on englanninkielisissä julkaisuissa käytetty vaihtelevia termejä, kuten esimerkiksi ”tangible” [Ishii and Ullmer, 1997], ”embodied” [Dourish, 2001] tai ”graspable” [Fitzmaurice *et al.*, 1995]. Vaikka eri termeillä hienoisia painotuseroja onkin, tarkoitetaan niillä pääosin samaa asiaa. Käyttämäni suomennos perustuu englanninkieliseen termiin ”tangible user interface”, jossa esiintyvä sana ”tangible” on puolestaan saanut alkunsa latinan kielestä. Latinankieliset sanat ”tangibilis” ja ”tangere” tarkoittavat suomen kielessä kutakuinkin ”jota voi koskea” ja ”koskea”.

##### **4.1.2. Määritelmiä ja malleja**

Tartuttavat käyttöliittymät määriteltiin alun perin sellaisiksi käyttöliittymiksi, joissa osa käyttöliittymän normaalisti virtuaalisista elementeistä saa aineellisen olomuodon [Fitzmaurice *et al.*, 1995]. Tämä määritelmä painottaa eroavuutta graafisiin käyttöliittymiin, sillä sen mukaan tartuttavat objektit tulevat aikaisempien syötelaiteiden korvaajiksi ja mahdollistavat joidenkin graafisen käyttöliittymän elementtien suoran käsittelemisen. Määritelmä ei kuitenkaan ota kantaa siihen, millaisia tulostuskanavia tartuttavien käyttöliittymien tulisi käyttää, ja käyttöliittymä saattaa uusista syötelaiteista huolimatta muistuttaa muilta osin graafisia käyttöliittymiä.

Suuressa osassa käsinkosketeltaviin käyttöliittymiin liittyvistä julkaisuista viitataan käsitteen määrittelyssä Ishiin ja Ullmerin [1997] Tangible Bits -visioon. Siinä on tavoitteena tehdä virtuaalisesta informaatiosta käsinkosketeltavaa ja sen taustalla vaikuttavat ihmisten luonnolliset kyvyt käsitellä esineitä ja asioita. Alun perin Ishii ja Ullmer vaativat käsinkosketeltavilta käyttöliittymiltä ainoastaan sen, että ne antavat käyttäjälle mahdollisuuden käsitellä digitaalista informaatiota tarttumalla informaatioon

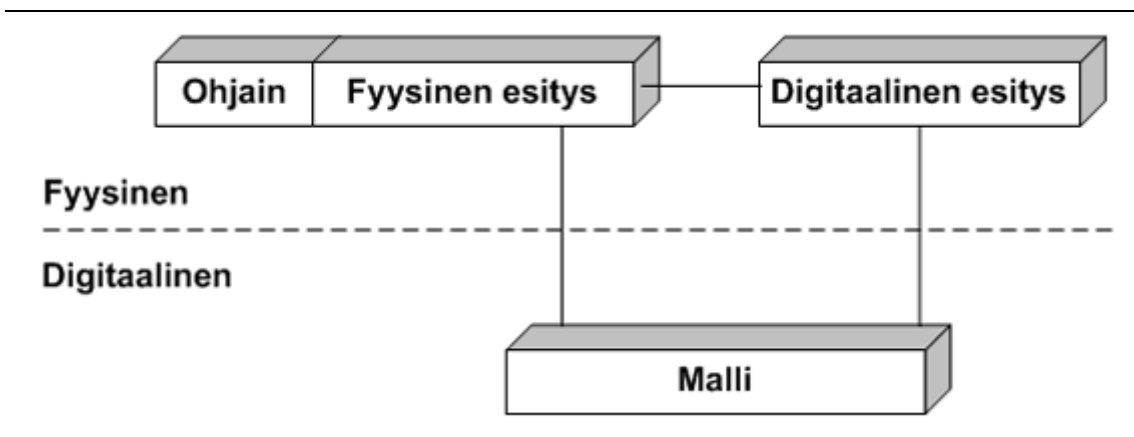


liitettyihin aineellisiin objekteihin ja käsittelemällä niitä. Myöhemmin Ullmer ja Ishii [2000] ovat lisänneet määritelmään vaatimuksen siitä, että käsinkosketeltavan käyttöliittymän ei tulisi tehdä eroa syöte- ja tulostuslaitteiden välille. Nykyisen määritelmän mukaan käsinkosketeltavissa käyttöliittymissä käytettävien aineellisten objektien tulee siis olla yhtäaikaista sekä esitys, joka osittain vastaa virtuaalista informaatiota, että ohjain, jonka avulla informaatiota voidaan muokata. Tällöin niissä yhdistyvät sekä syöte- että tulostuskanavan ominaisuudet.

Käsinkosketeltavilla käyttöliittymillä on määritelmän mukaan neljä ominaispiirrettä [Ullmer and Ishii, 2000]:

- aineellinen esitys on sidottu tietokoneen avulla digitaaliseen informaatioon,
- aineellinen esitys mahdollistaa vuorovaikutteisen kontrolloinnin,
- aineellinen esitys on sidottu havaittavasti aineettomaan esitykseen, ja
- aineellisten objektien tila ilmentää osaltaan järjestelmän virtuaalista tilaa.

Esitettyjen piirteiden perusteella käyttöliittymään kuuluviin aineellisiin objekteihin tulee olla liitettyä jotakin digitaalista informaatiota tai laskennallisia malleja, joiden tilaa järjestelmään kuuluvat aineelliset objektit osaltaan ilmentävät. Vaikka aineelliset objektit muodostavatkin osan esityksestä, liittyy siihen usein myös muita tulosteita (kuten grafiikkaa tai ääntä), joiden tulee liittyä havaittavasti aineellisiin objekteihin. Aineelliset objektit eivät myöskään ole olemassa pelkkää informaation esittämistä varten, vaan toimivat järjestelmän ensisijaisena ohjaimena.



Kuva 20. MCRpd-malli [Ullmer and Ishii, 2001].

Ullmerin ja Ishiin esittämään määritelmään liittyy olennaisesti käsinkosketeltavia käyttöliittymiä kuvaava malli (Kuva 20), joka perustuu tunnettuun graafisten sovellusten ohjelmointiin liittyvään malli-näkymä-ohjain-suunnittelumalliin (Model-View-Controller, MVC). Ullmer ja Ishii kutsuvat malliaan nimellä malli-ohjain-esitys (fyysinen ja digitaalinen) (Model-Control-Representation physical and digital, MCRpd). MCRpd-mallissa tehdään karkea jako fyysiseen ja digitaaliseen puoleen. Fyysisellä puolella ovat kaikki asiat, jotka ovat ihmisen aisteilla havaittavissa. Digitaaliselle

puolelle jää ainoastaan malli, joka ei ole suoranaisesti aistittavissa. Fyysiselle puolelle sijoittuvat sekä fyysinen että digitaalinen esitys. Fyysisellä esityksellä tarkoitetaan järjestelmään liittyviä kosketeltavissa olevia asioita. MCRpd-malli korostaa fyysisen esityksen toimimista sekä esityksenä että ohjaimena ja fyysinen esitys liittyykin suoraan järjestelmän ohjaimeen. Digitaalisia esityksiä voivat puolestaan olla näytölle piirrettävät grafiikat ja teksti sekä kaiuttimista kuuluvat äänet. Ne eivät kuitenkaan mahdollista järjestelmän ohjaamista.

Vielä Ishiin ja Ullmerin määritelmää laajemmin määritelmän käsinkosketeltaville käyttöliittymille esittää Fishkin [2004], jonka määritelmä ei ota yhtä tarkasti kantaa siihen, kuinka aineelliset objektit esittävät digitaalista informaatiota ja mahdollistavat sen kontrolloinnin. Tämä määritelmä kuitenkin kattaa myös monia sellaisia käyttöliittymiä, jotka aiemman määritelmän perusteella eivät olisi olleet luokiteltavissa käsinkosketeltaviksi. Määritelmän mukaan käsinkosketeltavan käyttöliittymän tulee toteuttaa ainoastaan seuraavat asiat [Fishkin, 2004]:

- käyttäjä saa aikaan syötetapahtuman käsittelemällä aineellisia objekteja,
- tietokonejärjestelmä havaitsee syötteen ja muuttaa tilaansa, ja
- järjestelmä laukaisee tulostustapahtuman, joka näkyy käyttäjälle jonkin aineellisen objektin ominaisuuden muutoksena (esimerkiksi objektin muoto, ääni, grafiikka näyttöpinnassa tai tuntopalaute).

Edellä mainitut piirteet vaativat järjestelmältä kaikessa yksinkertaisuudessaan vain sen, että siihen kuuluvia aineellisia objekteja käsittelemällä saadaan aikaan niihin liittyvää tietokoneellisesti ohjattua palautetta. Määritelmä on hyvin laaja ja siihen liittyykin myöhemmin esiteltävä luokittelu, jonka avulla järjestelmän käsinkosketeltavuuden taso on mahdollista määrittää.

Hornecker ja Buur [2006] esittävät mallin (Kuva 21), joka kattaa käsinkosketeltavat käyttöliittymät laajassa merkityksessä. Mallissa on tarkoitus ottaa paremmin huomioon käsinkosketeltaviin käyttöliittymiin liittyvät sosiaaliset ominaispiirteet, sekä yhdistää tietojenkäsittely-, ilmaisu- ja tilanäkökohdat. Tietojenkäsittelykeskeinen lähestymistapa tarkoittaa tässä sitä, että painopiste analysoinnissa on fyysisten esitysten ja digitaalisten mallien välisissä kytkennöissä. Ilmaisua taas korostetaan silloin, kun analyysin pääpaino on siinä, miten ihmiset liikkuvat ja toimivat ollessaan vuorovaikutuksessa käyttöliittymän kanssa. Tilakeskeisyys puolestaan painottaa interaktiivisten tilojen merkitystä ja on lähtöisin arkkitehtuurin piiristä.

Mallissa on neljä pääteemaa, jotka jakaantuvat kustakin teemasta riippuen erilaisiin alakohtiin. Teemat ovat kuitenkin osittain päällekkäisiä ja niiden alakohdilla on siten yhteneväisyyksiä.



Kuva 21. Ryhmätyöskentelykeskeinen malli käsinkosketeltaville käyttöliittymille [Hornecker and Buur, 2006].

Ensimmäinen alakohta on käsinkosketeltavuus, joka käsittää asioita, jotka Ishiin ja Ullmerin [1997] määritelmä sisällyttää käsinkosketeltaviin käyttöliittymiin. Siinä pääosassa ovat aineelliset objektit sekä käyttäjien vuorovaikutus näiden objektien kanssa. Käyttäjät voivat käsitellä objekteja suoraan käyttämättä apuvälineenä esimerkiksi hiirtä ja saavat samalla palautetta siitä, miltä käsiteltävät objektit tuntuvat. Vuorovaikutuksen tulisi olla kevyttä, jotta käyttäjät voivat kokeilla haluamiansa asioita ja saada kokeiluistaan välittömästi palautetta. Yhtenäisten vaikutusten ansiosta käyttäjien tulisi myös pystyä ymmärtämään tekemistensä ja niiden seurausten suhteet.

Tilallinen vuorovaikutus -alakohta ottaa huomioon käsinkosketeltavien käyttöliittymien tilallisen luonteen korostamalla kolmiulotteista tilaa, jossa vuorovaikutus aina tapahtuu. Tilassa olevia objekteja muokkaamalla ja järjestämällä voimme tehdä tilan merkitykselliseksi juuri itsellemme. Koko tila on tarkasteltavissamme ilman, että ruudun reunat sirpaloivat näkymää. Voimme myös käyttää koko vartaloamme vuorovaikutuksessa ja samalla esittää tai opettaa muille jotakin toimimme kautta.

Ruumiillistuneisuus tarkoittaa sitä, että todelliset rajoitteet ja käytännöt ohjaavat ja rajoittavat toimintaamme. Aineellisten objektien muodot ja koot vaikuttavat siihen, mitä niillä on mahdollista tehdä. Ne myös yleensä mahdollistavat usean käyttäjän osallistumisen yhtäaikaaisesti joko tarkkailijan tai toimijan roolissa. Järjestelmän tulisi myös räätälöityä käyttäjän kykyjen ja kehittymisen mukaan.

Esityksen ilmaisulla tarkoitetaan merkityksiä, joita käsinkosketeltavien käyttöliittymien aineelliset ja digitaaliset esitykset meille tuovat. Järjestelmän tilan olisi hyvä olla havaittavissa aineellisista objekteista ilman digitaalisten esitysten tukea. Käytämme aineellisia objekteja usein myös ajatustemme ilmaisemiseen ja muistin apuvälineenä. Kaikkien objektien ja niiden mahdollistamien toimintojen tulisi olla kytketty ymmärrettävästi niihin liittyviin digitaalisiin operaatioihin, siten että tehtyjen toimenpiteiden seuraukset olisivat loogisia.

Mallin eri kohdat sopivat paremmin tietyille sovellusalueille kuin joillekin muille. Osa kohdista esimerkiksi soveltuu parhaiten ryhmätyötä tukeviin järjestelmiin. Mallia voidaan kuitenkin käyttää kaikenlaisten käsinkosketeltavien käyttöliittymien analysointiin ja suunnitteluun sekä uusien kehityssuuntien löytämiseen.

#### 4.1.3. Sovellusalueita

Vaikka käsinkosketeltavien käyttöliittymien on todettu soveltuvan monenlaisiin tehtäviin, on olemassa tiettyjä sovellusalueita, joilla ne ovat alkaneet yleistyä muita enemmän. Vaikka kattavaa luokittelua sovellusalueista ei välttämättä ole järkevää tehdä, on yleiskuvan saamiseksi hyödyllistä määrittää yleisimpiä käsinkosketeltavien käyttöliittymien käyttötarkoituksia. Seuraavat sovellusluokat on havaittu laajoiksi ja kehittyviksi [Ullmer and Ishii, 2001]:

- informaatiota säilyttävät, hakevat ja muokkaavat,
- informaatiota visualisoivat,
- järjestelmiä mallintavat tai simuloivat,
- järjestelmiä hallinnoivat tai konfiguroivat, ja
- opetus-, viihde- tai ohjelmointisovellukset.

Sovellusalueiden sisälläkin järjestelmät voivat erota toisistaan merkittävästi ja painottua eri tavoin. Monilla edellä mainituilla sovellusalueilla käsinkosketeltavia käyttöliittymiä on toteutettu erityisesti tietokoneavusteisen ryhmätyön (computer supported collaborative work) tutkimiseksi.

## 4.2. Käsinkosketeltavien käyttöliittymien luokitteluja

Hyvät teoriat ja luokittelut ovat tarpeellisia kaikille tutkimussuuntauksille, sillä ne auttavat tutkimusalueen määrittelemisessä, helpottavat erilaisten sovellusten vertailua ja ohjaavat suunnittelua. Ne myös auttavat näkemään sovellusten hyviä puolia sekä alueita, joilla on vielä käyttämättömiä mahdollisuuksia.

Tässä kohdassa käsittelen teorioita ja luokitteluja, jotka helpottavat käsinkosketeltavien käyttöliittymien määrittelyä, ryhmittelyä, vertailua ja suunnittelua. Esittelen vaihtoehtoisia ja toisiaan täydentäviä luokitteluja sekä vertailen niitä keskenään. Tarkoitukseni on löytää luokitteluista ja teorioista eroja ja yhtymäkohtia sekä eri ratkaisuille ominaisia hyviä ja huonoja puolia. Olen ryhmitellyt luokittelut sen mukaan, perustuvatko ne aineellisten objektien merkityksiin vai niiden välisiin suhteisiin. Aloitan selvittämällä aihepiiriin liittyviä terminologisia ongelmia, minkä jälkeen esittelen eri luokittelut.

### 4.2.1. Terminologiaa

Jotta käyttöliittymään kuuluvat aineelliset objektit voidaan erottaa kaikista muista aineellisista objekteista, tarvitaan nimityksiä, joilla niihin voidaan helposti ja

yksiselitteisesti viitata. ”Aineellinen objekti” on jatkuvassa käytössä pitkä ja hankala, eikä tee eroa käyttöliittymään kuuluvien ja kuulumattomien objektien välille (sama ongelma pätee myös englanninkieliseen termiin ”physical object”). ”Aineellinen esine” on merkitykseltään samankaltainen ja siihen liittyvät samat heikkoudet. Lähes vastaava lyhempi termi on ”artefakti”, johon kuitenkin sisältyy vaatimus, että se on ihmisen tekemä. Tämä taas ei ole hyväksyttävää, sillä käsinkosketeltavissa käyttöliittymissä voidaan käyttää muitakin kuin ihmisten tekemiä objekteja [Ullmer and Ishii, 2001].

Englanninkielestä johdetun termin ”phicon” (”physical icon”) esittelivät Ishii ja Ullmer [1997], mutta ovat itsekin todenneet sen soveltumattomuuden kaikkia objekteja kuvaavaksi termiksi, sillä objektit eivät välttämättä ole merkitykseltään ikonisia, vaan voivat olla myös symbolisia [Ullmer and Ishii, 2000]. Myös termiä ”pyfo” on ehdotettu käytettäväksi samaan tarkoitukseen [Calvillo-Gómez *et al.*, 2003; Shaer *et al.*, 2004], mutta ainakaan vielä se ei ole levinnyt laajempaan käyttöön. Ishii ja Ullmer toteavat käyttökelpoiseksi muuallakin käytetyn termin ”tangible”, mutta päätyvät ehdottamaan termin ”token” käyttöä kuvaamaan käsinkosketeltaviin käyttöliittymiin kuuluvia aineellisia objekteja. Lisäksi he ehdottavat termiä ”reference frame” (suomen kielessä ”koordinaatisto”) kuvaamaan aineellista tilaa, jossa vuorovaikutus tapahtuu.

Käsitteiden eroavaisuudet aiheuttavat jonkin verran hankaluuksia jäljempänä esiteltävien teorioiden tulkinnassa ja vertailussa. Englanninkielinen termi ”token” tarkoittaa muiden merkitystensä ohella pientä osaa, joka edustaa kokonaista. Tätä painottaen Holmquist ja muut [1999] käyttävät sitä kuvaamaan aineellisia objekteja, jotka ilmentävät olemuksellaan ainakin jotakin virtuaalisen vastineensa piirrettä. He harkitsivat termin sijasta myös ”phicon”-termin käyttöä, mutta hylkäsivät sen, sillä sitä on aikaisemmissa julkaisuissa käytetty heidän käytöstään poikkeavalla tavalla. Ullmer ja Ishii [2000] puolestaan toteavat, että Holmquistin ja muiden ”token” käsitettä vastaa heidän käsitteistössään juuri ”phicon” ja heillä ”token” on yläkäsite kaikille käyttöliittymään kuuluville aineellisille objekteille.

Tässä tutkimuksessa käytän termiä ”aineellinen objekti” tarkoittamaan käyttöliittymään kuuluvia objekteja. Syynä tähän on edellä mainittu termien päällekkäisyys, joka saattaa aiheuttaa sekaannuksia. Kunkin luokittelun tai teorian yhteydessä käytetään suomennoksia siinä käytettävistä termeistä ja ne selitetään erikseen kunkin esittelyn kohdalla.

#### **4.2.2. Objektien merkityksiin perustuvat luokittelut**

Monissa tapauksissa käsinkosketeltavien käyttöliittymien analysointi on helpointa aloittaa siitä, miten niihin kuuluvia aineellisia objekteja käytetään digitaalisen informaation esittämiseen ja käsittelemiseen. Useat nykyisistä teorianluomisyrityksistä onkin tehty tällä tavalla.

Aikaisessa yrityksessään teorian luomiseksi Holmquist ja muut [1999] ehdottavat mallia, jossa käyttöliittymään kuuluvat aineelliset objektit voidaan jakaa kolmeen

luokkaan: säiliöt (container), merkit (token) ja työkalut (tool). Säiliöt ovat tässä luokittelussa yleisluontoisia aineellisia objekteja, jotka eivät ulkoisella olemuksellaan ilmennä niihin liittyvien virtuaalisten objektien luonnetta ja jotka voidaan tilanteesta riippuen liittää moneen erityyppiseen informaatioon. Säiliöitä käytetään usein informaation siirtämiseen eri kohteiden välillä ja ne ovat pääosin vain tilapäisesti liitettynä virtuaalisiin objekteihin. Merkit puolestaan muistuttavat vähintään joltakin aineelliselta ominaisuudeltaan niihin liitettyä virtuaalista objektia. Merkkien liitokset virtuaalisiin objekteihin ovat monesti luonteeltaan pysyvämpiä kuin säiliöiden vastaavat, eivätkä merkit yleensä voi edustaa kuin tietyn tyyppistä informaatiota. Työkalut toimivat edustajina laskennallisille funktioille. Ne voivat esimerkiksi toimia tartuntakahvoina virtuaalisiin objekteihin, tarjota linssin läpi vaihtoehtoisen näkymän käsiteltävään maailmaan tai muuttaa jotakin järjestelmään kuuluvaa virtuaalista ominaisuutta. Jotkut työkalut muistuttavat ominaisuuksiltaan edustamaansa laskennallista funktiota.

Käsiteltävien objektien luokittelun lisäksi Holmquist ja muut esittävät käytettäväksi termiä hana (faucet), jolla tarkoitetaan sellaista pistettä, jossa käyttäjän on mahdollista päästä käsiksi merkkeihin liitettyyn informaatioon. Hanoja voivat olla esimerkiksi näytöt, kaiuttimet tai muut vaaditun toiminnallisuuden toteuttavat tulostuslaitteet.

Termi informaation pääsy (access) puolestaan tarkoittaa merkin esittämistä hanalle, jolloin päästään käsittelemään informaatiota. Pääsyä voidaan kontrolloida merkkien saatavuutta rajoittamalla, vaatimalla usean merkin yhtäaikaista käyttöä tai asettamalla tietyt merkit toimimaan vain tietyissä paikoissa tai sijainneissa. Toinen merkkeihin liittyvä operaatio on informaation liittäminen (association), joka tarkoittaa digitaalisen informaation yhdistämistä merkkiin. Järjestelmästä riippuen liitos informaatioon voi olla muutettavissa tai kiinnitetty lopullisesti jo järjestelmän toteutusvaiheessa. Näistä vaihtoehdoista on käytetty myös nimityksiä muuttuva (dynamic) ja muuttumaton (static) liitos [Ullmer and Ishii, 2000]. Joitakin aineellisia objekteja voidaan myös ylikuormittaa (overload), jolloin yksi objekti liittyy useaan eri informaatioyksikköön. Tällöin objektin kautta päästään käsiksi tiettyyn, kontekstista riippuvaan informaatioon tai vaihtoehtoisesti kaikkeen siihen liittyvään informaatioon kerrallaan.

Taulukko 1. Objektien luokittelu Underkofflerin ja Ishiin [1999] mukaan.

<b>Puhdas objekti</b>	<b>Attribuutti</b>	<b>Substantiivi</b>	<b>Verbi</b>	<b>Uudelleenkonfiguroitava työkalu</b>
-----------------------	--------------------	---------------------	--------------	----------------------------------------

Kaikki luokittelut eivät välttämättä tähtää yleiskäyttöisten mallien luomiseen. Tämä pätee esimerkiksi seuraavan luokittelun kohdalla, joka edellisen luokittelun lailla koskee yksittäisiä objekteja, mutta vain erääseen tiettyyn sovellustyyppiin liittyen [Underkoffler and Ishii, 1999]. Objektien luokat on sijoitettu akselille sen perusteella,

kuinka tarkasti objektit järjestelmässä vastaavat sitä objektia, jota ne tosielämässä esittävät (Taulukko 1). Puhtaat objektit (object as pure object) ovat järjestelmän kannalta merkityksellisiä vain objekteina. Ne voivat olla joko yksilöityjä tai yksilöimättömiä, mutta niiden ominaisuuksilla ei ole merkitystä. Attribuuttiobjekteissa (object as attribute) järjestelmä välittää vain yhden niiden ominaisuuden arvosta, kuten esimerkiksi muodosta tai väristä. Substantiiviobjektit (object as noun) edustavat järjestelmässä koko merkitystään ja toimivat pääosin samalla tavalla kuin vastaavat objektit aineellisessa maailmassa. Verbiobjektit (object as verb) taas ovat substantiiviobjekteja abstraktimpia eivätkä esiinny järjestelmässä itsenäin, vaan vaikuttavat substantiiviobjekteihin tai koko ympäristöön. Uudelleenkonfiguroitavat työkalut (object as reconfigurable tool) toimivat eri tavoin riippuen kontekstista ja tilasta ja ne voidaan liittää tilanteesta riippuen eri objekteihin.

Taulukko 2. Objektien luokittelu Kolevan ja muiden [2003] mukaan.

<b>Yleiskäyttöinen työkalu</b>	<b>Erikois-työkalu</b>	<b>Tunniste</b>	<b>Edustaja</b>	<b>Heijastuma</b>	<b>Sama objekti</b>
--------------------------------	------------------------	-----------------	-----------------	-------------------	---------------------

Koleva ja muut [2003] ehdottavat käsinkosketeltavien käyttöliittymien luokittelua sen perusteella, kuinka yhtenäisesti aineelliset ja virtuaaliset objektit on liitetty toisiinsa (degree of coherence) (Taulukko 2). Samalla he esittelevät luokittelun ympärille rakentuvan teorian ja siihen liittyvät käsitteet.

Luokittelussa heikoin liitos on yleiskäyttöisillä työkaluilla (general purpose tool), jotka käyttäjä voi liittää mihin tahansa virtuaaliseen objektiin. Vahvimmillaan liitos taas on silloin, kun toisiinsa liitetyt aineellinen ja virtuaalinen objekti hahmotetaan samaksi objektiksi. Näin voi tapahtua esimerkiksi sovelluksissa, joissa objekti voi siirtyä aineellisesta todellisuudesta virtuaaliseen, ollessaan kerrallaan läsnä vain toisessa. Näitä kahta edellä mainittua luokkaa Koleva ja muut eivät laske käsinkosketeltaviksi käyttöliittymiksi, toisin kuin kolme niiden välissä olevaa luokkaa.

Ensimmäinen varsinainen käsinkosketeltavien käyttöliittymien luokka on erikoistyökalut (specialized tool), jolla tarkoitetaan tehtävältään yleiskäyttöisiä työkaluja tarkemmin rajoittuneita, mutta silti moniin eri objekteihin liittyviä työkaluja. Hieman vahvempi liitos on tunnisteilla (identifier), jotka vastaavat jotakin virtuaalista objektia ja tarjoavat pääsyn siihen. Edustajat (proxy) puolestaan ovat vielä vahvemmin ja pysyvämmiin liitettyjä virtuaalisiin objekteihin ja mahdollistavat niiden monipuolisemman käsittelyn. Viimeinen käsinkosketeltaviin käyttöliittymiin kuuluva luokka on heijastumat (projection), joihin liittyvät virtuaaliset objektit ovat suoria vastineita aineellisille objekteille ja riippuvaisia näiden olemassaolosta.

Koleva ja muut [2003] ovat myös jakaneet liitosten luonteen ominaisuuksiksi, joiden avulla yhtenäisyyttä voidaan arvioida. Ensimmäinen ominaisuus on muunnos (transformation), joka voi saada arvokseen joko kirjaimellinen (literal) tai muunnettu

(transformed), riippuen siitä, saako aineelliseen objektiin kohdistuva toiminta saman muutoksen aikaan vastaavassa virtuaalisessa objektissa vai onko näiden kahden välillä eroa. Toinen ominaisuus on vuorovaikutuksen havaitseminen (sensing of interaction), jonka avulla määritetään, mitkä aineelliseen objektiin kohdistuvat toiminnot ovat järjestelmän kannalta merkityksellisiä. Näitä voivat olla esimerkiksi pyörittäminen, liikuttaminen tai asettaminen. Muunnoksen konfiguroitavuus (configurability of transformation) puolestaan tarkoittaa sitä, voidaanko objektien välissä olevaa muunnosta muuttaa (configurable) vai onko se kiinteä (fixed). Liitoksen elinikä (lifetime of link) määrittää onko objektien välinen liitos väliaikainen (temporary) vai pysyvä (permanent). Itsenäisyys (autonomy) kuvaa missä määrin virtuaalinen ja aineellinen objekti ovat riippuvaisia toisistaan. Itsenäinen (autonomous) objekti on olemassa toisesta objektista riippumatta, mutta riippuvaiset (dependent) objektit luodaan vasta toisten objektien myötä ja tuhoataan niiden hävittyä. Liitosten lukumäärä (cardinality of link) kuvaa moneenko virtuaaliseen objektiin aineellinen objekti on yhtäaikaaisesti liitetty. Liitoksen lähde (link source) taas tarkoittaa onko käyttäjälle palautetta antava lähde aineellinen vai virtuaalinen.

Taulukko 3. Lisäys objektien luokitteluun van den Hovenin ja Eggenin [2004] mukaan.

	Pysyvä		Joustava	
	Symbolinen	Ikoninen	Symbolinen	Ikoninen
<b>Yleinen</b>				
<b>Henkilökohtainen</b>				

Eri luokittelut voivat myös laajentaa toisiaan. Van den Hoven ja Eggen [2004] ehdottavat laajennusta Ullmerin ja Ishiin [2000] luokitteluun, jotta se tukisi paremmin henkilökohtaisten esineiden käyttöä käyttöliittymän objekteina (Taulukko 3). Ensimmäisen luokitteluperusteen mukaan objekti on joko henkilökohtainen (personal), jolloin objektin käsittelijä on yleensä myös omistaja, tai yleinen, jolloin objektilla ei ole henkilökohtaista merkitystä käyttäjälle. Toisaalta jaotteluun vaikuttaa myös se, onko objektin ja informaation välillä oleva liitos pysyvä (fixed) vai joustava (flexible). Nämä vastaavat Ullmerin ja Ishiin [2000] muuttuvaa ja muuttumatonta liitosta. Kolmantena jaotteluna merkit voidaan vielä jakaa symbolisiin ja ikonisiin niiden semantiikan perusteella.

Koska yhtä ainoaa käsinkosketeltavuuden astetta määrittävää ominaisuutta ei ole löytynyt, voidaan luokitteluongelmaa yrittää ratkaista moniulotteisen asteikon avulla. Mitä useampi ulottuvuus asteikolla on, sitä tarkempi kuvaavuus saavutetaan, mutta sitä monimutkaisemmaksi luokittelu käy. Siksi kaksiulotteinen jaottelu saattaa hyvinkin olla sopiva kompromissi. Fishkinin [2004] kaksiulotteisessa luokittelussa (Taulukko 4) asteikkoina toimivat ruumiillistuvuus (embodiment) ja vertauskuvallisuus (metaphor).



Tässä lähestymistavassa käyttöliittymän käsinkosketeltavuuden taso voidaan määrittää sen perusteella, kuinka tiukasti käyttöliittymä noudattaa jotakin metaforaa ja kuinka tiukasti sen antama palaute on sidottu kulloinkin käsiteltävänä olevaan objektiin. Mitä enemmän nämä kaksi ominaisuutta käyttöliittymässä toteutuvat, sitä käsinkosketeltavampana käyttöliittymää voidaan pitää.

Taulukko 4. Järjestelmien luokittelu Fishkinin [2004] mukaan.

<b>Vertaus- kuvallisuus</b>	<b>Olematon</b>	<b>Substantiivi</b>	<b>Verbi</b>	<b>Substantiivi ja verbi</b>	<b>Täysi</b>
<b>Ruumiillistuvuus</b>					
<b>Täydellinen</b>					
<b>Läheinen</b>					
<b>Ympäristöllinen</b>					
<b>Kaukainen</b>					

Ruumiillistuvuus tarkoittaa sitä, kuinka tiukasti syöte ja tuloste on sidottu toisiinsa paikan suhteen. Tämä voidaan ymmärtää niin, että mikäli syöte ja tuloste ovat täysin toisiinsa sidottuja, käsittää käyttäjä käsittelemänsä objektin kokonaisuudeksi, joka sisältää kaiken tarvitsemansa toiminnallisuuden. Tässä luokittelussa ruumiillistuvuus voi olla täydellistä (full), jolloin syöteobjekti on sama kuin tulostusobjekti, läheistä (nearby), jolloin tuloste ilmenee syöteobjektin lähellä, ympäristöllistä (environmental), jolloin tuloste ilmenee käyttäjän ympäristössä tai kaukaista (distant), jolloin tuloste on käyttäjältä etäällä, esimerkiksi jollakin kaukaisella näytöllä.

Vertauskuvallisuus tarkoittaa käyttöliittymäobjektin toiminnan samankaltaisuutta reaali maailman kanssa. Vertauskuvallisuus voi olla olematonta (none), mutta jos sitä on, se voi vastata substantiivia (noun), verbiä (verb), substantiivia ja verbiä (noun and verb) tai olla täyttä (full), jolloin vastaavuus on niin ilmeistä, että mitään vertauksia ei tarvitse tehdä. Substantiivi vertauskuvana tarkoittaa sitä, että objekti näyttää esikuvaltaan, mutta ei toiminnaltaan juurikaan muistuta sitä ja verbi taas sitä, että objektin toiminta muistuttaa jotakin reaali maailman toimintaa. Kuten mainittu, nämä kaksi voivat toteutua samanaikaisesti, jolloin niille on oma luokkansa.

Objektien merkityksiin perustuvissa luokitteluissa on yksi perustavanlaatuinen ongelma. Jos niillä yritetään luokitella kokonaisia järjestelmiä, niin ongelmia tulee vastaan aina kun järjestelmässä on useanlaisia objekteja. Tällöin voidaan sijoittaa järjestelmä siihen luokkaan, johon sen tärkeimmät objektit kuuluvat tai sitten järjestelmän voidaan käsittää kuuluvan useaan luokkaan. Ensimmäinen vaihtoehto on huono, sillä se jättää huomioimatta järjestelmän kokonaisuutena, ja jälkimmäisessä taas menetetään luokkien erottelukyky. Esiteltyjä luokitteluja voidaan kuitenkin hyvin

käyttää järjestelmien analysointiin ilman luokittelupäämäärää. Tällöin voidaan käsitellä kaikki objektit erillisinä ja saadaan tarkempaa tietoa järjestelmästä.

Osittain erot luokittelujen välillä ovat terminologisia, mutta eri ratkaisuihin on myös konkreettisia eroja. Holmquistin ja muiden [1999] luokittelu on hyvin kuvaava ja helppo ymmärtää, mutta kuten tekijät itsekin huomauttavat, on yksiselitteinen luokkiin jakaminen siinä ongelmallista. Erityisesti objektien luokittelu joko merkiksi tai työkaluksi saattaa aiheuttaa vaikeuksia. Ongelmia seuraa ainakin silloin, jos virtuaalisen objektin ominaisuuksia voidaan muokata sitä itseään esittävällä aineellisella objektilla. Myös objektien tarkoitukseen otettava jyrkkä kanta rajaa luokittelun käyttökelpoisuutta eri konteksteissa.

Underkofflerin ja Ishiin [1999] sekä Kolevan ja muiden [2003] luokittelut ovat hyvin samankaltaisia, vaikkakin ensimmäisessä ovat lähtökohtana virtuaaliset ja aineelliset objektit ja toisessa niiden välinen linkki. Kummassakin voi kuitenkin olla vaikeuksia sijoittaa joitakin objekteja yhteen luokkaan, vaikka luokkia on useita. Henkilökohtaisiin objekteihin keskittyvä luokittelu [van den Hoven and Eggen, 2004] osoittaa koko hyötynsä vasta sellaisia sovelluksia luokiteltaessa, joissa käytetään henkilökohtaisia esineitä, mutta tuo myös uusia näkökulmia kaikkien sovellusten analysointiin. Myös viimeiseksi esiteltyä Fishkinin [2004] luokittelua heikentää se, että objekteja analysoidaan yksittäisinä, jolloin eri objektit voivat kuulua eri luokkiin. Siinä myös eri sovellukset voivat kattaa monta aluetta, minkä vuoksi luokittelu menettää tärkeimmän merkityksensä eli sovellusten yksiselitteisen jakamisen eri luokkiin. Lisäksi sovellusten sijoittaminen kummallakin eri asteikolla oikeaan kohtaan saattaa olla vaikeaa. Luokittelussa on silti myös monia hyviä puolia. Se kattaa monia muita luokitteluja ja onnistuu lisäksi luokittelemaan sovelluksia, jotka eivät sisälly mihinkään muuhun luokitteluun. Se myös kuvaa havainnollisesti sitä aluetta, jolle erilaiset käsinkosketeltavat käyttöliittymät kuuluvat ja välittää informaatiota siitä, mitä mahdollisuuksia on vielä käyttämättä ja millä alueella on ruuhkaa. Fishkinin luokittelu antaa myös kuvan siitä, kuinka käsinkosketeltavia sovellukset ovat. Tässä yhteydessä täytyy kuitenkin muistaa, että vaikka jonkin sovelluksen käsinkosketeltavuuden aste olisi korkea, ei siinä silti ole välttämättä parempi käyttöliittymä kuin käsinkosketeltavuuden asteeltaan matalammassa sovelluksessa.

#### **4.2.3. Objektien suhteisiin perustuvat luokittelut**

Vaikka objektien merkitykset ovatkin usein helpoiten havaittavissa, on useissa käsinkosketeltavissa käyttöliittymissä järjestelmän kannalta vielä enemmän merkitystä objektien suhteilla. Suhteisiin perustuvia luokitteluja on vähemmän kuin merkityksiin perustuvia, mutta ne ovat pääosin tuoreempia.

Taulukko 5. Objektien analysointi Shaerin ja muiden [2004] mukaan.

	Esitys		Toiminta		
	Merkki	Rajoitteet	Muuttuja	Tapahtuma	Palaute
TAC					

TAC-paradigma [Calvillo-Gómez *et al.*, 2003; Shaer *et al.*, 2004] tarjoaa hieman erilaisen työkalun käsinkosketeltavien käyttöliittymien analysointiin, sillä sen avulla ei ole tarkoituskaan luokitella käyttöliittymiä, vaan kuvailla ja arvioida niitä (Taulukko 5). Tässä teoriassa kaikki käyttöliittymään kuuluvat aineelliset objektit (pyfo) voivat olla joko merkkejä (token), rajoitteita (constraint) tai molempia.

Analysoinnin lähtökohtana ovat merkit, joilla tarkoitetaan niitä aineellisia objekteja, joihin voidaan tarttua ja jotka edustavat digitaalista informaatiota. Merkin käyttäytymistä ja käsittelemistä rajoittavat aineelliset objektit taas ovat sen rajoitteita. Yksi TAC (Token And its Constraints) koostuu siis merkistä ja sen rajoitteista. Jotta objekti voi olla merkki, täytyy siihen liittyä vähintään yksi rajoite sekä digitaalista informaatiota edustava muuttuja. Merkki voi rajoittaa muiden merkkien käyttäytymistä, jolloin se on rajoitteena muissa TACeissa. Yksi rajoite voi myös samanaikaisesti rajoittaa useita merkkejä. Kaikkien merkkien on oltava tietoisia omista rajoitteistaan, mutta rajoitteiden ei tarvitse tietää niihin liittyviä merkkejä. Digitaaliseen informaatioon merkit on sidottu muuttujien kautta ja jokaisen TACin on liityttävä johonkin muuttujaan ja muutettava sitä joko erillisen (discrete) tai jatkuvan (continuous) toiminnan kautta. Erilliseksi muuttujaksi voidaan lukea esimerkiksi napin painalluksesta vaihtuva muuttuja, kun taas jonkin objektin suuntaa ilmaiseva muuttuja on jatkuva.

Rajoite voi määrittää merkkien toimintaa kolmella erilaisella tavalla. Ensinnäkin sen havaittavissa olevat ominaisuudet voivat antaa käyttäjälle ohjeita siitä, kuinka rajoitteeseen liittyvää merkkiä tulisi käsitellä. Toisaalta rajoite saattaa myös fyysisesti määrittää sen tilan, jossa merkkiä voidaan liikuttaa. Kolmanneksi rajoite voi toimia koordinaatistona, jonka suhteen merkkien toimintaa tulkitaan ja muunnetaan laskennallisiksi funktioiksi.

Kun edellisessä luokittelussa käsinkosketeltavia käyttöliittymiä analysoitiin jokaisen niihin kuuluvan aineellisen objektin kohdalla erikseen, Ullmer ja Ishii [2000] esittävät neljä kategoriaa, joiden avulla voidaan tulkita useiden objektien muodostamaa järjestelmää kokonaisuutena (luokittelun ei kuitenkaan ole tarkoitus olla kattava). Heidän mukaansa luokittelun avulla voidaan analysoida objektien suhdetta digitaaliseen informaatioon. Alun perin luokkia oli neljä: miellelyhtymiin perustuva (associative), tilaan perustuva (spatial), suhteisiin perustuva (relational) ja rakenteisiin perustuva (constructive). Myöhemmin he kuitenkin jättivät miellelyhtymiin perustuvan luokan pois, koska eivät olleet sen hyödyllisyydestä yhtä varmoja kuin kolmen muun luokan [Ullmer and Ishii, 2000]. Tässä tutkimuksessa kaikki neljä ovat kuitenkin mukana, sillä

myös pois jätetyllä luokalla on osoitettu olevan arvoa [van den Hoven and Eggen, 2004].

Mielleyhtymiin perustuvissa lähestymistavoissa objektit liittyvät kukin itsenäisesti digitaaliseen informaatioon, eivätkä saa lisämerkityksiä suhteistaan muihin objekteihin. Tilaan perustuvissa järjestelmissä merkkien avaruudellinen sijainti ja suunta voidaan tulkita suoraan syötteeksi, kun taas suhteisiin perustuvissa järjestelmissä tulkinta tehdään merkkien peräkkäisyyksien, vierekkäisyyksien tai muiden vastaavanlaisten suhteiden perusteella. Rakenteisiin perustuvissa järjestelmissä merkit ovat moduuleita, jotka voidaan kiinnittää toisiinsa rakennuspalikoiden tavoin mekaanisesti.

Kokonaisia käsinkosketeltavia käyttöliittymiä on yritetty luokitella myös niiden rakenteen perusteella. Ullmerin ja muiden [2005] luokittelussa käsinkosketeltavat käyttöliittymät jaetaan kolmeen lajityyppiin: vuorovaikutteisiin pintoihin (interactive surface), rakennettaviin järjestelmiin (constructive assembly) ja merkki+rajoitteet-järjestelmiin (tokens and constraints). Vuorovaikutteisiksi pinnoiksi voidaan käsittää graafisesti rikastetut tasot ja seinät, joiden pinnalla sijaitsevia aineellisia objekteja käyttäjät käsittelevät. Rakennettaviksi järjestelmiksi puolestaan luetaan sellaiset käyttöliittymät, joissa vuorovaikutus perustuu rakennuspalikoiden tavoin toimivien objektien yhteen liittämiseen tai kokoamiseen. Merkki+rajoitteet-järjestelmissä merkit sisältävät viittauksia digitaaliseen informaatioon ja tietokone tulkitsee niiden käsittelemistä, joka tapahtuu niiden omien rajoitteiden asettamissa puitteissa. Merkki+rajoitteet-järjestelmien vuorovaikutus voidaan jakaa kahteen vaiheeseen: yhdistämiseen (associate) ja käsittelemiseen (manipulate). Yhdistämisessä merkki asetetaan johonkin rajoitteiden määrittämään ja järjestelmän kannalta merkitykselliseen paikkaan tai sijaintiin. Yhdistämisen jälkeen objektin käsitteleminen on mahdollista rajoitteiden sallimissa puitteissa, esimerkiksi objektia pyörittämällä tai liikuttamalla. Osa merkki+rajoitteet-järjestelmistä tukee vain yhdistämisvaihetta.

Hurtienne ja Israel [2007] yrittävät helpottaa käsinkosketeltavien käyttöliittymien analysointia ja luokittelua mielikuvaskaemojen (image schema) avulla. Mielikuvaskaemat ovat teoreettisia malleja, jotka syntyvät tarkkaillessamme maailmaa ja joiden avulla yritämme ymmärtää maailmaa. Esimerkki käsinkosketeltavissa käyttöliittymissä usein esiintyvästä skeemasta on säiliö (container), joka koostuu abstraktina käsitteenä ulkoisesta alueesta, suljetusta alueesta ja rajasta niiden välillä. Konkreettisenä osana käyttöliittymää säiliö-käsite voi vertauskuvallisesti edustaa esimerkiksi ryhmään kuuluvuutta, turvallisuutta tai täysinäisyyden astetta. Skeemojen vertauskuvalliset käyttötavat kuuluvatkin oleellisena osana esitettyyn luokitteluun.

Mielikuvaskaemat ovat riippumattomia modaliteeteista ja erilaisia skeemoja on tunnistettu useita kymmeniä. Hurtienne ja Israel luokittelevat skeemat kahdeksaan luokkaan: perus- (basic), tila- (space), sisältyvyys- (containment), identiteetti- (identity), monilukuisuus- (multiplicity), prosessi- (process), voima- (force) ja

ominaisuusskeemoihin (attribute). He myös toteavat, että heidän esittämäänsä luokittelua voi käyttää rinnakkain muiden luokittelujen kanssa, ja pitävät luokitteluaan hyödyllisenä siirryttäessä yhä abstraktimpiin käsinkosketeltavien käyttöliittymien käyttötapoihin.

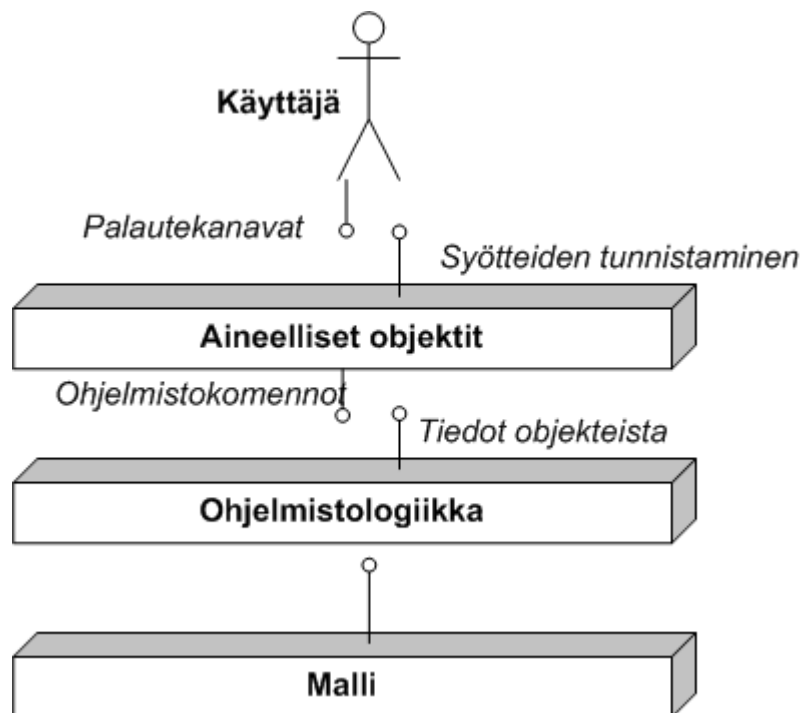
Vaikka objektien suhteisiin perustuvilla luokitteluilla olisi mielestäni paljon potentiaalia käsinkosketeltavien käyttöliittymien luokittelussa, liittyy niihin samankaltaisia ongelmia kuin objektien merkityksiin perustuviin luokitteluihin. Vaikka suhteisiin perustuvien luokittelujen avulla voidaan periaatteessa luokitella kokonaisia järjestelmiä, saattavat eri objektien suhteet vaihdella järjestelmän sisällä, jolloin yksiselitteistä luokittelua ei välttämättä voida saada aikaan.

Ensimmäisenä esitelty TAC-paradigma [Calvillo-Gómez *et al.*, 2003; Shaer *et al.*, 2004] huomioi objektien erilaisuuden käsittelemällä jokaisen objektin erikseen, mutta menettää samalla mahdollisuuden luokitteluun. Se on hyvä työkalu joidenkin käsinkosketeltavien käyttöliittymien ja niiden vuorovaikutuksen analysoinnissa, mutta ei sovi yhtä hyvin kaikkien sovellustyyppien analysointiin. Toisena esitellyn Ullmerin ja Ishiin [2000] jaottelun kolme ensimmäistä luokkaa sopivat hyvin yhteen, mutta vaikeuksia aiheuttaa neljäs, rakennettavia järjestelmiä kuvaava luokka. Mielestäni luokka on enemmänkin ominaisuus, joka voi yhtä hyvin liittyä mihin tahansa muista luokista, vaikka se on esitetty omana luokkana. Sen joukkoon kuulumattomuutta puolustaa myös se, että sama luokka esiintyy samanlaisena toisessa myöhemmin esitellyssä Ullmerin ja muiden [2005] luokittelussa, johon se sopii paremmin. Jälkimmäisen luokittelun ongelmana on kuitenkin se, että on olemassa käyttöliittymiä, jotka eivät kuulu mihinkään esitettyyn luokkaan. Luokittelua voidaan pitää enemmänkin hyvin kuvailevana kuin tarkasti eri ryhmiin jakavana. Luokat ovat kuitenkin erittäin konkreettisia ja antavat nopeasti yleiskuvan järjestelmän luonteesta. Mielestäni tämä luokittelu onkin lähimpänä onnistua kuvailemaan käsinkosketeltavia käyttöliittymiä yhdellä akselilla. Viimeisenä esitelty mielikuvaskaemoihin perustuva luokittelu [Hurtienne and Israel, 2007] menee muita luokitteluita syvemmälle etsien korkeamman tason malleja käytettyjen vertauskuvien takana. Se antaakin toivoa siitä, että käsinkosketeltavia käyttöliittymiä tullaan jatkossa soveltamaan entistä teoreettisemmille sovellusalueille.

### **4.3. Malli käsinkosketeltavien käyttöliittymien kehitykseen**

Edellä esitellyistä malleista ja luokituksista puuttuvat kaikki varsinaiseen käsinkosketeltavien käyttöliittymien kehittämiseen liittyvät seikat. Useat malleista on tarkoitettu valmiiden käyttöliittymien luokitteluun ja analysointiin. Vaikka Ullmerin ja Ishiin [2001] MCRpd-malli muistuttaakin graafisten käyttöliittymien kehityksessä käytettävää MVC-mallia, ei sitäkään ole suunniteltu varsinaisesti sovellusten kehittämiseen. Seuraavaksi esittelen korkean abstraktiotason mallin, jonka on tarkoitus korostaa käsinkosketeltavien käyttöliittymien kehityksessä oleellisia seikkoja.

Kerroksittainen malli koostuu neljästä konstruktiosta (Kuva 22). Alimmaisena on järjestelmän malli-osuus. Se tarkoittaa sovellusaluekohtaista tietomallia ja sitä ei tässä tutkimuksessa käsitellä, sillä samaa mallia voidaan käyttää sekä käsinkosketeltavassa että jossakin muunlaisessa järjestelmässä. Mallilla on rajapinta, jota käyttää järjestelmän ohjelmistologiikka. Se saa tietoja käyttäjän toiminnasta aineellisten objektien toimittamana ja vastaavasti malliin tekemiensä muutosten jälkeen antaa objekteille komentoja. Tämä ohjelmiston ja aineellisten objektien välinen rajapinta on nähdäkseni eräs tärkeistä suunnittelukohdista käsinkosketeltavissa käyttöliittymissä, sillä siinä täytyy suunnitella, mitä tietoja aineelliset objektit pystyvät ohjelmistolle välittämään ja mitä komentoja ottamaan vastaan.



Kuva 22. Malli käsinkosketeltavien käyttöliittymien kehitykseen

Toinen tärkeä rajapinta mallissa on käyttäjän ja aineellisten objektien välissä. Sen suunnittelussa on tärkeä huomioida, mitä käyttäjän antamia syötteitä aineelliset objektit pystyvät tunnistamaan ja mitä palautetta antamaan käyttäjälle. Nämä tekijät yhdessä niiden soveltamisen kanssa määrittävät pitkälti sen, kuinka miellyttävää ja tehokasta käyttäjän vuorovaikutus järjestelmän kanssa on.

#### 4.4. Yhteenveto

Objektien merkityksiin perustuvia jaotteluja on ollut olemassa jo melko pitkään, mutta täydellistä jaotteluperustetta ei ole vielä löytynyt. Lähimmäksi tässä pääsee Fishkin [2004], jonka luokittelun avulla voidaan saavuttaa hyötyjä sovellusten suunnittelussa, analysoinnissa ja vertailussa. Objektien merkityksiin perustuvissa luokitteluissa on

ongelmana se, että yhteen käyttöliittymään sisältyy useasti monen tyyppisiä objekteja, jolloin päätelmiä koko järjestelmästä on vaikea yleistää luokittelua varten.

Objektien suhteisiin perustuvia luokitteluja ei ole ollut olemassa yhtä kauan kuin objektien merkityksiin perustuvia luokitteluja. Niiden etuna ovat kuitenkin paremmat edellytykset kokonaisten järjestelmien luokitteluun. Vielä ne eivät analysoinnin tarkkuudessa kuitenkaan pärjää parhaille merkitysten analysointiin perustuvilla luokitteluille. Underkofflerin ja Ishiin [1999] luokittelu ei ole täydellinen, mutta kuvaa yksinkertaisesti muutamalla luokalla koko suunnittelutilaa.

Edellä esiteltyjen lisäksi on olemassa myös muita luokitteluja ja teorioita, joita voidaan soveltaa käsinkosketeltaviin käyttöliittymiin, mutta jotka kuitenkin sopivat myös muihin vuorovaikutustapoihin (esimerkiksi [Wensveen *et al.*, 2004]). Tässä luvussa olen kuitenkin keskittynyt vertailemaan vain niitä teorioita, jotka kuvaavat käsinkosketeltavia käyttöliittymiä kokonaisuutena määrittelyjen rajoja ylittämättä.

Mielestäni yksikään edellä esitellyistä käsinkosketeltavien käyttöliittymien luokitteluista ei ole niin hyvä, että tutkimus aihepiiriin liittyen kannattaisi lopettaa. Ei ole kuitenkaan selvää, kannattaisiko painopiste suunnata yksinkertaisen muutamaan luokkaan sovellukset jakavan luokittelun aikaansaamiseen vai useampiulotteisen luokittelun luomiseen aivan uusien akseleiden ympärille. Tutkimussuuntauksen tulevaisuudessa löytyy varmaankin tarvetta molemmille.

## 5. Käsinkosketeltavien käyttöliittymien arkkitehtuureja

Käsinkosketeltavat käyttöliittymät ovat osa pyrkimystä siirtyä ikkuna, ikoni, valikko ja osoitin -vuorovaikutusmallista kohti intuitiivisempia ja ihmisen luonnollisia kykyjä paremmin hyödyntäviä käyttöliittymiä. Siinä missä aikaisemmat käyttöliittymät ovat hyödyntäneet pääosin yhtä syöte- ja yhtä tulostuskanavaa sekä peräkkäistä ja vuoropohjaista vuorovaikutusta, uudet käyttöliittymät hyödyntävät jatkuvaa vuorovaikutusta, jossa on käytössä useita rinnakkaisia ja tahdistamattomia kanavia [Jacob *et al.*, 1999]. Shaer [2005] on käsinkosketeltavien käyttöliittymien hallintajärjestelmää kehittäessään havainnut uudenlaisten vuorovaikutustapojen suurimmiksi eroiksi vanhoihin verrattuna juuri useat samanaikaiset syötteet yhdeltä tai usealta käyttäjältä, rinnakkaisen fyysisen ja digitaalisen tulostuskanavan, jatkuvat ja erilliset syötteet sekä standardien puutteet syöte- ja tulostuslaitteissa. Nämä seikat eivät ainoastaan tee käyttöliittymien käyttämisestä erilaista, vaan myös asettavat uudenlaisia vaatimuksia niiden kehittämiseksi. Vanhat arkkitehtuuriratkaisut eivät enää vastaa uusia vaatimuksia niin hyvin, että niiden käyttäminen olisi järkevää. Tarvitaan siis uudenlaisia ratkaisuja, jotka mahdollistavat uusien vuorovaikutustapojen tehokkaan ja erityisesti käyttäjien mielikuvia vastaavan hyödyntämisen.

Uudenlaisia vuorovaikutustapoja hyödyntävien käyttöliittymien toteutusta tulisi myös pyrkiä helpottamaan luomalla työkaluja, jotka siirtävät kehitystyötä korkeammalle abstraktiotasolle. Työkalut voivat parantaa käyttöliittymäohjelmoijien työn laatua ja tuottavuutta esimerkiksi lisäämällä yhdenmukaisuutta käyttöliittymien välillä, tukemalla laajennettavuutta ja piilottamalla ympäristöstä riippuvaa monimutkaisuutta [Betts *et al.*, 1987]. Vaikka yleiskäyttöisiä tuoterunkoarkkitehtuureja ei käsinkosketeltaville käyttöliittymille vielä lähiaikoina onnistuttaisikaan toteuttamaan, voidaan uudelleenkäyttöä lisätä hyvin tarkoitukseensa sopivilla arkkitehtuureilla, joita voidaan käyttää uudelleen muiden sovellusten toteutuksissa.

### 5.1. Käsinkosketeltavien käyttöliittymien arkkitehtuuriratkaisuja

Tässä kohdassa kuvailen kuusi erilaista käsinkosketeltavien käyttöliittymien arkkitehtuuria tai arkkitehtuurissa käytettyä ratkaisua, joiden taustalla ovat erilaiset ongelmat. Ratkaisuista on kuitenkin löydettävissä lukuisia yhtäläisyyksiä, mikä saattaa osaltaan helpottaa hyvien toimintatapojen havaitsemista.

Ensimmäisessä alakohdassa esittelen ohjelmistokehyksen, joka mahdollistaa käsinkosketeltavien käyttöliittymien lisäksi myös useiden muiden vuorovaikutustapojen käyttämisen. Sen jälkeen kuvaan erityisesti käsinkosketeltavia käyttöliittymiä varten kehitetyn alusta-arkkitehtuurin, joka toteuttaa useita käsinkosketeltaviin käyttöliittymiin liittyviä teorioita käytännössä. Kolmannessa alakohdassa esiteltävä työkalukokoelma on tarkoitettu käsinkosketeltavien käyttöliittymien kehittämiseen ja erityisesti



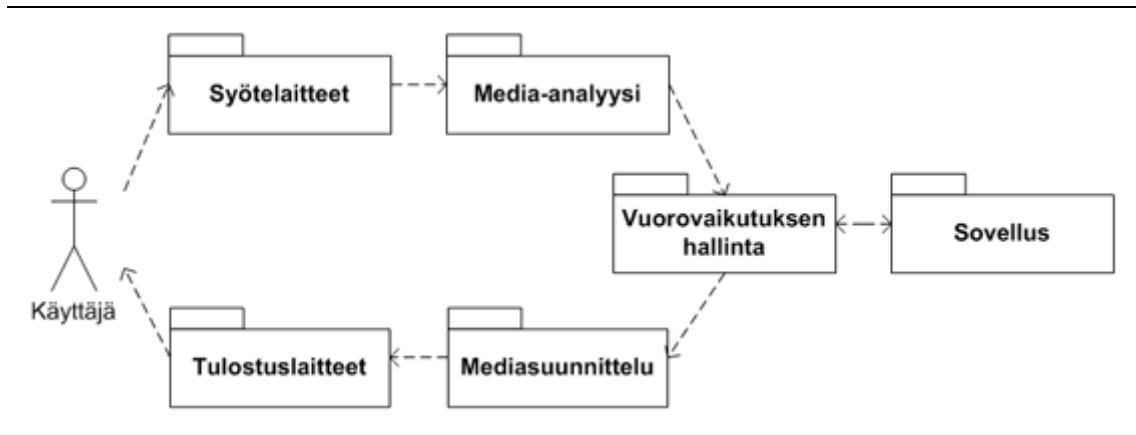
helpottamaan syötelaitteiden kanssa työskentelyä. Alakohdissa neljä, viisi ja kuusi kuvattavat ratkaisut on suunniteltu helpottamaan fyysisten syötelaitteiden liittämistä osaksi sovelluksia. Neljännessä alakohdassa kuvaan arkkitehtuurin, joka on alun perin tarkoitettu jokapaikan tietotekniikan sovelluksiin, mutta soveltuu myös käsinkosketeltavissa käyttöliittymissä käytettäväksi. Seuraavaksi esittelen laitekomponentteja, joita tarjotaan kehittäjien käyttöön sovelluksiin liitettäviksi ja jotka on varta vasten suunniteltu helpottamaan fyysisten käyttöliittymien kehittämistä. Kuudennessa alakohdassa esittelen arkkitehtuurin hajautettujen käsinkosketeltavien käyttöliittymien kehittämiseen.

### 5.1.1. DWARF

DWARF (distributed wearable augmented reality framework) [Sandor and Klinker, 2005] on useita uusia vuorovaikutustapoja ja niiden nopeaa vaihtamista tukeva ohjelmistokehys. Sandor ja Klinker kutsuvat tutkimaansa uusien käyttöliittymien aluetta jokapaikan lisätyksi todellisuudeksi (ubiquitous augmented reality) ja sanovat siihen kuuluvan multimediaa, moniaistisuutta, lisättyä todellisuutta sekä puettavaa ja jokapaikan tietotekniikkaa. Arkkitehtuuria käyttäen voidaan muiden vuorovaikutustapojen lisäksi toteuttaa myös käsinkosketeltavien käyttöliittymien piirteitä.

DWARF-kehiksen suunnittelu perustuu ongelma-alueen kartoitukselle. Eräs suunnitteluperusteista on resurssien liikkuvuus, josta aiheutuvaa joustavuutta tavoitellaan kehiksen modulaarisuuden avulla. Toinen suunnitteluperuste on moniaistisuuden järjestelmältä vaatima useiden kommunikointikanavien hyödyntäminen. Käsinkosketeltavien käyttöliittymien kannalta mielenkiintoisin suunnitteluperuste on aineellisten ja virtuaalisten objektien vuorovaikutuksen hallinta, jonka tekee mahdolliseksi arkkitehtuurin toteuttama aineellisten objektien jatkuva jäljittäminen.

Kehiksen päälle on mahdollista rakentaa monenlaisia sovelluksia. Sandor ja Klinker [2005] esittelevät tutkimuksessaan neljä kehiksen päälle rakennettua sovellusta, jotka kaikki toteuttavat käsinkosketeltavien käyttöliittymien piirteitä. Käytettäviä ominaisuuksia ovat esimerkiksi objektien tunnistaminen ja paikannus, olemassa olevien työkalujen hyödyntäminen sekä tarttumisen mahdollistavien linssien käyttö. Nämä ominaisuudet osoittavat, että huolimatta tuestaan muille vuorovaikutustavoille, tukee kehys myös käsinkosketeltavia käyttöliittymiä.



Kuva 23. DWARF arkkitehtuuri [Sandor and Klinker, 2005].

Kuvassa 23 on esitetty kehyksen arkkitehtuuri, joka koostuu viidestä alijärjestelmästä (syötelaitteet, media-analyysi, vuorovaikutuksen hallinta, mediasuunnittelu ja tulostuslaitteet). Jokainen niistä voi sisältää erilaisia komponentteja kulloinkin toteutettavasta järjestelmästä riippuen. Alijärjestelmät toimivat pääosin itsenäisesti muodostamalla omasta syötteestään tulosteen.

Syötelaitteita voi olla järjestelmässä useita ja niiden avulla voidaan toteuttaa moniaistisia järjestelmiä. Myös tulostuslaitteita voi olla käytössä samanaikaisesti useita, jolloin voidaan hyödyntää multimedian tarjoamia mahdollisuuksia.

Media-analyysin tehtävä on muuntaa syötelaitteilta saamansa käyttäjän antamat syötteen abstrakteiksi, järjestelmän sisäisiksi ilmauksiksi (token). Samassa alijärjestelmässä tapahtuu myös aineellisten syötteiden analysointi, joka voi tarkoittaa esimerkiksi aineellisten objektien paikannusta. Tulostuspuolella vastaava alijärjestelmä on mediasuunnittelu, jossa puolestaan valitaan informaatiota esitettäväksi käyttäjälle vaikkapa äänen tai kuvan muodossa.

Kehyksen sydän on vuorovaikutuksen hallinta, joka yhdistää eri syötelaitteilta tulevat ilmaukset ja päättelee niistä käyttäjän aikomukset. Ilmaukset voivat olla jatkuvia, eli saada mitä tahansa lukuarvoja, tai erillisiä, eli saada vain tiettyjä ennalta määrättyjä arvoja. Objektin kiertoa ilmaiseva astemäärä on esimerkki jatkuvasta ilmauksesta, kun taas puheentunnistuksen tunnistama sana on erillinen ilmaus. Jatkuvat ilmaukset voivat kuulua rajatulle välille tai saada mitä tahansa arvoja. Erilliset ilmaukset taas voivat olla totuusarvo- tai merkkijonotyyppisiä.

Saamistaan ilmauksista vuorovaikutuksen hallinta päättelee käyttäjälle näytettävän tulostuksen ja välittää sen eteenpäin komento-suunnittelumallia (command design pattern) [Gamma *et al.*, 1995] käyttäen. Multimodaalinen vuorovaikutus on toteutettu Petri-verkkojen (Petri net) avulla. Järjestelmän törmäystunnistus tunnistaa virtuaalisten objektien, todellisten objektien sekä virtuaalisten ja todellisten objektien törmäykset. Jäljitystieto voidaan välittää joko suoraan jäljityskomponenteilta näyttökomponenteille

tai käsitellä ensin putki-suodatin-suunnittelumallia (pipes and filters design pattern) [Buschmann *et al.*, 1996] käyttäen.

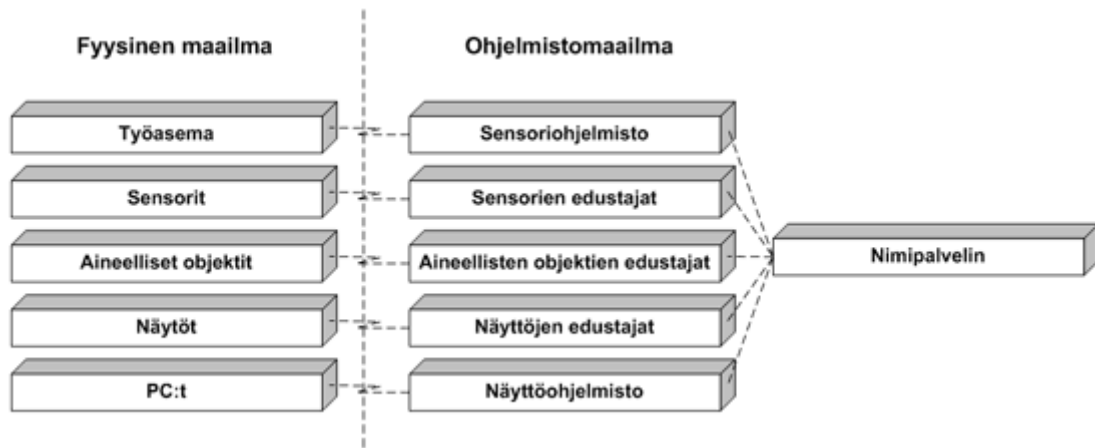
### 5.1.2. metaDESK

metaDESK [Ullmer and Ishii, 1997] on käyttöliittymäalusta, joka toteuttaa tekijöidensä Tangible bits -visiota [Ishii and Ullmer, 1997]. Siinä suunnittelun lähtökohtana on ollut tuoda tietokoneen työpöydältä tuttuja metaforia ihmisille tuttuun aineelliseen todellisuuteen. Aineelliset ikonit, työpöydän ikkunoita vastaavat linssit ja muut aineelliseen todellisuuteen tuodut graafisen käyttöliittymän elementit ovat alustassa käyttäjien kosketeltavissa ja käsiteltävissä. Niiden kautta voidaan myös tuoda aineelliseen todellisuuteen monia tietokoneen mahdollistamia lisäyksiä rikastuttamaan toimintaympäristöämme.

Arkkitehtuuri tukee vision metaforia antamalla kaikille aineellisille objekteille kyvyt tunnistaa oma tilansa, kommunikoida muiden objektien kanssa ja mahdollisesti näyttää tulosteita. Vain osassa objekteista sijaitsee todellisuudessa sensoreita, joilla edellä mainitut kyvyt voidaan itsenäisesti toteuttaa. Loput objekteista ovat täysin passiivisia eivätkä sisällä mitään elektroniikkaa. Tästä johtuen objektit ovat sovelluksen näkökulmasta erilaisessa asemassa. Ongelmaksi toteutuksessa tuli vaikeus säilyttää halutut vuorovaikutustapaan liittyvät metaforat ja samalla hallita järjestelmän monimutkaisuus. Ratkaisuksi tähän Ullmer ja Ishii [1997] kehittivät hajautettuihin edustajiin (proxy-distributed, proxdist) perustuvan arkkitehtuurin sekä alustariippumattoman meta-kielen [Ullmer, 1997], joka tarjoaa aineellisia objekteja vastaavat edustajaluokat.

Alustan fyysinen arkkitehtuuri koostuu useista objekteista ja laitteista. Alhaalta heijastettava, lähes vaakasuora taso toimii työpöytänä koko järjestelmälle. Sen yläpuolella varren varassa liikkuva aktiivinen linssi tarjoaa vaihtoehdoisen näkymän sovelluksen informaatioon. Passiivista linssiä ja aineellisia objekteja käyttäjä voi liikutella vapaasti tason pinnalla. Mainittujen osien lisäksi järjestelmän fyysiseen arkkitehtuuriin kuuluvat objektien sijainteja aistivat sensorit sekä tietokoneet, joille alusta on hajautettu.

Proxdist-arkkitehtuuri mahdollistaa tarvittavien kykyjen toteuttamisen passiivisten objektien puolesta, jolloin niiden varsinaiset toteutukset voivat sijaita hajautettuina järjestelmän tietokoneisiin. Kaikkia järjestelmän objekteja voidaan kuitenkin käsitellä samalla tavalla ja jokaiselle objektille voidaan määrittää rajapinta, joka määrää sen kyvyt. Palvelujen käyttäjät eivät kuitenkaan tiedä tarjoaako objekti itsessään palvelun vai tarjoaako sen objektin edustaja.



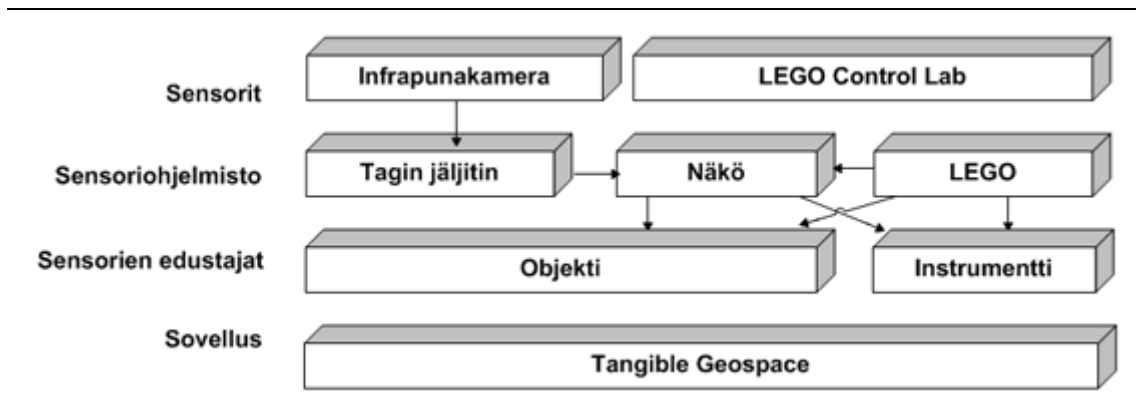
Kuva 24. metaDESK arkkitehtuuri. Muokattu lähteestä [Ullmer and Ishii, 1997].

Alustan arkkitehtuuri on kerroksittainen. Kuvan 24 kaavio esittää kerrokset eriteltyinä fyysiseen arkkitehtuuriin ja ohjelmistoarkkitehtuuriin. Fyysinen arkkitehtuuri koostuu tietokoneista, sensoreista, aineellisista objekteista ja näytöistä. Ohjelmistoarkkitehtuurin kerrokset vastaavat aineellisia laitteita sekä objekteja ja jakaantuvat keskeltä kahteen osaan riippuen siitä, onko kyseessä olevalla objektilla sensori- vai näyttöominaisuuksia. Alustaan kuuluu lisäksi nimipalvelin, joka abstrahoi ja koordinoi järjestelmän hajautettuja resursseja.

Alustan kerroksittainen rakenne vähentää järjestelmän riippuvuuksia yksityiskohtia abstrahoimalla. Asiakas- ja palvelinkerrokset mahdollistavat verkottuneen kommunikoinnin sensoreille ja näytöille. Sensorien ja näyttöjen edustajakerrokset taas piilottavat objektien edustajilta kykyjen toteuttamiseen käytettävät todelliset teknologiat, jolloin käytettävän teknologian määrää vasta sensoriohjelmisto. Objektien edustajakerros puolestaan piilottaa sovellukselta kunkin syöte- tai tulostuspalvelun oikean tarjoajan.

Alustan päälle on kehitetty Tangible Geospace -prototyypisovellus [Ullmer and Ishii, 1997], jossa käyttäjä voi rakennuksia symboloivia objekteja syötelaitteina käyttämällä liikuttaa pöytätasolle heijastettua karttaa. Samaan aikaan käyttäjä voi katsoa aktiivisen linssin läpi alueen kolmiulotteista näkymää, jonka kuvakulma vaihtuu aktiivisen linssin asennon ja sijainnin kulloinkin määräämänä. Käyttäjän on mahdollista myös katsoa vaihtoehtoista karttanäkymää tason päällä liikuteltavan passiivisen linssin läpi.

Kuvassa 25 on esitetty osa Tangible Geospace -sovelluksen arkkitehtuurista, joka perustuu proxdist-arkkitehtuuriin. Kuvasta nähdään, kuinka usean objektin edustajat voivat käyttää samaa sensoria ja yksi edustaja voi käyttää useita sensoreita. Kuvasta näkyy myös se, kuinka edustajakerrokset abstrahoiivat alemmat kerrokset pois ylempien kerroksien näkyvistä, yksinkertaistaen rakennetta.



Kuva 25. Tangible Geospace -arkkitehtuuri [Ullmer and Ishii, 1997].

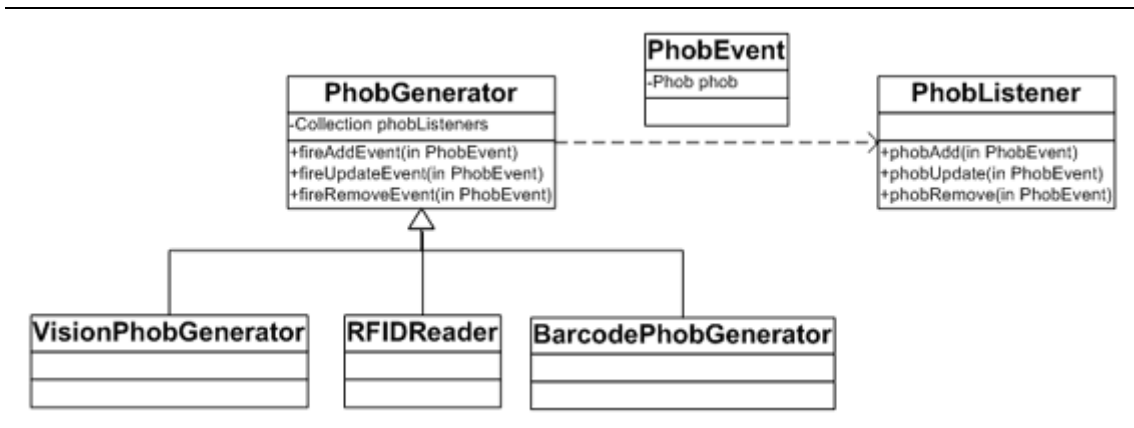
### 5.1.3. Papier-Mâché

Papier-Mâché [Klemmer *et al.*, 2004] on kokoelma työkaluja, joilla käsinkosketeltavien käyttöliittymien kehittämistä voidaan nopeuttaa ja helpottaa. Se tarjoaa kehittäjien käyttöön useita eri syötelaitteita tukevat kirjastot sekä erilaisia syötteiden kanssa työskentelyä helpottavia työkaluja. Työkalut voivat avustaa esimerkiksi virheiden etsinnässä ja syötteiden simuloinnissa.

Järjestelmän suunnittelun lähtökohtana on ollut helpottaa käsinkosketeltavien käyttöliittymien kehittäjien työskentelyä erilaisten syötelaitteiden ja niiltä saatujen syötteiden kanssa. Koska suuri osa käsinkosketeltavien käyttöliittymien kehityksajasta kuluu tällä hetkellä vaikeuksiin syötelaitteiden kehittämisessä sekä liittämässä järjestelmään, ja ongelmien ratkaisemiseen pystyvät vain alan asiantuntijat, on järjestelmän tarkoitus helpottaa ja nopeuttaa syötelaitteiden käyttöönottoa ja mahdollistaa niiden vaihtaminen mahdollisimman pienillä muutoksilla ohjelmakoodiin [Klemmer, 2003]. Järjestelmässä on toteutettuna kolme syöteteknologiaa: konenäkö, radiotaajuinen tunnistus (radio frequency identification, RFID) ja viivakooditunnistus. Mainittujen teknologioiden kanssa työskentelyä Papier-Mâché tukee abstrahoimalla laitteilta tulevan informaation käyttäjäystävälliseen ja syötelaiteriippumattomaan tapahtumamalliin, minkä ansiosta kehittäjä voi unohtaa syötelaittekohtaiset yksityiskohdat ja keskittyä korkeamman tason käsitteisiin.

Papier-Mâché helpottaa myös prototyyppien luomista mahdollistamalla konenäön käyttämisen. Sovelluksen prototyyppi voidaan kehittää käyttäen konenäköä, jonka jälkeen lopullinen versio voidaan toteuttaa käyttämään jotakin muuta syötelaiteteknologiaa. Vaatimuksena on vain se, että lopullisen version syöteteknologia pystyy tuottamaan sovelluksessa käytettävät tapahtumat. Papier-Mâchéssa on erillinen käyttöliittymä, joka tarjoaa näkymiä objektien ja tilojen seurantaan. Siinä on myös tälle sovellusalueelle edistyksellisen Ozin velho -toiminto (Wizard of Oz, WOz), joka mahdollistaa syötelaitteiden simuloinnin ilman niiden olemassaoloa ja helpottaa virheiden etsintää sekä testausta.

Papier-Mâchéä käyttäen on kehitetty useita käsinkosketeltavia käyttöliittymiä. SiteView-järjestelmän [Beckmann and Dey, 2003] avulla käyttäjä voi hallita kodin automaatiota. Käyttäjä käsittelee rakennuksen pohjapiirustuksen päällä aineellisia ikoneita, joiden avulla hän luo sääntöjä tietyin ehdoin tapahtuville automaatioiminnoille. Järjestelmä käyttää sensoreita ympäristön tilan havainnointiin ja antaa kuvallista palautetta sääntöjen vaikutuksista. Aineellisten ikonien tunnistus tapahtuu konenäön avulla; ikonien koko ja väri ovat tunnistamisessa käytettyjä ominaisuuksia.



Kuva 26. Papier-Mâché'n tapahtuma-arkkitehtuuri [Klemmer, 2003].

Papier-Mâché on kokoelma avoimen lähdekoodin Java-luokkia, joista osa esitetään kuvassa 26. Tapahtumien tuottaja (PhobGenerator) luo tunnistustensa perusteella tapahtumia (PhobEvent), jotka sisältävät tiedon niihin liittyvästä tunnistetusta objektista (Phob). Tunnistusteknologiasta riippuen käytössä on jokin tietty tuottaja, kuten viivakoodinlukijan tapauksessa BarcodePhobGenerator. Yhteystaulut (AssociationMap) liittävät tapahtumat konkreettisiin toimintoihin, kuten esimerkiksi äänitiedoston soittamiseen. Tapahtumakuuntelijat (PhobListener) puolestaan kuuntelevat tuottajien luomia tapahtumia ja pystyvät reagoimaan niihin. Tapahtumasuodattimet (EventFilter) mahdollistavat ennalta määriteltyjen objektien havaitsemisen ja paikallistamisen konenäön avulla.

Tapahtumat voivat sisältää toisistaan eroavaa informaatiota riippuen syötelaitteen tarjoamista mahdollisuuksista. Konenäkö esimerkiksi välittää kohteen koon, asennon, värin ja sijainnin, kun taas radiotaajuinen tunnistus pelkästään kohteen ja lukijan tunnisteet. Tapahtumat ovat kuitenkin eri laitteiden välillä yhdenmukaisia, joten rajoitetusti käytettynä ne mahdollistavat laiteriippumattomuuden.

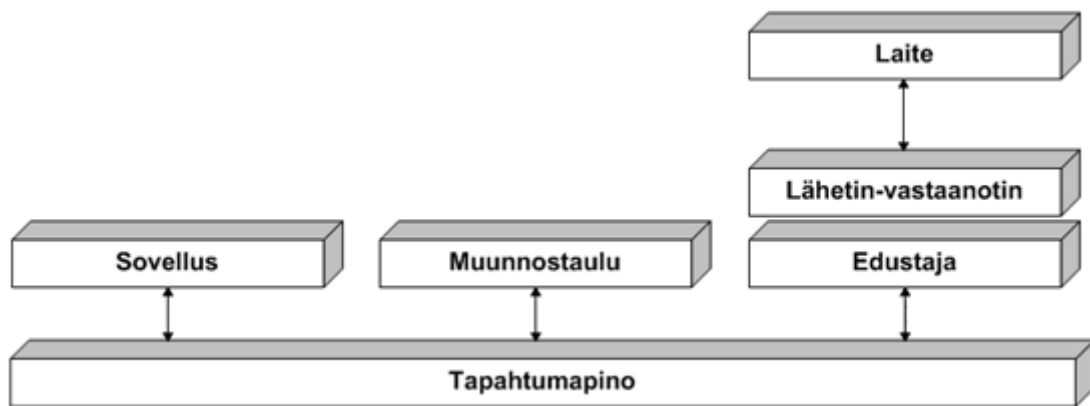
#### 5.1.4. iStuff

iStuff [Ballagas *et al.*, 2003] on kehitetty helpottamaan uusia käyttöliittymäteknikoita toteuttavien prototyyppien kehittämistä erityisesti jokapaikan tietotekniikan alueella. Sen tarkoituksena on mahdollistaa monien eri syöte- ja tulostuslaitteiden samanaikainen

käyttö. iStuffin avulla voidaan yhdistää lukuisia langattomia laitteita sovellukseen siten, että eri laitteet voivat toimia ohjaimina samoille toiminnoille.

iStuff on ollut käytössä neuvottelutilaa muistuttavassa, interaktiivisessa iRoom-huoneessa. iStuff-laitekomponentteja on kehitetty toimintoihin, joilla pystytään vaikuttamaan huoneessa käytössä oleviin sovelluksiin. Erilaisia kytkimiä ja sensoreita on liitetty esimerkiksi vaikuttamaan ympäristön tilaan ja huoneen seinällä olevan näyttötaulun tapahtumiin.

Kehittäjät asettivat järjestelmälle vaatimuksiksi laajan laitevalikoiman tukemisen, riippumattomuuden alustasta ja langattomasta protokollasta, olemassa oleviin järjestelmiin liittämisen helppouden ja tuen monille samanaikaisille käyttäjille. Tätä varten he kehittivät kuvassa 27 esitetyn iStuff-arkkitehtuurin.



Kuva 27. iStuff arkkitehtuuri [Ballagas *et al.*, 2003].

Fyysinen laite ja siihen liittyvä ohjelmistoedustaja muodostavat iStuff-komponentin, johon kuuluvan laitteen monimutkaisuus voi vaihdella melko monimutkaisesta hyvin yksinkertaiseen. Jotta jokin laite voidaan liittää iStuff-ympäristöön, täytyy laitetta varten olla toteutettuna edustaja. Jokaisella laitteella täytyy siis olla edustaja, mutta sama edustaja voi edustaa myös muita laitteita. Riippuen siitä, onko laite syöte- vai tulostuslaite, täytyy sen edustajan joko kapseloida dataa tapahtumiin tai osata ottaa sitä sieltä. Komponenttirakenne tekee laitteiden liittämistä ympäristöön helppoa ja mahdollistaa liittämisen myös olemassa oleville kolmansien osapuolten kehittämille laitteille.

iStuff-komponentit kommunikoivat sovellusten kanssa monikoiden (tuple) avulla. Monikoiden käyttötarkoituksesta johtuen niitä kutsutaan tässä tapahtumiksi. Kullakin tapahtumalla on tyyppi sekä avain-arvo-pareja, joissa on tapahtuman kannalta oleellista informaatiota. Tuottajat välittävät tapahtumia tapahtumapinolle (event heap) [Johanson and Fox, 2002], joka taas välittää ne kuluttajille. Kuluttajat saavat käsiteltäväkseen vain ne tapahtumat, joita ne ovat rekisteröityneet kuuntelemaan. Tuottajat ja kuluttajat voivat myös kuulua ryhmiin, jolloin tapahtumien lähetys ja vastaanotto voidaan määritellä ryhmien perusteella.

Järjestelmän keskeisin osa on tapahtumapino, joka on tapahtumia asiakkailta vastaanottava ja niitä eteenpäin määrätyille vastaanottajille välittävä palvelinprosessi. Kaikki eri laitteiden ja sovellusten tapahtumat menevät tapahtumapinoon, joka välittää ne eteenpäin kaiken tyyppisiä tapahtumia kuuntelevalle muunnostaululle (patch panel). Muunnostaulun [Ballagas *et al.*, 2004] avulla komponenttien tapahtumat voidaan kytkeä haluttuihin sovelluksiin, sillä muunnostaulu määrittää, kuinka laitteilta tulevat tapahtumat muunnetaan sovelluksen ymmärtämiksi tapahtumiksi.

Muunnokset tehdään alkuperäiset tapahtumat säilyttäen. Yksi lähdetapahtuma voidaan jakaa useiksi eri tapahtumiksi ja monta tapahtumaa voidaan määrittää yhdessä toteutuessaan laukaisemaan uusi tapahtuma. Muunnoksen yhteydessä voidaan suorittaa muutoksia tai vaihdoksia myös tapahtumien avaimille ja arvoille.

Muunnostaulu voidaan ajonaikaisesti määrätä tekemään uudentyyppisiä tapahtumamuunnoksia ja se mahdollistaa tilakoneiden käytön tapahtumien avulla. iStuff-komponentit määräävät tuottamiensa tapahtumien tyypit, mutta kehittäjiä kehoitetaan käyttämään kunkin sovelluksen kontekstissa merkityksellisiä tapahtumia, jotta laitteiden vaihtaminen olisi mahdollista ilman muutoksia sovelluksen koodiin.

### 5.1.5. Laitekomponentit

Phidgetsien [Greenberg and Fitchett, 2001] tarkoitus on helpottaa käsinkosketeltavien käyttöliittymien luontia samalla tavalla kuin valmiit käyttöliittymäkomponentit (widget) ovat helpottaneet graafisten käyttöliittymien kehittämistä, eli abstrahoimalla syöte- ja tulostuskokonaisuuksia valmiiksi komponenteiksi. Ratkaistavana ongelmana on siis laitteiden yksinkertainen liittäminen osaksi sovelluksen ohjelmakoodia, sillä nykyisellään yksinkertaistenkin elektronisten laitteiden kokoaminen ja ohjelmointi on hidasta ja vaivalloista. Liitettäviä laitteita voivat olla erilaiset sensorit, moottorit, kamerat tai näytöt.

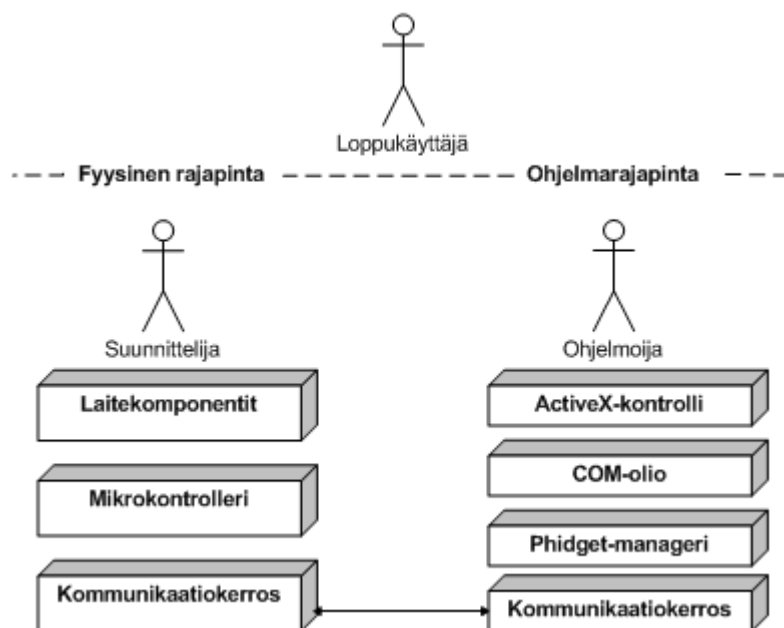
Olellaisena osana graafisten käyttöliittymäkomponenttien menestystä ovat olleet tarkasti määritellyt rajapinnat, jotka ovat mahdollistaneet komponenttien helpon käytön, vaikka itse toiminnallisuus onkin ollut kapseloituna kehittäjän näkymättömiin. Siten kehittäjät ovat pystyneet keskittymään olennaiseen eli vuorovaikutuksen suunnitteluun. Phidgetseillä on kuitenkin graafisista käyttöliittymäkomponenteista eroavia ominaisuuksia, jotka lisäävät niille asetettuja vaatimuksia. Niiden on pystyttävä informoimaan sovellusta kulloinkin saatavilla olevista laitteista ja erottelemaan yksiselitteisesti laitteet toisistaan. Kehittäjillä olisi myös hyvä olla mahdollisuus sovelluksen testaukseen, vaikka jotakin laitetta ei olisikaan saatavilla.

Phidgetsejä käyttäen on kehitetty monia pieniä sovelluksia testaamaan arkkitehtuurin toimivuutta. Alan opiskelijoille annettiin tehtäväksi toteuttaa Phidgetsejä käyttämällä mielikuvituksellinen käyttöliittymä ilman aikaisempaa kokemusta kyseisistä laitteista [Greenberg and Fitchett, 2001]. Tuloksena oli pieniä sovelluksia, joissa oli käytetty hyväksi erilaisia elektronisia sensoreita ja moottoreita. Esimerkkinä voidaan



mainita sovellus, joka näyttää kääntyvän viisarin avulla puhelimeen tulleet vastaamattomat puhelut. Toteutuksessaan se käyttää hyväkseen puhelimelle varta vasten suunniteltua jalustaa, joka aistii puhelun tärinää havaitsevilla sensoreilla ja kääntää viisaria ohjelmaan liitetyn moottorin avulla.

Kuvassa 28 on esitetty Phidgetsien arkkitehtuuri, joka voidaan jakaa karkeasti kolmeen osaan: itse fyysiseen laitteeseen, siihen liittyvään ohjelmistokoodiin ja niiden väliseen kommunikointiprotokolla. Fyysinen laite koostuu elektronisista komponenteista ja on liitettävissä tietokoneen USB-porttiin. Kommunikointiprotokolla välittää informaatiota laitteen ja tietokoneen välillä, mutta ei näy käyttäjälle. Laitteet osaavat välittää tyyppinsä, tunnistenumeronsa ja omaan tilaansa liittyviä viestejä sekä vastaanottaa tietokoneen esittämiä pyyntöjä.



Kuva 28. Phidgets arkkitehtuuri [Greenberg and Fitchett, 2001].

Ohjelmistopuolella Phidget-manageri tarjoaa kehittäjälle tavan päästä käsiksi tietokoneeseen sillä hetkellä kiinnitettynä oleviin laitteisiin. Se ilmoittaa sovelluksille uusien laitteiden liittämistä ja vanhojen poistamisesta. Phidget-manageri luo kaikista laitteista COM-objektin, jota kehittäjä voi käyttää joko yleisen rajapinnan tai laitteille yksilöllisten rajapintojen kautta. Tietyille laitteille ominainen rajapinta on laajempi ja sisältää kutsut kaikkiin kyseisen laitteen toteuttamiin toimintoihin. COM-objekteista luodaan myös ActiveX-kontrollit, jotka mahdollistavat graafisen näkymän laitteisiin ja laitteiden simuloinnin ohjelmakoodin avulla.

Phidgetsin kanssa samoihin haasteisiin vastaa Calder-toolkit [Lee *et al.*, 2004], joka keskittyy kenties vielä enemmän prototyyppien toteuttamisen helpottamiseen, ottamalla huomioon suunnittelijoiden työssään käyttämät menetelmät. Myös Calder-toolkit tarjoaa kehittäjille joukon uudelleenkäytettäviä syöte- ja tulostuskomponentteja, jotka

voivat Calder-toolkitin tapauksessa olla myös langattomia. Toteutuksen arkkitehtuuri keskittyy suurelta osin elektronisiin laitteisiin. Ohjelmistoarkkitehtuurissa jokaista laitetta vastaa olio, joka esittää laitteen tilan ja kapseloi sen toiminnallisuuden siten, että laitteen ja olion tilat vastaavat aina toisiaan. Myös Calder-toolkit näyttää laitteissa tapahtuvat muutokset ohjelmistopuolen tapahtumina.

Hieman erilainen lähestymistapa on otettu käyttöön Smart-Its-teknologiassa [Gellersen, 2005], jossa yksi Smart-It muodostaa modulaarisen kokonaisuuden, jossa voi olla vaihteleva määrä sensoreita, kommunikaatiolaitteita ja muita komponentteja liitettynä virtalähteeseen ja mikroprosessoriin. Smart-It voidaan kiinnittää lähes mihin tahansa objektiin, jolloin sitä voidaan esimerkiksi jäljittää tai siitä voidaan sensoreiden avulla kerätä tietoa. Smart-Itteihin voidaan upottaa ohjelmia, jolloin ne voivat esimerkiksi käsitellä keräämäänsä dataa ja lähettää sitä verkon yli eteenpäin.

### 5.1.6. Shared Phidgets

Shared Phidgets [Marquardt and Greenberg, 2007] rakentuu edellä esitellyn perinteisen Phidgets-arkkitehtuurin päälle ja laajentaa sitä soveltuvaksi hajautettujen järjestelmien kehittämiseen. Phidgets-laitekomponentteja voidaan käyttää hyväksi aivan kuten normaalistikin, mutta myös muiden laitteiden käyttäminen onnistuu arkkitehtuurissa. Laitekomponentti liitetään USB-liittimellä isäntäkoneeseen, joka voi olla PC tai esimerkiksi PDA. Kaikki isäntäkoneet liittyvät keskitettyyn palvelimeen, mutta ne voivat tehdä sen monella tavalla – esimerkiksi langattomasti. Palvelimella säilytetään avain-arvo-pareja, jotka sisältävät järjestelmään kuuluvien laitteiden tilat. Laitteiden isäntäkoneilla toimivat yhdistäjä-prosessit (connector) kirjoittavat dataa palvelimelle ja palvelin ilmoittaa muutoksista kaikille rekisteröityneille kuuntelijoille.

Shared Phidgets -sovelluksia voidaan ohjelmoida kolmella eri abstraktiotasolla. Ohjelmointi voidaan suorittaa suoraan vasten keskitetyn palvelimen tarjoamaa ohjelmointirajapintaa, jolloin päästään tehokkaimmin kiinni laitteiden toiminnallisuuteen. Toisen tavan tarjoavat edustaja-suunnitelumallin (proxy) [Gamma *et al.*, 1995] mukaiset kirjastot, jotka mahdollistavat keskitetyn tietovaraston käyttämisen, mutta piilottavat sen normaalin Phidgets-toiminnallisuuden taakse. Kolmas käyttötapa on toteuttaa laitteille graafiset vastineet, jolloin laitteiden tilaa voidaan tarkkailla ja ohjata normaalia graafista käyttöliittymää käyttäen.

Laitteisiin voidaan liittää keskitetysti säilytettyä, kontekstisidonnaista metadataa, jolloin järjestelmään kuuluvaa yleistä informaatiota voidaan säilyttää yhtenäisellä tavalla. Shared Phidgets -arkkitehtuuria voidaan myös laajentaa perimällä abstrakteista kantaluokista omia luokkia. Ympäristö tarjoaa ylläpitotyökalut keskitetyn tietovaraston ja laitteiden tilojen tutkimiseen. Shared Phidgets arkkitehtuuri helpottaa hajautettujen käsinkosketeltavien käyttöliittymien toteutusta, sillä se:

- löytää automaattisesti verkon eri koneisiin kiinnitettyt laitteet,

- piilottaa verkko-ohjelmoinnin sovelluskehittäjiltä,
- käyttää kaikkia laitteita saman rajapinnan kautta,
- rakentuu kerrosarkkitehtuurille,
- tukee ilmoituksia laitteilta,
- mahdollistaa laitteiden simuloinnin,
- helpottaa graafisten liittymien tekemistä laitteille, ja
- tarjoaa ylläpitotyökalut verkon ja siihen kuuluvien laitteiden hallintaan.

Shared Phidgets -arkkitehtuuri on osa suuntausta kohti entistä hajautetumpia käsinkosketeltavia käyttöliittymiä. Hyvä esimerkki hajautuksen äärimmilleen vievästä käsinkosketeltavasta käyttöliittymästä on Siftables [Merrill *et al.*, 2007], joka ammentaa inspiraationsa sensoriverkoista. Siftables koostuu tulitikkurasiaa pienemmistä näytöllisistä yksiköistä, jotka pystyvät tunnistamaan omat liikkeensä ja osaavat kommunikoida langattomasti keskenään radioteitse. Toisin kuitenkin kuin Shared Phidgets -arkkitehtuurissa, Siftables-komponentit eivät käytä lainkaan keskitettyä tietovarastoa.

## 5.2. Yhteenveto

Seuraavaksi vertailen tässä luvussa esittelemiäni arkkitehtuureja kahdesta eri näkökulmasta sekä esitän joitakin arvioita käsinkosketeltavien käyttöliittymien arkkitehtuurien tulevaisuudesta. Aluksi arvioin arkkitehtuureja sen perusteella, kuinka hyvin niissä käytetyt ratkaisut tukevat käyttöliittymänhallintajärjestelmältä (user interface management system, UIMS) vaadittavia piirteitä. Sen jälkeen vertailen kuinka hyvin arkkitehtuurit toteuttavat uusien vuorovaikutustapojen asettamia vaatimuksia. Kummassakin tapauksessa arviointi perustuu kunkin arkkitehtuurin esittelyn yhteydessä mainittuihin artikkeleihin.

Käyttöliittymänhallintajärjestelmältä vaadittavien ominaisuuksien vertailuun käytän valitsemiani ja muokkaamiani ominaisuuksia Bettsin ja muiden [1987] luokittelusta. Uudelleenkäytettävyyden ja laajennettavuuden vertailua hankaloittaa arkkitehtuurien erilaisuus. DWARF-kehukseen toteutettuja komponentteja voidaan käyttää uudestaan muissa kehysten päälle toteutetuissa sovelluksissa ja kehys on itse laajennettavissa komponenteilla. Edustajien avulla uudelleenkäyttöä ja laajennettavuutta parantavat metaDESK ja iStuff sekä tapahtumamallin avulla Papier-Mâché ja iStuff. Kaikki arkkitehtuurit ovat myös itsessään tarkoitettu uudelleenkäytettäväksi.

Prototyypin kehittämistä arkkitehtuurit tukevat eri tasoilla. DWARF tarjoaa sovellusten pohjaksi ohjelmistokehysten ja metaDESK taas fyysisistä laitteista ja ohjelmistosta koostuvan alustan. Papier-Mâché ja iStuff sisältävät valmiin tapahtumamallin, jota sovellukset voivat hyödyntää. iStuff ja Phidgets mahdollistavat lisäksi laitteiden nopean liittämisen.

Syötelaitteiden vaihtaminen on hyvin tuettua. Laiteliittymiä toteuttamalla sen mahdollistavat DWARF-kehys, metaDESK ja iStuff. Papier-Mâchéssa vaihdon voi tehdä valmiiksi toteutettujen syöteteknologioiden välillä. Syötelaitteiden testauksen avuksi Papier-Mâché ja Phidgets tarjoavat simulointityökalun.

Seuraavaksi arvioin mitä uusille vuorovaikutustavoille tyypilliseksi havaittuja ominaisuuksia [Jacob *et al.*, 1999; Shaer, 2005] tässä luvussa esitellyt arkkitehtuurit tukevat. DWARF-kehys tukee vaadittuja ominaisuuksia parhaiten, mutta myös metaDESK ja iStuff tukevat niitä hyvin. Muut arkkitehtuurit on havaintojen perusteella suunniteltu pääasiassa helpottamaan erilaisten syötelaitteiden liittämistä ja niitä käyttävien sovellusten kehittämistä. Tähän arkkitehtuurit käyttävät lähinnä aikaisemmista vuorovaikutustavoista tuttuja tekniikoita. Alla vertailtavien piirteiden lisäksi arkkitehtuurin hyödyllisyyteen vaikuttavat paljon myös järjestelmälle asetetut reaaliaikaisuusvaatimukset, joita ei kuitenkaan tässä voida analysoida.

Edellä kuvatun perusteella DWARF-kehys soveltuu käytettäväksi silloin, kun täytyy kehittää uusia vuorovaikutustekniikoita, sillä kehys mahdollistaa kehittyneiden ominaisuuksien toteuttamisen ja on laajennettavissa komponenteilla. Se ei kuitenkaan tue käsinkosketeltavien käyttöliittymien ominaispiirteitä niin hyvin kuin varta vasten siihen tarkoitukseen suunnitellut järjestelmät. metaDESK, joka sekin toteuttaa hyvin vaativia ominaisuuksia, on puolestaan melko tiukasti sidottu omaan fyysiseen arkkitehtuuriinsa ja siten ei ole kovinkaan helposti käytettävissä erilaisten käsinkosketeltavien käyttöliittymien prototyyppien kehittämiseen. Valmiit laitetoteutukset sisältävä Papier-Mâché ja erilaisten laitteiden liittämisen mahdollistava iStuff ovat helposti laajennettavissa niiden omien tapahtumamallien päälle. Tapahtumiin perustuva vuorovaikutus ja kehittyneiden ominaisuuksien puuttuminen kuitenkin rajoittavat niiden tarjoamia mahdollisuuksia. Phidgets tarjoaa kehittäjien käyttöön valmiita syötelaitekokonaisuuksia, joita hyväksi käyttäen on mahdollista rakentaa kehittyneitäkin vuorovaikutustekniikoita, mutta jotka eivät kuitenkaan tue käsinkosketeltavien käyttöliittymien toteutusta laiterajapinnasta ylöspäin.

Arkkitehtuuriratkaisuja tulee käyttää käsinkosketeltavien käyttöliittymien toteutuksessa hyödyksi, sillä käsinkosketeltavuus asettaa järjestelmille vaatimuksia, joita aikaisemmat arkkitehtuurit eivät täytä. Olen esitellyt kuusi erilaista käsinkosketeltavien käyttöliittymien arkkitehtuuria ja niissä käytettyjä ratkaisuja. Tärkeimmät ratkaisumallit, joilla vaatimukseen on esitellyissä arkkitehtuureissa vastattu, ovat komponenttipohjaisuus, edustaja-suunnittelumalli ja tapahtumapohjaisuus. Ratkaisut tukevat hyvin sovellusten prototyyppien luomista ja erilaisten syöte- ja tulostuslaitteiden liittämistä, mutta puutteellisesti uudenlaisien vuorovaikutustekniikoiden toteuttamista.

Mitä uusia vuorovaikutustapoja arkkitehtuurien sitten tulisi tukea? Näyttäisi siltä, että valikko- ja komentopohjaisista käyttöliittymistä periytyvä tapahtumamalli jossain

määrin rajoittaa uusien vuorovaikutusten kehitystä tai ainakin ohjaa sitä tiettyyn suuntaan. Uusia arkkitehtuureja varten tulisi kehittää vuorovaikutukseen sellainen malli, joka mahdollistaa rinnakkaiset ja toisiinsa vaikuttavat komennot sekä tukee usean käyttäjän yhtäaikaista työskentelyä. Samalla mallin tulisi mahdollistaa reagoiminen käyttäjän toimiin ja helpottaa vaikeasti tulkittavien komentojen päättelemistä.

## 6. Käsinkosketeltavien käyttöliittymien suunnittelu ja prototyypointi

Käsinkosketeltavista käyttöliittymistä on kehitetty lukuisia toimivia prototyyppejä monille eri sovellusalueille. Johtuen osaltaan käsinkosketeltavuuden tarjoamista monipuolisista mahdollisuuksista, ovat sovellukset niin heterogeenisiä, että pelkästään aikaisempia sovelluksia tutkimalla on vaikea ymmärtää niitä suunnitteluratkaisuja, jotka vaikuttavat oleellisesti sovellusten hyödyllisyyteen. Monissa tutkimuksissa on kuitenkin käsitelty käsinkosketeltavien käyttöliittymien suunnittelua joko erillään sovelluksista tai sovellusten ohella.

Tässä luvussa pyrin kokoamaan tietämystä käsinkosketeltavien käyttöliittymien suunnittelusta. Esittelen sekä suunnitteluperiaatteita, jotka vaikuttavat tehtyjen suunnitteluratkaisujen taustalla että menetelmiä, joita on onnistuneesti käytetty käsinkosketeltavia käyttöliittymiä suunniteltaessa.

### 6.1. Suunnitteluperiaatteet

Klemmer ja muut [2006] esittelevät viisi teemaa, jotka heidän mielestään kuvastavat hyvin ruumiin ja fyysisen tekemisen osuutta vuorovaikutuksessa:

- Tekemällä ajatteluun liittyy tekemällä oppiminen, elehtiminen, episteemiset toiminnot, prototyypin rakentaminen sekä parhaan esitystavan valinta.
- Suorituskykyä ruumiillisissa toimissa parantavat toimintakeskeiset taidot, käsien rikas käyttäminen, motorisen muistin hyödyntäminen ja reflektion korvaaminen reflekseillä.
- Ryhmätyöskentelyä tukeva fyysisten toimintojen näkyvyys auttaa muita oppimaan toisia seuraamalla, auttaa koordinoimaan muiden tekemisiä, mahdollistaa esityksestä nauttimisen ja vaikeuttaa vilppiä.
- Fyysisiin toimintoihin liittyvä lisääntynyt riski voi olla avuksi esimerkiksi lisäämällä luottamusta muihin, vahvistamalla sitoutumista omiin valintoihin ja vastuun kantamista niistä sekä parantamalla tarkkaavaisuutta.
- Perinteistä fyysistä toimintaa ei tulisi korvata kokonaisuudessaan digitaalisilla järjestelmillä, vaan olemassa olevan rikkaan vuorovaikutuksen perusolemus tulisi säilyttää parantaen sitä digitaalisilla piirteillä.

Toisaalta Blackwell [2003] analysoi käsinkosketeltavia käyttöliittymiä niiden kognitiivisten ominaisuuksien perusteella. Hän esittelee ominaisuuksia, joiden mukaan käsinkosketeltavat käyttöliittymät

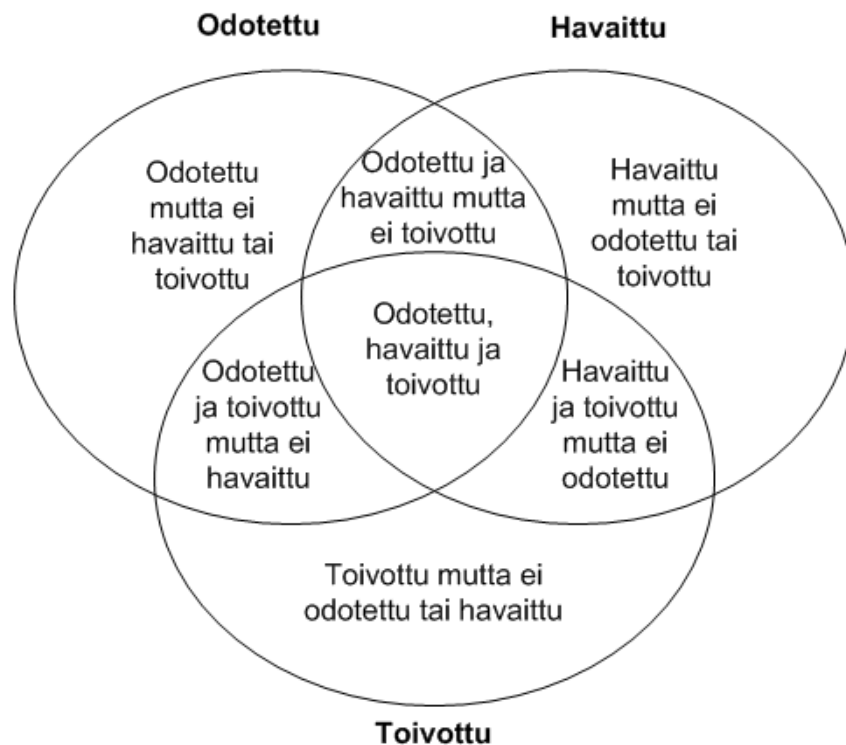
- ovat hetkellisiä eli minkä tahansa järjestelmään kuuluvan objektin liikuttaminen muuttaa järjestelmän merkitystä,
- käyttävät usein toista esityskerrosta, kuten esimerkiksi graafista näyttöä,
- vaativat paljon tilaa eivätkä ole skaalautuvia,

- ovat helposti muokattavissa, mutta ajallisiin esityksiin ei voi tehdä muutoksia toistamatta koko esitystä,
- mahdollistavat joskus muokkaamisen tavoilla, jotka eivät vaikuta järjestelmän tilaan, esimerkiksi kirjoittamalla objekteihin,
- eivät ole hyviä esittämään objektien välisiä liitoksia,
- mahdollistavat objektien aikaisemmin tunnettujen merkitysten hyväksikäytön,
- eivät yleensä tue ajallisten ja järjestykseen liittyvien tapahtumien tulkitsemista,
- eivät mahdollista helposti palautteen tuottamista,
- mahdollistavat nopeat ja helpot kokeilut, ja
- eivät tue abstraktien muuttujien ja funktioiden esittämistä aineellisilla objekteilla.

Kuten yllä olevista teemoista selviää, on suunnittelupäätösten vaikutusalue laaja. Niinpä en pyrikään esittämään kaikkia mahdollisia periaatteita, vaan olen poiminut tähän yhteyteen laajimmat löytämäni kokonaisuudet.

### 6.1.1. Tunnistus

Benford ja muut [2005] analysoivat käyttöliittymien ohjaamiseen tarkoitettuja liikkeitä sen mukaan ovatko ne odotettuja, havaittavissa tai haluttuja. Suunnittelukehys (Kuva 29) on tarkoitettu kaikenlaisten ympäristöään aistimaan kykenevien järjestelmien suunnitteluun.



Kuva 29. Malli odotetuista, havaituista ja halutuista liikkeistä [Benford *et al.*, 2005].

Käsinkosketeltaviin käyttöliittymiin kehys liittyy läheisesti, sillä ne käyttävät tunnistusteknologioita aineellisten objektien tunnistamiseen. Käsinkosketeltavaa käyttöliittymää suunniteltaessa on siis tarpeen miettiä mitä objektien liikkeitä halutaan tunnistaa, mitä liikkeitä pystytään tunnistamaan ja mitä liikkeitä käyttäjät tulevat yrittämään. Liikkeiden analysoinnissa on hyvä käyttää Benfordin ja muiden [2005] listausta liikkeiden ominaisuuksista, joita ovat liikkeen laajuus, vapausasteet (degrees-of-freedom), nopeus, tarkkuus, vakaus ja kesto.

Odotetut liikkeet ovat sellaisia liikkeitä, joita käyttäjät luonnostaan tekevät joutuessaan käyttöliittymän kanssa tekemisiin. Toisaalta myös vähemmän odotetut liikkeet voivat ovat merkityksellisiä suhteessa kyseiseen objektiin. Siksi odotettujen liikkeiden lisäksi on hyvä pyrkiä tunnistamaan myös täysin odottamattomia liikkeitä, jotka saattavat olla järjestelmän kannalta oleellisia.

Havaitut liikkeet ovat puolestaan sellaisia liikkeitä, jotka järjestelmä pystyy sensoreillaan tunnistamaan ja joihin se pystyy reagoimaan. Eri tunnistusteknologiat mahdollistavat erilaisten liikkeiden ja toimintojen tunnistamisen, mikä kannattaa ottaa huomioon teknologioita valitessa.

Haluttuja liikkeitä ovat sellaiset liikkeet, jotka ovat tarpeellisia järjestelmän käyttämisen kannalta. Ne eivät välttämättä ole täysin samoja kuin odotetut tai tunnistetut liikkeet. Vaikka olisi hyvä pystyä tunnistamaan kaikki halutut liikkeet, aina se ei ole mahdollista ja joudutaan tekemään kompromisseja.

Suunnittelukehyyksen käyttäjän tulee Benfordin ja muiden [2005] mukaan analysoida odotetut ja odottamattomat liikkeet, analysoida havaittavissa olevat liikkeet sekä huomioida tunnistusteknologioiden puutteet, analysoida halutut liikkeet ja lopuksi pohtia eri liikkeiden raja-alueita ja yhteneviä kohtia. Erilaisten liikekategorioiden rajat ovat otollista maastoa etsiä uusia ja innovatiivisia ratkaisumalleja käyttöliittymiin. Mikäli odotetut ja havaitut liikkeet eivät ole samoja, voidaan järjestelmää yrittää parantaa tehostamalla tunnistusta, rajoittamalla mahdollisia liikkeitä, osoittamalla käyttäjälle tunnistuskyvyn rajat tai antamalla käyttäjän itse oppia ne. Toisaalta, jos jokin liike tunnistetaan, vaikka sitä ei odotetakaan, täytyy järjestelmän käyttäytyä silti loogisesti.

Järjestelmät, jotka osaavat itse havaita käyttäjän toimia ja reagoida niihin, ovat nuori suuntaus käyttöliittymätutkimuksessa. Käsinkosketeltavat käyttöliittymät ovat osa tätä suuntausta ja siten niitä koskevat monilta osin samat suunnittelusäännöt kuin esimerkiksi puhe- tai elekkäyttöliittymiä, joissa vakiintuneita käytäntöjä ei vielä ole [Bellotti *et al.*, 2002]. Bellotti ja muut ovat ottaneet vaikutteita ihmisten välisestä kommunikaatiosta ja päätyneet luokittelemaan havaitsemista käytävien käyttöliittymien vuorovaikutusongelmat viiteen eri luokkaan.

Ensimmäinen Bellottin ja muiden luokista kuvaa ongelmia, jotka liittyvät siihen, miten käyttäjä tietää, kuinka järjestelmälle voi antaa komentoja. Perinteisiä järjestelmiä



käyttäessään käyttäjä tietää, että hiirtä ja näppäimistöä käyttäen hän pystyy suorittamaan toimintoja. Käsinkosketeltavissa käyttöliittymissä ei kuitenkaan ole vakiintuneita syötelaitteita. Käyttäjä ei esimerkiksi välttämättä tiedä mitkä aineelliset objektit liittyvät järjestelmään, eikä välttämättä ymmärrä eroa tunnisteita sisältävien objektien ja muiden objektien välillä.

Toinen ongelma liittyy läheisesti ensimmäiseen. Siinä pohditaan, kuinka käyttäjä tietää milloin järjestelmä ottaa vastaan ja kuuntelee annettuja komentoja. Graafisissa käyttöliittymissä näytöllä vilkkuva kursori ja muut graafiset efektit näyttävät käyttäjälle että järjestelmä kuuntelee. Käsinkosketeltavissa käyttöliittymissä ei tällaisia vakiintuneita käytäntöjä ole. Koska aineelliset objektit säilyttävät asemansa vaikka vuorovaikutteinen järjestelmä olisikin kytkeyty pois päältä, on ensiarvoisen tärkeää, että käyttäjä tietää järjestelmän tilan.

Käyttäjän täytyy myös tietää mitä toimintoja hänen tulee suorittaa saavuttaaksensa haluamansa tuloksen ja kuinka määrittää toimintansa kohteet. Graafisissa käyttöliittymissä käyttäjälle tarjotaan yleensä lista toimintoja, joista hän voi valita haluamansa. Käsinkosketeltavissa käyttöliittymissä käyttäjän tulee normaalisti objektien ulkonäöstä osata päätellä mahdolliset komennot ja niiden vaikutusalueet.

Bellottin ja muiden neljäs kysymys on tarkoitettu selvittämään, mistä käyttäjä tietää, että järjestelmä on ymmärtänyt hänen komentonsa ja suorittanut sen oikein. Graafisissa käyttöliittymissä palautetta järjestelmän toiminnasta antavat ruudulla näkyvät muutokset. Käsinkosketeltaviin käyttöliittymiin ei aina kuulu näyttöä ja palautetta pitäisi silti saada annettua. Aineellisten objektien liikuttaminen tietokoneohjatusti on vaikeaa, mutta siihen tulisi pyrkiä. Myös muiden modaaliteettien hyödyntämistä kannattaa harkita.

Viides kysymys liittyy siihen, miten käyttäjä huomaa virheet ja osaa toipua niistä. Graafisissa käyttöliittymissä tehtyjen komentojen kumoaminen ja toistaminen on helppoa, mikäli järjestelmä vain pitää niistä kirjaa. Käsinkosketeltavissa käyttöliittymissä käyttäjän yleensä pitää palauttaa tila itse. Kokeileminen on käsinkosketeltavissa käyttöliittymissä kuitenkin yleensä helppoa, ja siksi yksittäisten virheiden korjaaminenkin käy nopeasti.

### **6.1.2. Merkityksen esittäminen objektien suhteilla**

Taikuutta ja yliluonnollisia ilmiöitä on ehdotettu hyviksi metaforiksi käsinkosketeltaville käyttöliittymille [Svanaes and Verplank, 2000]. Vaikka jotkin ilmiöt (kuten telepatia ja taikaesineet) saattavatkin auttaa käyttäjää käsitteellisen mallin (conceptual model) muodostamisessa, niin yleisesti ottaen taikuus on huono metafora. Metaforien sijaan olisikin ehkä parempi pyrkiä luonnolliseen vuorovaikutukseen.

Luonnollisuuden taustalla vaikuttaa useita tekijöitä. Eräs niistä on Normanin [1990, ss. 9–12] yleiseen tietoisuuteen tuoma käsite affordanssi (affordance). Myöhemmin Norman [1999] on tarkentanut käsitettään havaituksi affordanssiksi (perceived

affordance). Erona näillä kahdella on se, että ensimmäinen tarkoittaa suhdetta, joka toimijalla ja kohteella on, jälkimmäinen tarkoittaessa vain sellaista suhdetta, jonka toimija havaitsee. Esimerkiksi ensimmäistä kertaa pankkiautomaatilla asioiva henkilö pystyy syöttämään pankkikortin sille tarkoitettuun aukkoon siinä missä kuka tahansa muukin. Eri asia on, havaitseeko hän sen olevan mahdollista tai pitääkö hän sitä hyödyllisenä tai merkityksellisenä.

Sekä todelliset että havaitut affordanssit ovat tärkeitä käsinkosketeltavien käyttöliittymien yhteydessä. Graafisia käyttöliittymiä suunniteltaessa suunnittelijoilla on käytössään vain havaitut affordanssit ja nekin vain osittain. Suunnittelija ei voi esimerkiksi vaikuttaa siihen, mitä käyttäjä voi tietokoneella tehdä: aina on mahdollista liikuttaa hiiren avulla osoitinta näytöllä ja painella hiiren painikkeita. Suunnittelija voi vain graafisin keinoin osoittaa missä klikkauksia kannattaisi tehdä. Käsinkosketeltavissa käyttöliittymissä suunnittelijan mahdollisuudet ovat paljon laajemmat. Palloa voi vierittää, suurennuslasin läpi voi katsoa, kahvasta voi kääntää ja kappaleen voi työntää itsensä muotoiseen aukkoon. Käytettävien objektien ulkonäkö siis kertoo miten niitä on mahdollista ja hyödyllistä käyttää.

Havaittuihin affordansseihin läheisesti liittyvä käsite on ennakkopalaute (feedforward) [Djajadiningrat *et al.*, 2002]. Kun havaituilla affordansseilla tarkoitetaan käyttäjän havaitsemia mahdollisia toimintoja, niin ennakkopalautteella tarkoitetaan sitä, kuinka käyttäjälle kommunikoidaan toimintojen tarkoitus. Djajadiningratin ja muiden [2002] mielestä toimintojen seurausten päättämiseen jo ennen niiden suoritusta tulisi olla jokin parempi keino kuin pelkkä metaforan käyttö. Normanin [1990, ss. 23–27] luonnollinen sidonta (natural mapping) soveltuu heidän mielestään tilanteisiin, joissa käyttöliittymän objekteilla on jokin tosielämän vastine. Abstraktimpiin käyttötarkoituksiin se soveltuu kuitenkin huonosti. Niinpä heidän mielestään objektien ulkoasun tulisi tuoda esiin toiminnon seurauksia. Djajadiningrat ja muut pitävät käsinkosketeltavuutta vaatimuksena ennakkopalautteen hyödyntämiselle.




---

Kuva 30. Esimerkki luonnollisesta ja tilallisesta sidonnasta rautatie-sovellusalueella [Djajadiningrat *et al.*, 2004].

Tilallinen sidonta (spatial mapping) on luonnollisen sidonnan erikoistapaus. Sharlinin ja muiden [2004] mukaan sillä tarkoitetaan tilallista suhdetta aineellisen objektin ja sen digitaalisen vastineen välillä. Parhaassa tapauksessa sovellusalue on luonnostaan tilallinen (Kuva 30). Mikäli näin ei ole, tulee tilallisen sidonnan hyödyntää ihmisten aikaisemmin oppimia konventioita.

Luonnollisuuteen liittyy myös Normanin [1990, ss. 82–86] esittämä luokittelu rajoitteista (constraints). Jos objektia yritetään työntää aukkoon, josta se ei mene läpi, estävät tämän silloin fyysiset (physical) rajoitteet. Niitäkään ei graafisen suunnittelijan käytössä ole; hän ei voi esimerkiksi mitenkään estää liikuttamasta hiirtä tietyn alueen ulkopuolella, kun taas käsinkosketeltavassa käyttöliittymässä tällainen suunnittelu on mahdollista. Muita rajoitetyyppejä – loogisia (logical) ja kulttuurisia (cultural) rajoitteita – voidaan käyttää myös graafisissa käyttöliittymissä. Fyysiset rajoitteet ovat kuitenkin kaikista tehokkaimpia, sillä toisin kuin loogisia ja kulttuurillisia rajoitteita, ei niitä voida kiertää [Norman, 1999].

Luonnollisesti toimivissa objekteissa objektin käyttämiseen tarvittava toiminta (action), vastavaikutus (reaction) sekä tulos (function) liittyvät toisiinsa. Wensveen ja muut [2004] jakavat toiminnan ja sen tuloksen välisen liitoksen kuuteen osatekijään. Osatekijät määrittävät, että toiminnan ja sen vastavaikutuksen

- välillä ei kulu aikaa,
- paikka on sama,
- suuntaus on sama,
- nopeus ja muut dynaamiset ominaisuudet vastaavat toisiaan,
- käyttämät modaliteetit ovat sopusoinnussa, ja
- ilmaisun jälki on samanlaista.

Elektronisissa järjestelmissä nämä osatekijät eivät aina kohtaa, sillä ohjelmallisesti on mahdollista suunnitella sidontoja, jotka rikkovat esiteltyjä sääntöjä. Käsinkosketeltavissa käyttöliittymissä on kuitenkin mahdollista seurata näitä sääntöjä, mikä mahdollistaa luonnollisen vuorovaikutuksen suunnittelun.

Wensveen ja muut [2004] jakavat palautteen kolmeen eri luokkaan. Toiminnallinen palaute (functional feedback) on sellaista palautetta, joka kuuluu järjestelmän varsinaiseen toiminnallisuuteen. He käyttävät esimerkkinä televisiota, jossa kuvan ilmestyminen ruudulle on toiminnallista palautetta. Lisätty palaute (augmented feedback) on puolestaan palautetta, joka on varten vastaan lisätty tuotteeseen. Esimerkkinä mainittakoon punaisen virtavalon syttyminen televisioon. Luontainen palaute (inherent feedback) taas on toiminnan luonnollisista seurauksista saatua palautetta. Television virtapainikkeen painamisen tapauksessa tällaista on napin painumisen tunteminen. Edellä esitellyin perustein Wensveen ja muut jakavat myös ennakkopalautteen toiminnalliseen, lisättyyn ja luonnolliseen ennakkopalautteeseen.

### 6.1.3. Merkityksen esittäminen objektien ominaisuuksilla

Tutkiessaan aineellisten objektien vaikutusta teollisessa suunnittelussa, Brereton ja McGarry [2000] havaitsivat, että objektien merkityksen tulkinta ja objektien käyttötapa riippuvat siitä ympäristöstä, jossa objekti on. He sanovat, että käsinkosketeltavia käyttöliittymiä suunniteltaessa täytyy tehdä kompromisseja yleiskäyttöisten sekä spesifisten objektien välillä.

Musikaalisia käyttöliittymiä suunnitellessaan Kaltenbrunner ja muut [2004] suunnittelivat haptisia ominaisuuksia musiikin tuotannossa käytettäville objekteille. Eriytyypisiä objekteja oli suunniteltu äänten tuottamista sekä muokkaamista varten ja objektit eivät sisältäneet elektroniikkaa, vaan niiden sijainti tunnistettiin konenäön avulla. Abstraktien objektien suunnittelussa Kaltenbrunner ja muut tunnistivat seuraavia objektien ominaisuuksia:

- Muoto on helposti tunnistettavissa sekä tunnustellen että katsoen, kunhan liian montaa erilaista muotoa ei käytetä.
- Väri on tunnistettavissa vain näköaistilla ja soveltuu siis huonosti käytettäväksi erottamaan objekteja käsinkosketeltavissa käyttöliittymissä.
- Koko soveltuu käytettäväksi joissakin tapauksissa, mutta sen käyttöä rajoittaa se, että erikokoisia objekteja täytyy kaikkia pystyä käsittelemään.
- Pintamateriaali voi olla joko kiinteässä aineessa oleva kaiverrus tai tunnistettavissa oleva materiaali.
- Aineet voivat erota ominaisuuksiltaan esimerkiksi painon tai lämpimyyden perusteella ja aineen ei tarvitse olla sama kuin pintamateriaalin.

Kaltenbrunner ja muut käyttivät myös arkipäiväisiä, ympäristöstä löytyviä objekteja hyväkseen. Niiden käyttämisessä täytyy kuitenkin huomata, että järjestelmästä täytyy tällöin ilmetä, mitä kaikkia objekteja se pystyy tunnistamaan.

Materiaaliset ominaisuudet, joita objektit ilmentävät, ovat usein staattisia eivätkä ole käyttäjän tai järjestelmän muokattavissa. Myös erilaisten materiaalien ominaisuuksien yhdistely on hankalaa, sillä tietyt ominaisuudet sulkevat toisia ominaisuuksia pois. Objektien pinnoilla voidaan kuitenkin esittää kuvioita, jolloin ne voivat edustaa sellaista informaatiota, joka ei erottamattomasti kuulu objektiin itseensä [Blackwell, 2003]. Johtuen muiden ominaisuuksien huonosta dynaamisesta muokattavuudesta, on tämä mahdollisuus tällä hetkellä hyödyllinen useissa käsinkosketeltavissa käyttöliittymissä.

Blackwellin [2003] mukaan objektien sijainti voi hyödyntää kolmea ulottuvuutta, mutta käytännössä objektit pysyvät paikoillaan vain tasoilla. Toisaalta objekteja voidaan pyörittää kaikkien kolmen akselin ympäri, jolloin niiden suunta on kolmiulotteisesti määrättävissä. Tähänkin liittyy kuitenkin rajoitus, sillä paikallaan objektit pysyvät yleensä vain tasaisten pintojensa päällä. Myös ajallisessa dimensiossa objektit ovat

melko rajoittuneita. Normaalisti ei ole mahdollista palata mihinkään aiemmin vallinneeseen tilaan automaattisesti, vaan tätä halutessaan käyttäjän on itse palautettava aiempi tila. Edge ja Blackwell [2006] suosittavatkin, että näytöllä esitetään animointeja ajan havainnollistamiseksi. Aineellisten objektien puutteista seuraa, että käsinkosketeltavat käyttöliittymät ovat nykyisellään pääosin syötelaitteita, eivätkä pysty dynaamisesti esittämään informaatiota käyttäjälle.

#### **6.1.4. Yritys ja erehtyminen**

Yrityksen ja erehtymisen kautta oppiminen on tärkeää ihmisen ja tietokoneen vuorovaikutuksessa. Hyvän käyttöliittymän tulisikin Sharlinin ja muiden [2004] mukaan tehdä erehtymisestä mahdollisimman vähän harmia aiheuttavaa. Graafisissa käyttöliittymissä tätä on yritetty saavuttaa perumiskomennoilla (undo), mutta ne ovat usein puutteellisia eivätkä toimi niin kuin käyttäjä olettaa.

Hyvin suunnitelluissa käsinkosketeltavissa käyttöliittymissä perumiskomentoja ei tarvita. Koska objektit on usein sidottu kukin yhteen toimintoon, voidaan toiminto perua yksinään tuota objektia käsittelemällä. Tällöin ei kaikkia sitä seuranneita toimintoja täydy perua, niin kuin graafisissa käyttöliittymissä usein täytyy. Suorittamassaan testissä Sharlin ja muut [2005] (kts. myös [Sharlin *et al.*, 2002]) saivat käyttäjiltä palautetta yrittämisen helpoudesta käsinkosketeltavissa käyttöliittymissä. Testihenkilön mukaan käsinkosketeltavalla käyttöliittymällä ”voi aina yrittää”, kun taas muutoin täytyy vain käyttää mielikuvitusta.

#### **6.1.5. Ryhmätyöskentely**

Hornecker [2005] suosittelee, että usean käyttäjän samanaikaiseen käyttöön tarkoitetut käsinkosketeltavat käyttöliittymät tulisi suunnitella siten, että ne ohjaisivat käyttäjiä jakamaan tehtäviä ja auttamaan toisiaan tai jopa edellyttäisivät sitä. Jos esimerkiksi käyttöliittymän koko on sellainen, että vain yksi ihminen kerrallaan mahtuu sitä käyttämään, ei edellytyksiä ryhmätyölle ole. Vastakkaisesti, jos tehtävän suorittamiseksi yhden käyttäjän tarvitsisi olla useassa eri paikassa yhtä aikaisesti, tarvitsee hän väkisin apua muilta käyttäjiltä. Käyttöliittymä pitäisi myös suunnitella niin, että käyttäjät sijoittuvat siten, että pystyvät tekemään yhteistyötä. Tällöin esimerkiksi asetelma, jossa kaikki käyttäjät ovat käyttöliittymän ympärillä, saattaa toimia. Ongelmana tosin tällöin on tekstin ja kuvakkeiden suunta, joiden ei tulisi suosia mitään erityistä asemaa käyttöliittymän ympärillä.

Käyttöliittymään pitäisi päästä käsiksi useasta eri pisteestä (multiple access points) samanaikaisesti siten, että kaikilla ryhmän jäsenillä on tasapuoliset mahdollisuudet osallistua. Tällöin yksittäinen henkilö ei voi ottaa koko järjestelmää hallintaansa. Eri käyttäjien tulisi myös pystyä toimimaan samanaikaisesti eri pisteistä käsin, jotta turhalta odottamiselta vältytään ja keskittyminen yhteisen ymmärryksen muodostamiseen helpottuu.

Horneckerin [2005] mukaan käsinkosketeltavien käyttöliittymien tulisi samanaikaisesti olla helposti opittavissa ja mahdollistaa eksperttien kehittyneet käyttötavat. Tällä tavoin eritasoiset käyttäjät voisivat käyttää yhdessä samaa käyttöliittymää. Käyttöliittymän tulisi myös hyödyntää käyttäjien aiempaa osaamista, silti asettamatta käyttäjiä eriarvoiseen asemaan heidän tietämyksensä perusteella.

#### 6.1.6. Minimalismi

Wren ja Reynolds [2004] argumentoivat minimalismin puolesta käsinkosketeltavissa käyttöliittymissä. Heidän mielestään toiminnon suorittamisessa pitäisi käyttää siinä normaalistikin käytettäviä objekteja ja digitaalisten ominaisuuksien tulisi huomaamattomasti tukea normaalia toimintaa. Wren ja Reynolds [2004] jakavat minimalismin kahteen osatekijään:

- Yksinkertaisuus (parsimony) tarkoittaa tässä sitä, että mahdollisimman vähän uusia käyttöliittymäelementtejä tai metaforia lisätään olemassa olevaan järjestelmään.
- Läpinäkyvyys (transparency) tarkoittaa digitaalisten ominaisuuksien lisäämistä olemassa olevaan järjestelmään siten, että mahdollisimman vähän kognitiivista lisätaakkaa aiheutuu käyttäjälle.

Esimerkkinä minimalistisesta suunnittelusta Wren ja Reynolds [2004] esittelevät käsinkosketeltavan käyttöliittymän go-peliin. Siinä pelaajat pelaavat go:tta aivan niin kuin normaalistikin, mutta järjestelmä tarkkailee pelin kulkua ja pystyy esimerkiksi tallentamaan pelitilanteen, näyttämään viimeisimmän siirron ja osoittamaan virheelliset siirrot. Minimalistisen suunnittelun ongelma on mielestäni kuitenkin se, että koska ylimääräisiä käyttöliittymäelementtejä ei ole haluttu tuoda järjestelmään, niin kaikki itse peliin liittymättömät toiminnot on vaikea toteuttaa. Esimerkiksi aiempaan pelitilanteeseen palaaminen on vaikeaa pelkän go-laudan ja pelinappuloiden avulla.

Yksinkertaisuuden merkitystä käyttöliittymien suunnittelussa painottavat myös Chang ja muut [2007], joiden mukaan käsinkosketeltavien käyttöliittymien pitäisi olla mahdollisimman yksinkertaisia, mutta samalla ilmaisuvoimaisempia kuin tavalliset painonappeja ja LED-valoja käyttävät suunnitteluratkaisut. Changin ja muiden mielestä yksinkertaisuuden asettamat rajoitukset voivat ruokkia mielikuvitusta ja siten edistää innovatiivisten ratkaisujen syntymistä. He ovat tiivistäneet minimalistisen suunnitteluprosessin kahteen vaiheeseen:

- Valitse suunnitteluongelmasi ratkaisuun mahdollisimman pieni määrä syöte- ja tulostuslaitteita, joita kutakin tyyppiä saa olla korkeintaan yksi. Käyttöliittymä voi esimerkiksi koostua yhdestä painikkeesta ja LED-valosta.
- Mieti miten voisit lisätä suunnitelmasi ilmaisuvoimaa mahdollisesti metaforaa käyttämällä tai korvaamalla tyypilliset painikkeet ja LED-valot muilla ratkaisuilla.

Esimerkkinä yhdestä prosessin avulla syntyneestä suunnitteluratkaisusta Chang ja muut esittelevät puhelinvastaajan suunnitelman. Ensimmäinen suunnitteluvaiheen jälkeinen versio oli tyypillinen laatikko, jonka päällä oli LED-valo ilmaisemassa saapuneista viesteistä ja painike viestien kuuntelemista varten. Sovellettuun suunnitelmansa toisen suunnitteluvaiheen mukaiseksi, tuli puhelinvastaajasta lopulta ruukkumaisella alustalla sijaitseva saapuneiden viestien määrän mukaan kasvava kupla, josta viestejä voidaan puristaa ulos kuplaa alaspäin painamalla. Vaikka syntyneen ratkaisun parempaa käytettävyyttä ei olekaan osoitettu, on se osoitus siitä, kuinka vaatimus yksinkertaisuudesta voi ohjata suunnittelijoita uudentlaisiin ratkaisuihin.

## 6.2. Prototypointimetodit

Jensen ja muut [2005] ovat teollisissa oloissa tapahtuvaa fyysistä vuorovaikutusta tutkiessaan huomanneet, että ihmisten monipuolisista taidoista käsitellä aineellisia objekteja hyödynnetään vain pientä osaa. He ovat tutkineet ihmisten liikkeitä ja ovat sitä mieltä, että käsinkosketeltavissa käyttöliittymissä käytettävien liikkeiden tulisi sopia ihmisten taitoihin, sallia taitojen kehittäminen, tukea ryhmätyötä, sopia kuhunkin ympäristöön ja olla miellyttäviä tai jopa taiteellisia. Liikkeiden tulisi myös olla paljon nykyistä painikkeiden painamista rikkaampia. Tätä rikkautta voidaan yrittää parantaa käyttämällä erilaisia prototypointimenetelmiä.

Kranz ja Schmidt [2005] jakavat eniten käytetyt prototyypitekniikat neljään luokkaan:

- Paperiprototyyppejä käytetään suunnittelun apuna ja yksinkertaisiin käyttäjätesteihin.
- Prototyypimalleilla (mock-up) tarkoitetaan ruudunkaappauskuvia tai näyttöjä, joilla on vain rajoitetusti vuorovaikutusta.
- Ozin velho -prototyypissä käyttäjä käyttää järjestelmää tietämättä, että vuorovaikutteisia osia ohjaa ihminen tietokoneen sijasta.
- Videoprototyypin avulla voidaan visualisoida vuorovaikutteisen järjestelmän toimintaa animoimalla prototyypin avulla käyttöliittymän osien toimintaa.

Kranzin ja Schmidin luokat eivät kovinkaan hyvin mahdollista käsinkosketeltavien käyttöliittymien prototypointia. Tässä kohdassa olen jakanut käsinkosketeltavien käyttöliittymien prototyypit seuraaviin luokkiin: skenaariot ja videoprototyypit, karkeat prototyypit sekä täsmäprototyypit.

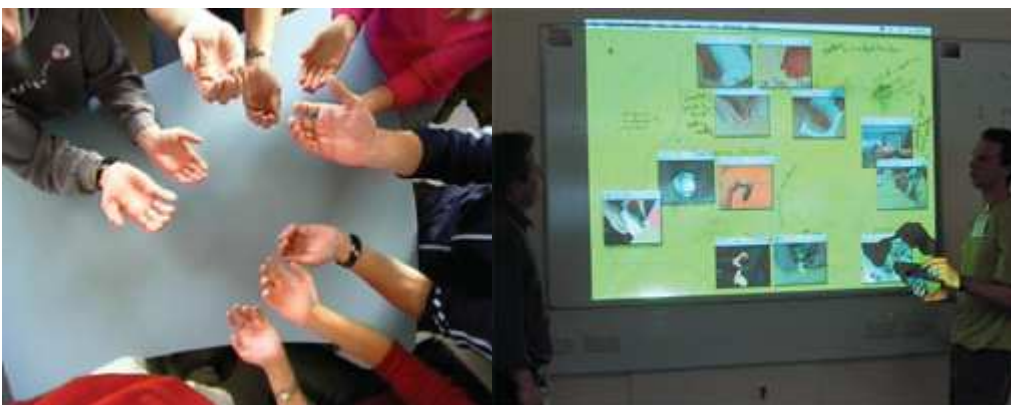
Käsinkosketeltavien käyttöliittymien prototypointi on usein tyypiltään iteratiivista. Stringer ja muut [2005] ovat käyttäneet lähestymistapaa, jossa kehitettävää järjestelmää testataan säännöllisin väliajoin oikeassa käyttöympäristössä siten, että siirrytään aina asteittain teknisesti monimutkaisempaan prototyyppiin. Heidän tutkimuksensa alkoi tarkkailuvaiheella, jossa tutkija vietti kolme viikkoa kohdeympäristössä tarkkaillen tulevien loppukäyttäjien toimintaa siihen mahdollisimman vähän puuttuen.

Tarkkailujakson tuloksena tutkijat olivat paikallistaneet vaihtoehtoisia tapoja, joilla he voisivat käsinkosketeltavan käyttöliittymän avulla kehittää kohderyhmän toimintaa.

Tarkkailuvaihetta seurasivat iteraatiokierrokset. Ensimmäisellä iteraatiokierroksella tutkijat kehittivät ideoiden pohjalta käsityömateriaaleista kokeellisia malleja ideoiden testaamiseksi. Toisella iteraatiokierroksella he testasivat paperiprototyypin avulla, kuinka kohderyhmä työskentelee aihealueen parissa ilman käsinkosketeltavan käyttöliittymän tukea ja kehittivät tarkkailun pohjalta mallin, joka kuvaa kohdeprosessia. Kolmannella ja neljännellä iteraatiokierroksella tutkijat antoivat koehenkilöiden käyttöön karkeita prototyyppejä, jotka eivät vielä sisältäneet vuorovaikutteisia piirteitä. Näiden mallien avulla tutkijat huomasivat suunnitelmassaan puutteita, jotka saatiin korjattua ennen varsinaista prototyyppien toteutusta. Viidennestä kierroksesta eteenpäin koehenkilöille annettiin käyttöön asteittain kehittyviä käsinkosketeltavia käyttöliittymiä, alkaen passiivisista karkeista prototyypeistä, jatkuen Ozin velho -prototyypeillä ja päätyen aina teknologisesti kehittyneisiin, vuorovaikutteisiin täsmäprototyyppeihin, joita testattiin oikeassa käytössä.

### 6.2.1. Skenaariot ja videoprototyypit

Näytellyt skenaariot ja roolileikkitekniikat (kts. [Boess *et al.*, 2007]) kuuluvat usein osaksi suunnittelua tuotekehitysprosesseissa. Jensenin ja muiden [2005] mukaan erityisesti teollisen suunnittelun piirissä lavastettujen tilanteiden näytteleminen on korvannut tekstipohjaisia kuvauksia vuorovaikutuksen suunnittelussa. Koska käsinkosketeltavissa käyttöliittymissä tarvitsee kiinnittää erityistä huomiota käsien toimintaan, on tätä tarkoitusta varten kehitetty käsiskenaariotekniikka (hands-only scenario) [Buur *et al.*, 2004; Jensen *et al.*, 2005] täydentämään muita skenaariotekniikoita. Käsiskenaarioiden avulla on mahdollista suunnitella tuotteen käsittelyyn ja käyttämiseen tarvittavat käsien liikkeet ennen varsinaisen tuotteen ulkomuodon suunnittelua. Käsiskenaarioissa (Kuva 31) pelkästään kädet suorittavat manipulointiin tarvittavia liikkeitä.



Kuva 31. Käsiskenaario- ja videoseinätekniikat käytössä [Buur *et al.*, 2004].



Käsiskenaario-tekniikkaa saadaan Jensenin ja muiden kokemusten mukaan tarkemmaksi, mikäli skenaarioon osallistuvia henkilöitä pyydetään suorittamaan liikkeet yhtäaikaaisesti ja liikkeet videoidaan. Tekniikan tarkoituksena on auttaa suunnitteluryhmää ymmärtämään erilaisia käyttäjien liikkeitä ja toimia sekä suunnittelemaan rikkaat liikkeet mahdollistavaa vuorovaikutusta.

Videon käyttö on suositeltavaa käsinkosketeltavia käyttöliittymiä suunniteltaessa, sillä sen avulla pystytään kuvaamaan monimutkaisia ja pitkäkestoisia toimintasarjoja oikeassa kontekstissaan. Videota voidaan käyttää esimerkiksi vuorovaikutustapojen suunnitteluun ja esittelemiseen ennen varsinaisten prototyyppien rakentamista [Mazé and Bueno, 2002], ihmisen toiminnan analysointiin ja vertailuun [Buur *et al.*, 2004; Jensen *et al.*, 2005] tai perinteisellä tavalla käyttäjätestien tallentamiseen.

Videseinätekniikassa [Buur *et al.*, 2004; Jensen *et al.*, 2005] kankaalle tai näytölle heijastetaan rinnakkain toistakymmentä keskustelukohteena olevaa videonpätkää toimintaa suorittavien ihmisten liikkeistä (Kuva 31). Videot ovat käynnissä samanaikaisesti, toistuvat aina alusta niiden loputtua ja niitä voidaan liikutella alueella vapaasti. Tämä asetelma mahdollistaa liikkeiden vertailun keskenään, sillä liikkeitä voidaan ryhmitellä ja niihin voidaan liittää tekstikommentteja. Tekniikan tarkoitus on edistää toimivamman vuorovaikutuksen suunnittelua rohkaisemalla suunnittelijoita ja käyttäjiä keskustelemaan toimista ja liikkeistä.

### 6.2.2. Karkeat prototyypit

Karkeilla prototyypeillä (low fidelity prototypes) tarkoitetaan tässä sellaisia prototyyppejä, jotka eivät sisällä elektroniikkaa tai muunlaista kehittyntä teknologiaa, vaan mahdollisia vuorovaikutteisia piirteitä kontrolloivi ihmisoperaattori.

Kolmiulotteiset mallit ovat paperiluonnoksia parempia käsinkosketeltavien käyttöliittymien konseptisuunnittelussa, sillä niiden avulla on helpompaa hahmottaa käyttöliittymän suhdetta dataan sekä mahdollisia käytettävyyshäkökohtia [Blackwell *et al.*, 2005]. Karkeiden prototyyppien on myös havaittu helpottavan näyttelijöiden toimintaa skenaariotekniikoissa [Boess *et al.*, 2007]. Yksinkertaisenkin prototyypin avulla on mahdollista saada konkreettisuutta suunnitteluun pohtimalla esimerkiksi sitä, miten objektit voivat liittyä toisiinsa ja niitä ympäröivään tilaan, miten tietokone tulkitsee tietoa prototyypistä, miten objektien ulkonäkö kertoo niiden toiminnallisuudesta ja miten prototyypin ominaisuudet ohjaavat käyttämään sitä oikein [Blackwell *et al.*, 2005]. Niiden avulla voidaan myös testata osaavatko ihmiset käyttää prototyyppejä. Muovailuvahan tai muiden samankaltaisten askarteluvälineiden käyttö prototyyppien valmistamisessa on suotavaa (Kuva 32).



Kuva 32. Esimerkkejä karkeista prototyypeistä [Blackwell *et al.*, 2005].

Ozin velho -tekniikka on käsinkosketeltavien käyttöliittymien tapauksessa hyödyllinen tekniikka, sillä toimivien elektronisten järjestelmien kehittäminen on tällä hetkellä vielä vaikeaa, ja siksi toiminnallisuutta olisi hyvä päästä testaamaan jo ennen sitoutumista tiettyihin toteutustekniikoihin.

Ihmisoperaattorin on helpompi simuloida joitakin käsinkosketeltavien käyttöliittymien ominaisuuksia kuin toisia. Esimerkiksi käsinkosketeltavaa palautetta (kuten aineellisten objektien liikettä) on vaikeaa, ellei jopa mahdotonta, tuottaa koehenkilön sitä huomaamatta. Toisaalta käsinkosketeltavan syötelaitteen tuottaessa palautetta näytölle, on operaattorin helppo katsoa koehenkilön suorittamat toiminnot ja kutsua vastaavia funktioita tietokoneelta sopivan palautteen tuottamiseksi näytölle. Näin menettelivät Andersson ja muut [2002] testatessaan käsinkosketeltavaa nukkekäyttöliittymää. Heidän tapauksessaan Ozin velho -testauksen tarkoituksena oli selvittää käyttötarkoitukseen parhaiten soveltuva nukke, sekä löytää luontaisia liikesarjoja, joita ihmiset nukella tuottavat. Testissä käyttäjillä oli käsissään nukke, jonka kerrottiin sisältävän liikkeitä tunnistavia sensoreita. Nuken lisäksi koehenkilöiden edessä oli näyttö, jolla näkyi ihmishahmo. Koehenkilöille kerrottiin, että nuken liikkeiden avulla he pystyvät ohjaamaan tätä hahmoa. Todellisuudessa kuitenkin hahmoa ohjasi ihmisoperaattori, joka pystyi nappeja painamalla pistämään hahmon ilmentämään jotakin tunnetilaa tai toimintaa. Testin avulla tutkijat pystyivät kehittämään paremman nuken, suunnittelemaan sensorit siten, että ne pystyvät erottamaan tarvittavat tunnetilat ja suunnittelemaan pelissä tarvittavat tunnetilat sen mukaan, kuinka helposti ne ovat ilmaistavissa ja tunnistettavissa.

### 6.2.3. Täsmäprototyypit

Täsmäprototyypin (high fidelity prototypes) valmistaminen, – eli ”luonnostelu laitteilla” [Holmquist, 2006] (Kuva 33) – on tarpeellista ennen valmiin tuotteen toteutusta. Tällaisten elektroniikkaa sisältävien prototyypin valmistamista helpottamaan on olemassa valmiita elektronisia komponentteja tai ohjelmistoja kuten Phidgets [Greenberg and Fitchett, 2001], Smart-Its [Gellersen *et al.*, 2004], Calder [Lee

*et al.*, 2004], Particles [Decker *et al.*, 2005], iStuff [Ballagas *et al.*, 2003], Papier-Mâché [Klemmer *et al.*, 2004], Switcharoo [Avrahami and Hudson, 2002], BOXES [Hudson and Mankoff, 2006], d.tools [Hartmann, 2006], CookieFlavors [Kimura *et al.*, 2006], Voodoo IO [Villar and Gellersen, 2007] ja VoodooFlash [Spiessl *et al.*, 2007]. Näiden työkalujen yhteinen tarkoitus on helpottaa suunnittelijoiden työtä tarjoamalla jonkinlainen valmis kerros, jonka päälle toteuttamista voidaan jatkaa. Avrahami ja Hudson perustelevat Switcharoo-työkalunsa tarpeellisuutta sillä, että sen käyttäminen helpottaa ulkoasun ja muodon suunnittelua samanaikaisesti vuorovaikutuksen kanssa, auttaa suunnittelijoita havaitsemaan ongelmakohdat ajoissa, auttaa loppukäyttäjiä olemaan mukana suunnitteluprosessissa ja avustaa loppukäyttäjiä toiveiden ilmaisemisessa.



Kuva 33. Esimerkkejä työkaluilla kehitetyistä prototyypeistä BOXES [Hudson and Mankoff, 2006], d.tools [Hartmann, 2006] ja Calder [Lee *et al.*, 2004].

Erilainen tapa testata käsinkosketeltavien käyttöliittymien vuorovaikutteisia ominaisuuksia on käyttää lisätyn todellisuuden mahdollisuuksia. Tekniikassa on kyse siitä, että kameran avulla tarkkaillaan optisten merkkien sijaintia ja rotaatiota kolmiulotteisesti (Kuva 34).



Kuva 34. Esimerkki lisättyä todellisuutta hyödyntävästä käsinkosketeltavasta käyttöliittymäprototyypistä [Hornecker and Psik, 2005].

Vaikka lisätty todellisuus ei tuokaan esiin kaikkea sitä rikkautta, jota muotoiltujen aineellisten objektien käsittelyyn liittyy, voidaan sen avulla testata erityisesti

vuorovaikutukseen liittyviä ominaisuuksia. Hornecker ja Psik [2005] puoltavat tätä tekniikkaa sillä perusteella, että se auttaa suunnittelijoita valitsemaan varsinaisen toteutustekniikan mahdollisimman myöhäisessä vaiheessa ja mahdollistaa erilaisten vaihtoehtojen testauksen ilman ongelmia laitteiden kanssa.

Lisätyn todellisuuden käytöllä käsinkosketeltavien käyttöliittymien prototypoinnissa on sekä hyviä että huonoja puolia. On olemassa valmiita ja testattuja kirjastoja, joiden avulla kehittäminen on halpaa, nopeaa ja helposti opittavissa. Konenäön avulla tapahtuva tunnistus on myös nopeaa ja antaa monipuolista informaatiota ilman ylimääräisiä johtoja. Ongelmina ovat yleensä huono tunnistuskyky puutteellisissa valaistusolosuhteissa, kameroiden näkökentän peittyminen ihmisten liikkuaessa sekä tarve sille, että merkit ovat koko ajan kameran näkökentässä, jotta ne pystytään tunnistamaan.

### **6.3. Yhteenveto**

Käsinkosketeltavien käyttöliittymien suunnittelu on vaativaa, sillä siihen liittyy niin paljon erilaisia vaihtoehtoja ja mahdollisuuksia. Toisaalta tämä on suuri voimavara, sillä täysin uudenlaisten ratkaisujen löytyminen vanhoihin ongelmiin on mahdollista. Sovellusten suunnittelu täytyy kuitenkin yleensä aloittaa ruohonjuuritasolta, sillä vakiintuneita käytäntöjä ei ole. Myös täysin käyttötarkoitukseensa soveltumattomiin ratkaisuihin päätyminen on helppoa.

Suunnittelijan tehtävää helpottavat erilaiset ohjeistukset hyvistä suunnitteluperiaatteista. Käsinkosketeltavien käyttöliittymien saralla periaatteet ovat kuitenkin vielä kehitysvaiheessa ja kattavat vain pienen osan suunnittelijan vastuulla olevista päätöksistä. Periaatteet ovat myös usein melko abstrakteja, eivätkä vastaa yksityiskohtaisiin toteutusongelmiin.

Suunnittelun vaikeudesta johtuen on erityisen tärkeää, että jo aikaisessa kehitysvaiheessa sovellusta testataan prototyyppien avulla. Tällöin on mahdollista saada käytännön kokemuksia järjestelmän käytöstä, jolloin hyvät ja huonot ratkaisut pystytään tunnistamaan mahdollisimman aikaisessa vaiheessa. Prototyypit voivat vaihdella yksinkertaisesta muoviluvahamallista täysin toimivaan elektroniseen kokoonpanoon. Tärkeintä on, että prototyyppi on käyttötarkoitukseensa sopiva ja antaa suunnittelijoille lisää tietoa oikeiden ratkaisujen tekemiseen.

## **7. Prototyypisovelluksen testaus**

Tähän asti tässä tutkimuksessa olen esittänyt tietämystä siitä, mitä käsinkosketeltavat käyttöliittymät ovat ja kuinka niitä tulisi suunnitella. Tässä luvussa menen konkreettisemmalle tasolle. Kuvaan kahden käsinkosketeltavan käyttöliittymän prototyypin toteutuksen ja testaan niiden soveltuvuutta tuotantolinjan ohjausjärjestelmän testaukseen.

Tutkimuksen tässä osassa on piirteitä van den Hovenin ja muiden [2007] määrittelemästä suunnittelun kautta tapahtuvasta tutkimuksesta (research through design), jota he suosittelevat käytettäväksi käsinkosketeltavien käyttöliittymien tutkimuksessa. Erona perinteiseen suunnitteluun on se, että prosessin tarkoitus ei ole saada aikaan tuotetta, vaan tietämystä suunnittelusta tuotteesta ja sen suunnitteluprosessista. Monista muista tutkimusmenetelmistä se puolestaan eroaa tilannekohtaisuutensa osalta; sille on ominaista, että suunnittelijan taidot ovat oleellinen osa prosessia ja että tulokset ovat yleistettävissä vain kyseisen tilanteen laajuudelle.

Luku jakaantuu karkeasti kahteen osaan. Ensimmäisen osan muodostaa luvun ensimmäinen kohta, jossa kuvaan suunnitteluprosessin tuloksena syntyneet karkeat prototyypit. Samalla esittelen sovellukselle suunnitellun toimintaympäristön, toiminnan yleisellä tasolla ja korkean tason arkkitehtuurin, joka mahdollistaa käsinkosketeltavan käyttöliittymän kytkemisen varsinaiseen ohjausjärjestelmään testauksen suorittamista varten. Esittelen myös kaksi vaihtoehtoista vuorovaikutustapaa tuotantolinjaa simuloivalle käyttöliittymälle. Luvun toinen osa koostuu kolmesta seuraavasta kohdasta, joissa esittelen testauksen, jonka suoritin käyttäen karkeita prototyyppejä ja Ozin velho -menetelmää. Aluksi esittelen käyttämäni testausjärjestelyn, jonka jälkeen käyn läpi testin tulokset. Lopuksi arvioin testattujen käyttöliittymien toimivuutta ja hyödyllisyyttä testin tulosten sekä yleisesti tunnettujen käsinkosketeltavien käyttöliittymien ominaisuuksien valossa.

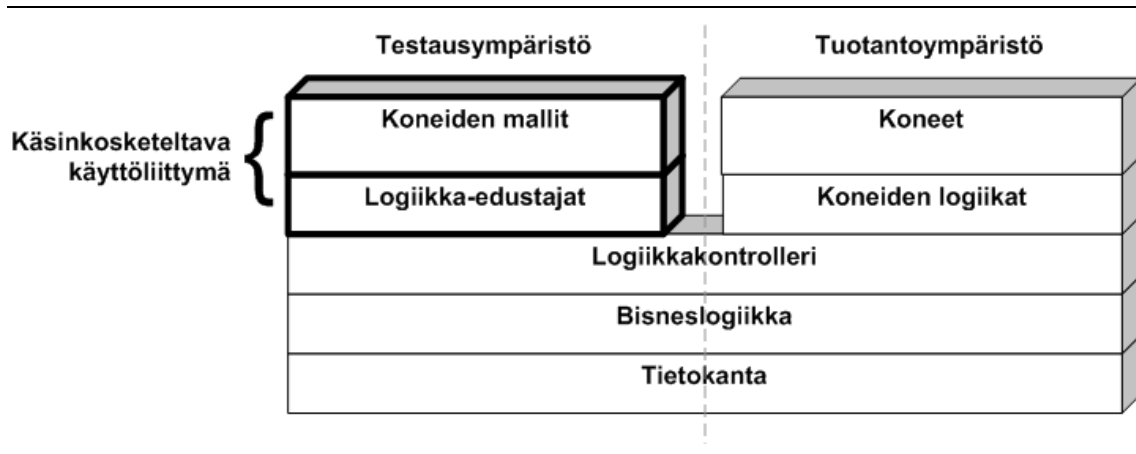
### **7.1. Sovelluksen kuvaus**

Testattavat prototyypit edustavat sovellusta, joka on tarkoitettu sovelluskehittäjien ja heidän muodostamiensa pienten ryhmien käytettäväksi. Sovelluksen avulla on tarkoitus simuloida tuotantolinjan toimintaa testauksen helpottamiseksi.

#### **7.1.1. Arkkitehtuuri**

Sovelluksen ohjelmistoarkkitehtuuri on suunniteltu siten, että mahdollisimman vähän muutoksia tarvitaan muodostettaessa uutta tuotantolinjaa tai siirryttäessä testausympäristöstä tuotantoympäristöön. Kuvassa 35 on esitetty sovelluksen arkkitehtuuri korkealla tasolla. Sovellukseen kuuluvan tietokannan rakenne on sama kummassakin ympäristössä. Myös järjestelmän varsinaisen toiminnallisuuden sisältävä

bisneslogiikkakerros säilyy samana. Ylin yhteinen kerros on logiikkakontrollerikerros, joka yhdistää koneilta tulevat syötteen järjestelmän ymmärtämään muotoon.

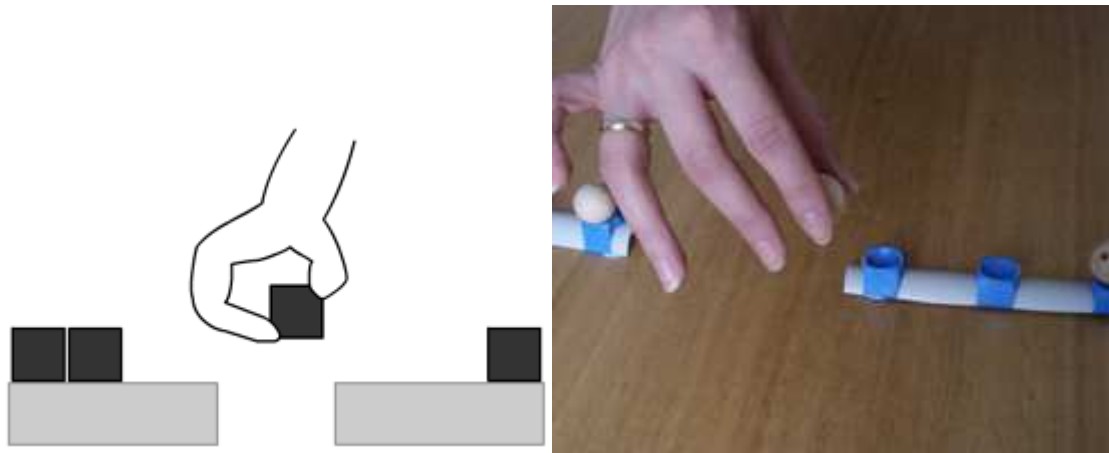


Kuva 35. Järjestelmän korkean tason arkkitehtuuri.

Yhteisten kerrosten yläpuolelle tulevat tuotantoympäristössä linjan oikeat koneet ja niiden omat logiikat. Testausympäristössä nämä koneet korvataan käsinkosketeltavalla käyttöliittymällä, eli aineellisilla objekteilla sekä niihin liittyvillä ohjelmistoedustajilla. Käyttöliittymän aineellinen osuus muodostuu siis linjan koneita kuvaavista kosketeltavista objekteista, sekä linjalla kulkevia tuotteita kuvaavista kappaleista, joiden liikkeet tunnistetaan sensoreiden avulla. Käsinkosketeltavien käyttöliittymien arkkitehtuureja vertailtaessa tunnistettu edustaja-malli toteutetaan sovelluksessa logiikka-edustajilla, joiden tehtävinä on jäljitellä oikeiden koneiden logiikoiden toimintaa sensoreilta saamiensa syötteiden mukaisesti. Näin saavutetaan arkkitehtuuri, joka on muunnettavissa vastaamaan minkä tahansa tuotantolinjan toimintaa.

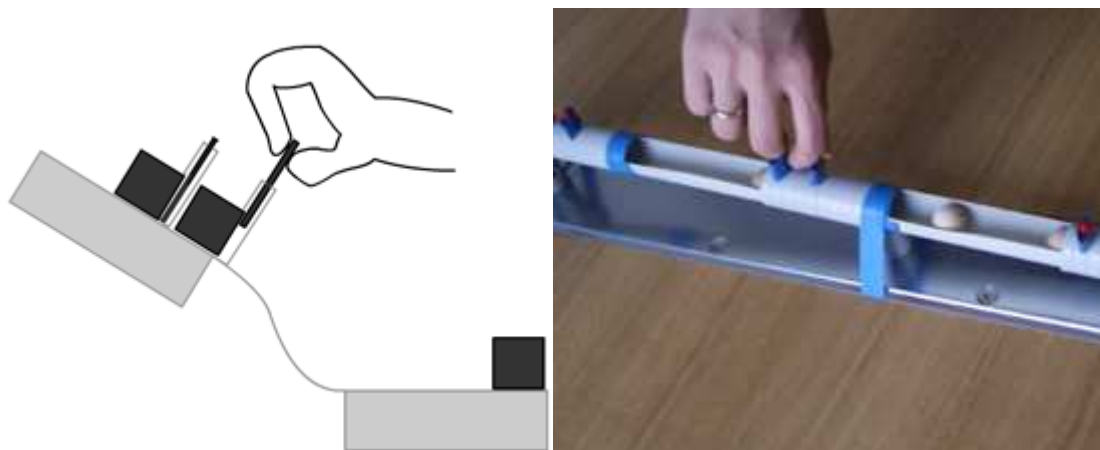
### 7.1.2. Vuorovaikutukset

Käyttäjätestausta varten rakensin kaksi karkeaa prototyyppiä, jotka mahdollistivat testauksen suorittamisen Ozin velho -menetelmää käyttäen. Tutkimuksen alussa esitelyjen systeemidynamiikan sovellusten [Zuckerman and Resnick, 2003; Zuckerman *et al.*, 2005] lailla myös prototyypeissä on mahdollista yhdistellä osia vastaamaan erilaisia linjakokonaisuuksia. Kummassakin prototyypissä käytin vuorovaikutuksien suunnitteluun erilaista lähestymistapaa. Molemmat lähestymistavat perustuvat tuotantolinjan henkilöstön toimintaa tarkkaillessa tehtyihin havaintoihin sekä innoittaviin sovelluksiin.



Kuva 36. Kappaleiden siirtäminen käsin: periaate (vasemmalla) ja prototyyppi (oikealla).

Ensimmäisessä vuorovaikutustavassa kappaleen liikuttaminen koneelta toiselle tapahtuu käsin kappaletta siirtämällä (Kuva 36), vastaavasti kuten saapuneita viestejä käsitellään Marble Answering Machinessa [Poynor, 1995]. Usein ongelmatilanteiden sattuessa oikeata tuotantolinjaa valvova henkilöstö joutuu puuttumaan kappaleiden liikkumiseen siirtämällä pudonneet tuotteet takaisin linjalle tai palauttamalla tuotteet takaisin edeltävään vaiheeseen. Vaihtoehtoisesti voidaan myös ajatella, että käsi vastaa robottia, joka nostaa tuotteen koneelta toiselle. Tästä vuorovaikutustavasta käytän jatkossa nimitystä siirtäminen.



Kuva 37. Kappaleiden siirtäminen liu'uttamalla: periaate (vasemmalla) ja prototyyppi (oikealla).

Toisessa vuorovaikutustavassa ei käsitellä suoraan kappaleita, vaan siinä käytetään koneita itseään ohjaamaan linjan toimintaa. Vastaavaa tapaa tuotantolinjaa valvova henkilöstö käyttää esimerkiksi koneen uudelleen käynnistämiseen pysähdyksen jälkeen sekä automaattisten koneiden käsikäyttöiseen ohjaamiseen. Käsinkosketeltavassa käyttöliittymässä toteutin tämän siirtotavan niin, että koneilta kappaleita päästettäessä

ne liukuvat kaltevaa kourua pitkin seuraavalle koneelle asti (Kuva 37), hieman samalla tavalla kuin uudet viestit saapuvat Marble Answering Machinessa [Poynor, 1995]. Kappaleen siirtämiseksi nostetaan peräkkäin kahta lukitusvipua, jolloin yksittäinen kappale liukuu seuraavalle lukituspisteelle asti. Vuorovaikutustapa on myös esimerkki yksinkertaisen aktiivisen palautteen hyödyntämisestä käsinkosketeltavassa käyttöliittymässä. Tästä vuorovaikutustavasta käytän jatkossa nimitystä liu'uttaminen.

Testin tulosten analysoinnin näkökulmasta vuorovaikutustavat eroavat toisistaan ensinnäkin sen suhteen, kuinka paljon koehenkilö saa suorituksestaan kinesteettistä palautetta ja toisaalta sen suhteen, kuinka vaativaa liikuttaminen motorisesti on. Käsien siirtäessään henkilön voidaan olettaa saavan kinesteettistä palautetta siitä, mihin kohtaan linjaa hän on kappaleita siirtänyt, kun taas liu'utettaessa palaute on luultavasti vähäisempää. Toisaalta yksinkertaisempi liu'uttamisliike ja sulkimen liikkeistä saatava haptinen palaute saattaisivat mahdollistaa paremmin katseen pitämisen poissa käyttöliittymästä, mikä puolestaan saattaisi helpottaa testausvirheiden tarkkailua. Testasin erilaisten vuorovaikutustapojen toimintaa käytännössä seuraavaksi kuvattavalla tavalla.

## **7.2. Testin järjestelyt**

### **7.2.1. Testihenkilöt**

Testiin osallistui viisi henkilöä. Henkilöistä kolme oli naisia ja kaksi oli miehiä. Nuorin henkilöistä oli 24 vuotta ja vanhin 56 vuotta. Kaikilla kokeeseen osallistuneilla henkilöillä oli runsaasti aikaisempaa kokemusta tietokoneen käyttämisestä ja kaikki osallistujat olivat oikeakätisiä. Kukaan kokeeseen osallistuneista henkilöistä ei työskennellyt ohjelmistoalalla.

### **7.2.2. Laitteisto**

Testissä vertailin kolmen erilaisen vuorovaikutustavan toimivuutta yksinkertaisessa tuotantolinjan testaustilanteessa. Ensimmäisessä käyttötavassa kutakin kolmea konetta vastasi teline, johon testihenkilö saattoi asettaa enintään kolme puukuulaa (Kuva 38). Aloitustilanteessa puukuulat olivat pöydällä, mistä testihenkilö saattoi siirtää ensimmäisen puukuulan ensimmäiseen telineeseen. Testihenkilöt saivat vapaasti päättää, mihin kohtaan telinettä kuulia asettivat. Kuulan käsitettiin olevan koneella aina, kun se oli jossakin telineen kolmesta paikasta.



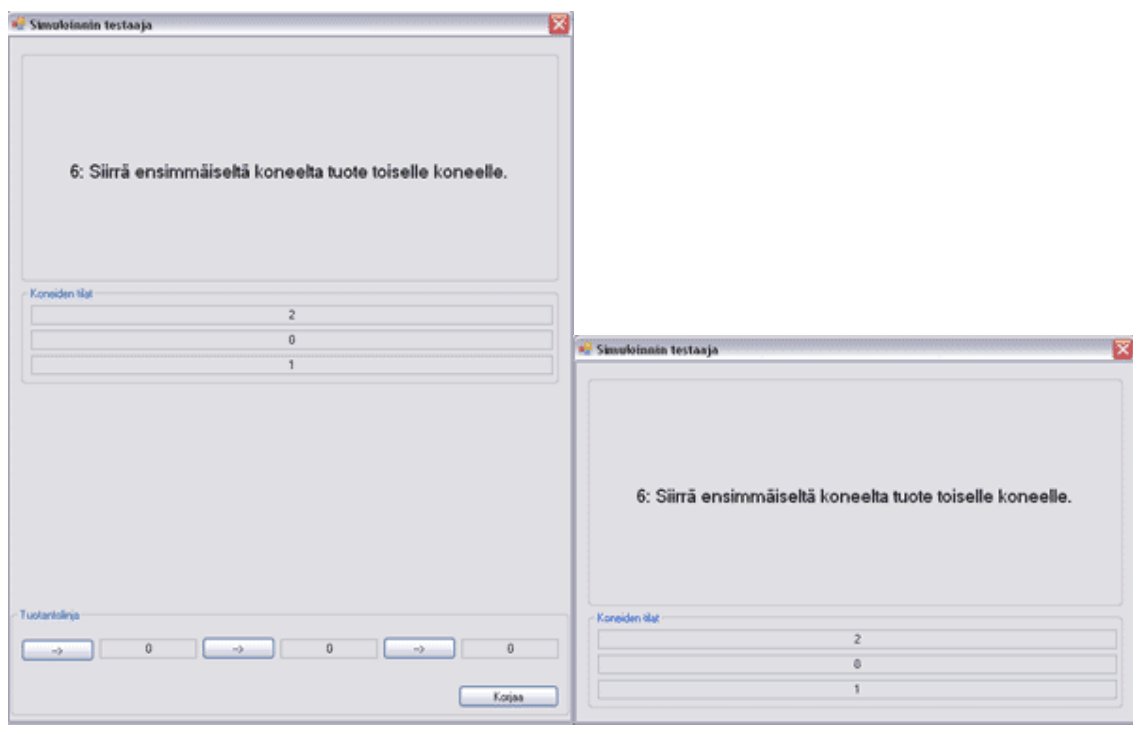


Kuva 38. Koeasetelmat: siirto (vasemmalla), liu'uttaminen (keskellä) ja hiiriohjaus (oikealla).

Toisessa käytössä olevassa testihenkilö sai eteensä prototyypin, joka muodostui viistosti viettävästä kourusta ja siihen kiinnitetyistä koneita vastaavista sulkimista (Kuva 38). Alkutilanteessa kappaleita vastaavat puukuulat olivat kourun yläpäässä. Kahta ensimmäistä sulkijaa vuorotellen hetkellisesti nostamalla testihenkilö saattoi vapauttaa yhden kuulan, jolloin se vieräi ensimmäiselle koneelle. Vastaavasti kourussa oli samanlaiset sulkijamekanismit kappaleiden siirtämiseen toiselle ja kolmannelle koneelle. Johtuen prototyypin yksinkertaisuudesta, tuli sulkijan toteutustavasta lopulta monimutkaisempi kuin alun perin suunnittelin. Tarkoitus oli, että yhden kuulan pystyisi vapauttamaan yhdellä liikkeellä, ilman että sen perässä olevat kuulat pääsisivät sulkijan läpi. Tämä kuitenkin ei onnistunut saatavilla olleista materiaaleista, vaan vapauttaakseen yhden kuulan kerrallaan, täytyi testihenkilöiden tehdä kaksi perättäistä sulkijan nostoliikettä.

Kolmannessa käytössä olevassa testihenkilö ohjasi näytöllä liikkuvaa kursoria normaalin tapaan hiirellä (Kuva 38). Näytön alareunassa oli kolme tuotantolinjan konetta vastaavat kolme rinnakkaista numerokenttää, joissa kussakin näkyi vastaavalla koneella kyseisenä hetkenä olevien kappaleiden määrä (Kuva 39). Lisäksi näytöllä oli kolme nuolipainiketta, joilla koehenkilö pystyi siirtämään kappaleita ensimmäiselle koneelle, ensimmäiseltä koneelta toiselle koneelle ja toiselta koneelta kolmannelle koneelle. Hiirikäyttöliittymä oli toteutettu siten, että napin painallus vähensi yhden edellisestä kentästä ja samalla lisäsi yhden seuraavaan kenttään.

Koko testien ajan testihenkilöiden edessä oli näytöllä nähtävissä suoritettava tehtävä sekä kolme numerokenttää, joihin heidän tuli verrata kulloinkin käyttämänsä testauskäyttöliittymän tilaa (Kuva 39). Hiirikäyttöliittymän tapauksessa näytöllä oli vielä siirtämiseen tarvittavat painikkeet ja käyttöliittymän tilaa ilmaisevat numerokentät.



Kuva 39. Näyttö hiirellä siirrettäessä (vasemmalla) ja käsinkosketeltavia käyttöliittymiä käytettäessä (oikealla).

Tarkkailtavat virheet ilmenivät näytöllä kahdenlaisina eroavaisuuksina. Järjestelmän tilaa esittävissä kentissä saattoi hetkellisesti välkähtää jokin muu arvo, kuin siinä oleva käyttöliittymän tilaa vastaava arvo. Toisaalta kentän arvo saattoi myös virheellisesti kasvaa yhdellä ja pysyä virheellisenä niin pitkän ajan, kunnes käyttäjä asian huomasi. Välähdysten kesto oli yksi sekunti ja ne tapahtuivat satunnaisesti puolesta neljään sekuntia kappaleen liikuttamisen jälkeen. Välähdysten tarkoitus oli mitata, kuinka hyvin testihenkilöt pystyvät tarkkailemaan näytöllä olevia kenttiä samalla kun suorittavat tehtäviään. Pysyvien virheiden tarkoitus taas oli selvittää, kuinka hyvä käsitys testihenkilöillä on eri koneilla olevien kappaleiden määristä. Kussakin sarjassa tapahtui kuusi välähdystä ja kolme pysyvää muutosta.

Testauksen tarkkailijan apuna oli ohjelmisto, joka vastasi tehtävien antamisesta, järjestelmän tilan ylläpidosta sekä eroavaisuuksien tuottamisesta käyttöliittymän ja tarkkailtavien kenttien välille. Mistään käyttöliittymästä ei ollut liityntää näytöllä oleviin numerokenttiin, vaan tilojen yhtenäisyydestä vastasi tarkkailija. Ohjelmistossa oli omat tilat hiirellä ohjaamiselle ja käsinkosketeltavalla käyttöliittymällä ohjaamiselle. Tilat erosivat vain hiiritilassa näkyvien, ohjaamisen mahdollistavien käyttöliittymäkontrollien osalta. Ohjelmistossa voitiin valita yksi kolmesta ennalta määrätystä ohjelmasta kappaleiden siirtojärjestykseksi ja virheiden ilmenemisjärjestykseksi.

Tarkkailija pystyi ohjaamaan testin kulkua näppäimistön avulla. Testissä käytettiin siis luvussa 6 esiteltyä Ozin velho -tekniikkaa. Aina käyttäjän siirtäessä kappaletta

tarkkailija muutti näytöllä esitetyn kappaleiden määrän oikeaksi. Koska testisarjat noudattivat tiettyä ohjelmaa, saattoi tarkkailija muuttaa näytön tilan oikeaksi yhdellä painikkeen painalluksella. Ohjelmisto antoi seuraavan tehtävän heti tarkkailijan painettua painiketta edellisen tehtävän valmistumisen merkiksi.

Tarkkailijan vastuulla oli myös reagoida käyttäjän huomaamiin virheisiin ja palauttaa järjestelmän tila oikeaksi. Pysyvän virheen huomaamisen jälkeen tarkkailija pystyi painiketta painamalla palauttamaan näytöllä olleiden kenttien arvot oikeiksi. Käytössään olleita kahta muuta painiketta tarkkailija käytti merkitsemään oikein havaitut välähdykset, väärin havaitut välähdykset sekä väärin suoritettut siirrot. Käyttäjää seuraamalla tehtyjen havaintojen tueksi tallentui lokitiedostoon testin kokonaiskesto, oikein havaittujen välähdysten määrä, virheellisesti havaittujen välähdysten määrä, havaittujen pysyvien virheiden määrä, pysyvien virheiden havaitsemisajat sekä virheellisten eli ohjeiden vastaisten siirtojen määrä.

### 7.2.3. Testin kulku

Testihenkilöiden tehtävänä oli liikuttaa kappaleet tuotantolinjan läpi ruudulla esitettyjen ohjeiden järjestystä noudattaen ja samalla tarkkailla mahdollisia eroavaisuuksia kulloinkin käytettävän käyttöliittymän ja näytöllä näkyvien numerokenttien arvojen välillä. Tilanne vastasi käsinkosketeltavan käyttöliittymän käyttämistä ympäristön ohjaamiseen kuten SiteView-järjestelmässä [Beckmann and Dey, 2003], mutta tässä tapauksessa näytölle tuotettiin tahallisesti virheellistä palautetta.

Kukin testiin osallistuneista henkilöistä käytti kaikkia kolmea erilaista käyttötapaa. Käyttöjärjestys oli lisäksi tasapainotettu siten, että kutakin käyttöliittymää käytettiin vuorollaan ensimmäisenä, keskimmäisenä ja viimeisenä. Lisäksi käytössä oli kaikki kolme erilaista virheiden ilmenemishjelmaa, jolloin niiden järjestystä tai ajoitusta ei voinut oppia. Vaikka virheiden ilmenemisjärjestykset vaihtuivat eri ohjelmien välillä, oli ne kaikki laadittu hyvin samankaltaisiksi.

Testin aluksi kullekin testihenkilölle ohjeistettiin testin kulku. Hänelle kerrottiin taustatarina tuotantolinjasta ja sen testauksesta sekä kuvattiin kaksi erilaista virhetyyppiä niitä kuitenkaan ennalta näyttämättä. Seuraavaksi testihenkilöille ohjeistettiin eri käyttöliittymätyyppien toiminta ja heidän annettiin vapaasti harjoitella kunkin käyttöliittymän käyttämistä sopivaksi tuntemansa ajan verran.

Kun testihenkilö tunsu osaavansa käyttää kaikkia käyttöliittymiä tarpeeksi hyvin, aloitettiin testi. Aluksi kaikki viisi kappaletta olivat lähtötilanteessa, eli mikään niistä ei sijainnut millään koneista. Testihenkilö seurasi näytölle ilmestyviä ohjeita, jotka olivat muotoa: ”Siirrä ensimmäiseltä koneelta tuote toiselle koneelle”. Virheen havaittuaan testihenkilö ilmaisi asian ilmoittamalla siitä tarkkailijalle. Koko testin ajan tarkkailija piti yllä järjestelmän tilaa sekä kirjasi kerättävät tiedot lokitiedostoon. Lopputilanteessa kaikki viisi kappaletta olivat kolmannella koneella. Testihenkilön suoritettua kaikki

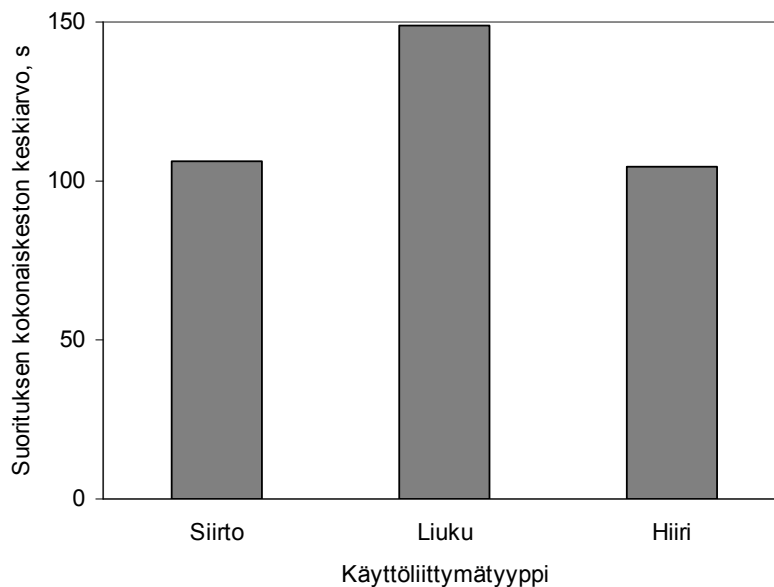
viisitoista tehtävää yhdellä käytettävällä, siirryttiin käyttämään seuraavaa tapaa, kunnes tehtävät oli suoritettu kaikkia kolmea erilaista käyttötapaa käyttäen.

Testin loputtua testihenkilöille suoritettiin lomakkeen (Liite 1) avulla haastattelu ja heiltä kysyttiin mielipiteitä eri käyttöliittymien hyödyllisyydestä. Kokonaisuudessaan yhdeltä henkilöltä kului testin suorittamiseen noin puoli tuntia.

### 7.3. Tulokset

#### 7.3.1. Suorituksen nopeus

Keskimääräinen suorituksen kokonaiskestoaja (Kuva 40) oli suurin liu'uttamisen tapauksessa (149 sekuntia). Siirtämällä ja hiirellä liikuttamalla kokonaiskestoajat olivat keskimäärin miltei samoja (106 sekuntia ja 104 sekuntia).



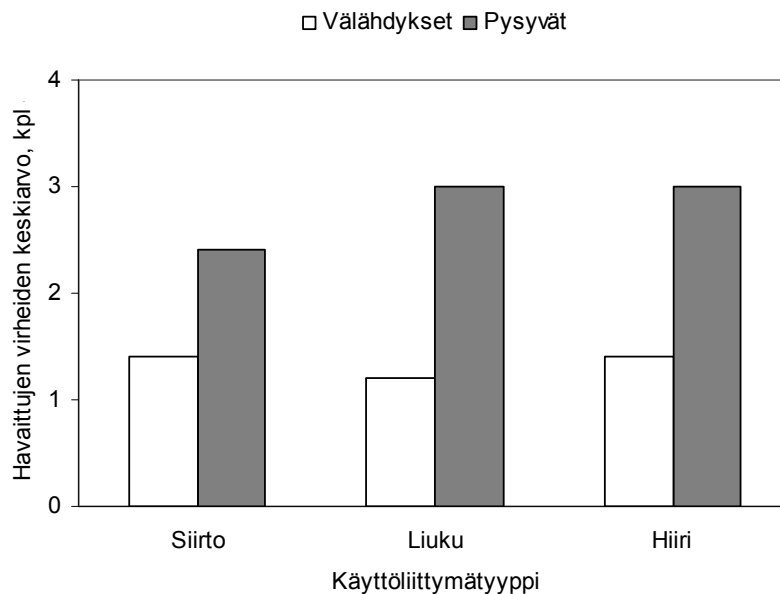
Kuva 40. Suoritusten kokonaiskestoajojen keskiarvot käyttöliittymätyypeittäin.

Testihenkilöiden kommentit eri käyttöliittymätyyppien nopeakäyttöisyydestä vaihtelivat. Yhden mielestä hiirellä käyttämisen haittapuolena oli ”hitaus kohdistamisessa”, kun taas toisen mielestä hiirellä oli ”nopea tehdä” ja kolmas jopa arveli, että ”helposti tulee virheitä, kun pääsee niin nopeasti siirtelemään tuotteita”. Liu'uttamista yksi testihenkilöistä kuvasi sanoilla ”aikaa vievä”.

Kukaan testihenkilöistä ei suorittanut virheellisiä siirtoja, eli kukaan ei liikuttanut kappaletta eri koneelle kuin mitä tehtävän kuvauksessa ohjeistettiin. Kukaan ei myöskään ilmoittanut virheestä sellaisessa tilanteessa, jossa virhettä ei oikeasti tapahtunut. Tästä johtuen suoritusten kestot ja havaituista virheistä kerätyt tiedot ovat vertailukelpoisia.

### 7.3.2. Virheiden havaitseminen

Havaittujen virheiden määrät olivat käsinkosketeltavien käyttöliittymien osalta vastakkaiset kuin mitä olin odottanut (Kuva 41). Siirtämisessä välähdyksiä havaittiin keskimäärin 1,4 kappaletta, joka oli täsmälleen sama luku kuin hiirellä käytettäessä. Yllättäen liu'uttamisessa välähdyksiä havaittiin hieman vähemmän (keskimäärin 1,2 kappaletta), vaikka näytön tarkkailun olisi luullut olevan helpompaa katseen vapautumisen johdosta. Pysyviä virheitä liu'uttamisen aikana havaittiin keskimäärin 3 kappaletta, joka puolestaan oli sama määrä kuin hiirellä käyttämisen aikana. Yllätyksenä tuli kuitenkin se, että siirtämisen aikana havaittujen pysyvien virheiden keskiarvo oli pienempi (2,4 kappaletta), vaikka oletin, että siirtämisestä saatu kinesteettinen palaute helpottaisi kappaleiden määrien muistamista.



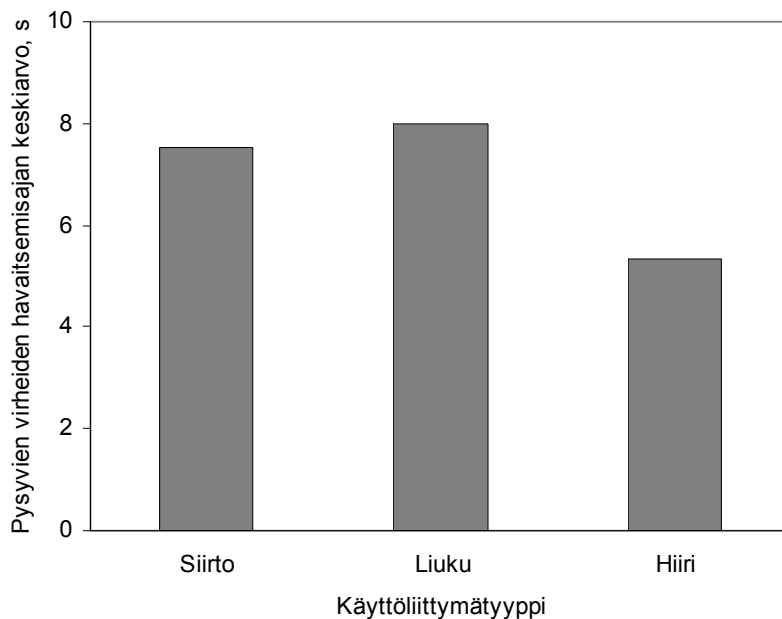
Kuva 41. Havaittujen virheiden lukumäärien keskiarvot käyttöliittymätyypeittäin.

Hiirellä käytettäessä oli testihenkilöiden mielestä ”katse [...] helppoa pitää ruudussa” ja virheet havaittavissa, koska ”vertailu [oli] helppoa” ja ”yhellä katseella pystyi vertailemaan”. Tämä olikin odotettua, sillä kaikki vertailtavat kentät sijaittivat samalla näytöllä, joskin sen eri osissa.

Myös siirtoon liittyvät kommentit olivat odotettuja. Siirrossa oli testihenkilöiden mielestä ”tilanteen hahmottaminen helppoa” ja siinä ”pysyi parhaiten perillä tuotteiden määristä eri koneilla”, mikä tukee oletusta kinesteettisen palautteen vaikutuksesta. Oletusta katseen käytöstä tukee myös se, että ”katse [oli] enemmän nappuloissa kuin ruudussa”, mistä johtuen ”ruudulla tapahtuvien asioiden huomiointi kärsi”. Siirtämisen katsottiin myös ”vaati[van] oppimista” ja vertailu oli sitä käyttäen ”vaikeampaa kuin hiirellä”.

Liu'uttamisesta oltiin yllättäen sitä mieltä, että ”katse [oli] siirtovälineessä, samoin [kuin] huomio”. Liu'uttamista kuvailtiin samaan aikaan sekä sanoilla ”hankalaa” ja ”monimutkaista” että ”havainnollista”. Liu'uttamisen hitautta eräs testihenkilö kommentoi sanomalla, että ”oli tarpeeksi aikaa seurata virheitä”. Yleisesti ottaen käsinkosketeltavien käyttöliittymien ”konkreettisuutta” pidettiin hyvänä.

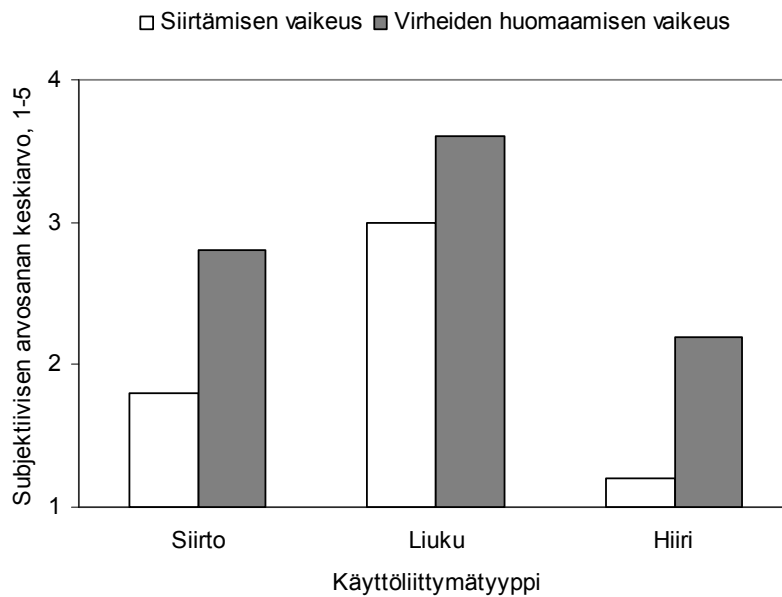
Myös pysyvien virheiden havaitsemisajat mitattiin (Kuva 42). Nopeimmin pysyvät virheet havaittiin hiirellä käytettäessä, jolloin keskiarvo havaitsemisajalle oli 5,4 sekuntia. Siirtäessä virheet havaittiin hieman nopeammin kuin liu'uttamisessa, keskimääräisten havaitsemisaikojen ollessa 7,5 ja 8,0 sekuntia.



Kuva 42. Pysyvien virheiden havaitsemisaikojen keskiarvot käyttöliittymätyypeittäin.

### 7.3.3. Subjektiiiset arvosanat

Testauksen suorittamisen jälkeen testihenkilöiltä kysyttiin haastattelulomakkeen (Liite 1) avulla subjektiivisia arvosanoja eri käyttöliittymätyyppien käytöstä. Jokainen käyttötapa sai kaksi arvosanaa (siirtämisen vaikeus ja virheiden huomaamisen vaikeus), jotka mitattiin asteikolla 1-5 (1=helppoa, 5=vaikeaa) (Kuva 43). Parhaiten subjektiivisissa arvosanoissa menestyi hiiri, jota käytettäessä sekä siirtämistä (arvosana 1,2) että virheiden havaitsemista (arvosana 2,2) pidettiin helpoimpina. Seuraavaksi parhaiten menestyi siirtäminen arvosanoilla 1,8 ja 2,8. Vaikeimmaksi kummankin ominaisuuden suhteen arvioitiin liu'uttaminen, joka sai arvosanoikseen siirtämisestä 3,0 ja virheiden huomaamisesta 3,6.



Kuva 43. Subjektiiivisten arvosanojen keskiarvot käyttöliittymätyypeittäin (1=helppoa, 5=vaikeaa).

#### 7.4. Pohdintaa

Suuria eroja vertailtujen käyttöliittymätyyppien välillä ei löytynyt, mutta hiirikäyttöisenä järjestelmää pidettiin helpoimpana käyttää ja sen avulla virheet myös havaittiin nopeimmin. Syyksi hiirikäyttöisyyden suosiolle ja helppokäyttöisyydelle epäilen sitä, että testihenkilöt olivat tottuneet käyttämään hiirikäyttöisiä järjestelmiä ja näin ollen oppivat sen helposti, jolloin käsinkosketeltavien käyttöliittymien intuitiivisuus ei päässyt esiin. Toisaalta hiiren tapauksessa kummatkin tarkkailtavat asiat olivat samalla näytöllä ja numeromuotoisia kenttiä pystyi vertailemaan suoraan toisiinsa koneilla olevien kappaleiden määriä syvällisemmin sisäistämättä. Hiirikäyttöliittymässä olisi toki voinut siirtää tarkkailtavat kentät toiselle näytölle, jolloin tilanne olisi vastannut tarkemmin tilannetta käsinkosketeltavissa käyttöliittymissä. Halusin kuitenkin pitää yhden käyttötavoista mahdollisimman lähellä parasta tämän hetkistä vaihtoehtoa testausjärjestelyille, sillä mahdollistaahan graafinen käyttöliittymä testausikkunan ja vertailuikkunan järjestämisen rinnakkain samalle näytölle.

Yllättävänä havaintona testissä selvisi, että liu'uttaminen ei mahdollistanutkaan näytön kiinteää seuraamista suorituksen aikana. Syyksi tähän epäilen prototyypin lukitusmekanismia, josta tuli lopulta hankalampi käyttää kuin alun perin oli tarkoitus. Tähän viittaavat myös liu'uttamisen kokonaissuoritusajan pituus sekä käyttäjien kommentit käyttötavan hitaudesta ja monimutkaisuudesta. Eniten tästä ongelmasta paljasti kuitenkin se testejä tarkkaillessani tekemä havainto, että testihenkilöt joutuivat

kiinnittämään liiaksi huomioita lukitusmekanismien käsittelyyn, jolloin näytöllä tapahtuvien muutosten seuraaminen vaikeutui. Mekaanisen käsittelyn ongelmista huolimatta säilyi liu'uttamista käytettäessä järjestelmän tila kuitenkin hyvin testihenkilöiden tietoisuudessa, sillä pysyviä virheitä havaittiin hyvin.

Toinen yllättävä testissä selvinnyt seikka oli se, että käsin siirtäessään testihenkilöt havaitsivat vähemmän pysyviä virheitä kuin muita käyttöliittymätyyppejä käyttäessään. Vaikka testihenkilöiden kommentit tukivatkin oletusta kinesteettisen palautteen helpottavasta vaikutuksesta koneilla olevien määrien muistamiseen, eivät mitatut tulokset tukeneet tätä hypoteesia. Havaintoa on kuitenkin vaikea selittää, sillä testihenkilöt huomasivat pysyviä muutoksia paremmin myös liu'uttamista käyttäessään, vaikka sen mekaaninen toteutus olikin puutteellinen, ja näin ollen testihenkilöiden huomio kiinnittyi lukumäärien tarkkailun sijasta kappaleiden liikuttamiseen. Vaikka siirtämisessä ei ollut tätä ongelmaa ja testihenkilöt pitivät siirtämistä ja tuotteiden määrien tarkkailua helppona, ei jostain syystä pysyviä virheitä huomattu yhtä hyvin kuin muissa vaihtoehdoissa.

Testin tulokset erosivat jonkin verran aikaisemmista tutkimustuloksista käsinkosketeltavien käyttöliittymien hyödyllisyyden mittaamiseksi. Vaikka käytetyt käsinkosketeltavat kappaleet olivatkin arkisia, eivät käsinkosketeltavat käyttöliittymät osoittautuneet tässä testissä yhtä hyödyllisiksi kuin Huangin [2004] suorittamassa testauksessa. Myöskään kinesteettisestä palautteesta saatu hyöty ei osoittautunut yhtä vahvaksi kuin Pattenin ja Ishiin [2000] testissä.

Suurimmat puutteet suoritettussa testauksessa liittyivät käsinkosketeltavien käyttöliittymien monimutkaisimpien ominaisuuksien puuttumiseen testiasetelmasta. Testihenkilöt toimivat yksinään ja tehtävien suorittaminen ei vaatinut ryhmätyötä. Tästä syystä ei tarvittu ulkoisia apuvälineitä kommunikoinnin tehostamiseen, johon tarkoitukseen käsinkosketeltavat käyttöliittymät on havaittu sopiviksi [Maher and Kim, 2005]. Tehtävien suorittaminen ei myöskään vaatinut monimutkaisia liikkeitä, tarkkaa koordinaatiota tai kahden käden käyttöä, jotka kaikki erottavat käsinkosketeltavat käyttöliittymät edukseen muista käyttöliittymätyypeistä. Esimerkiksi Waldner ja muut [2006] havaitsivat käsinkosketeltavien käyttöliittymien edut juuri luonnollisen manipuloinnin hyödyntämisessä ja Fitzmaurice ja Buxton [1997] tilallisen limittyneisyyden toteutuessa.

Se, että kappaleita kuvastaneet aineelliset objektit eivät muistuttaneet ulkonäöllisesti reaali maailman vastineitaan (kuten Fitzmauricen ja Buxtonin [1997] testissä), oli harkittu seikka, sillä tällöin samoja kappaleita voidaan käyttää esittämään erilaisia tuotteita. Jos koneilla olevat kappaleiden määrät olisivat kuitenkin olleet nähtävissä suoraan käsinkosketeltavista käyttöliittymistä ilman tarvetta erilliselle näytölle, olisivat käsinkosketeltavien käyttöliittymien edut tulleet paremmin hyödynnetyiksi. Tällöin käsinkosketeltaville käyttöliittymille tyypillinen syötteiden ja



tulostusten yhdyntyminen olisi saattanut parantaa tuloksia käsinkosketeltavien käyttöliittymien osalta, kuten kävi Jacobin ja muiden [2002] ja Fjeldin ja muiden [2002] testeissä.

Suoritetun testauksen perusteella näyttäisi siltä, että vaikka käsinkosketeltavat käyttöliittymät osoittautuivat vertailukelpoisiksi hiirikäyttöisen käyttöliittymän kanssa, vaaditaan tuotantolinjan testaustavalta tiettyjä ominaisuuksia, jotta käsinkosketeltava käyttöliittymä erottautuisi edukseen. Jotta näin kävisi, tulisi testauksen sisältää esimerkiksi monipuolisia ja vaativia liikkeitä tai ryhmätyötä. Pienen parannuksen nyt testattujen käyttöliittymien toimivuuteen saisi luultavasti lisäämällä siihen ilmaisimet koneilla olevien kappaleiden määrille.

Jatkossa tulisi tutkia, kuinka käsinkosketeltavat käyttöliittymät toimivat tilanteessa, jossa testaaja itse päättää testauksen järjestyksestä eli testitapauksista. Jos testihenkilö olisi tässä tutkimuksessa saanut itse valita, mihin kappaleen kulloinkin siirtää – eli minkälaisia testitapauksia suorittaa – olisivat tulokset saattaneet olla toisenlaisia. Myös monipuolisten liikkeiden ja ryhmätyön todelliset vaikutukset järjestelmän tehokkuuteen tulisi selvittää ennen käsinkosketeltavan testauskäyttöliittymän hyödyllisyyden lopullista arviointia. Parhaiten se onnistuisi suunnittelun kautta tapahtuvaa tutkimusta [van den Hoven *et al.*, 2007] noudattaen eli viemällä jatkokehitetty prototyyppi oikeaan käyttöympäristöönsä ja jatkamalla iteratiivista kehittämistä.

## 8. Yhteenveto

Tässä tutkimuksessa tutkin, kuinka käsinkosketeltava käyttöliittymä voisi helpottaa tuotantolinjan ohjausjärjestelmän testausta. Tutkimuksessa selvisi, että käsinkosketeltavan käyttöliittymän kehitysvaiheessa tehdyillä suunnitteluratkaisuilla on suuri vaikutus käyttöliittymän lopulliseen hyödyllisyyteen. Valinnanmahdollisuuksia on suunnitteluratkaisuissa niin paljon, että pelkästään kokeilemalla on onnistuneeseen lopputulokseen päätyminen epätodennäköistä. Kuten tätäkin tutkimusta innoittaneista sovelluksista huomaa, löytyy olemassa olevista sovelluksista piirteitä, joita on mahdollista hyödyntää uusilla sovellusalueilla. Aikaisemmin toteutettuja käsinkosketeltavia käyttöliittymiä kannattaakin tutkia ennen oman sovelluksen toteuttamista, sillä niiden kehitys on yleensä tapahtunut iteratiivisen testauksen tuloksena.

Teknologian taso asettaa rajoituksia käsinkosketeltavien käyttöliittymien kehitykselle. Tulevaisuudessa tullaan näkemään uudenlaisia sovelluksia, kun teknologian kehittyessä on mahdollista toteuttaa yhä pienemmistä komponenteista koostuvia syöte- ja tulostuslaitteita. Itse uskon, että erityisesti aktiivisen haptisen palautteen tuottaminen tulee lisääntymään. Uskon myös, että tulevaisuudessa tullaan kehittämään yhä hajautetumpia arkkitehtuureja sekä abstraktimpia ja yleiskäyttöisempiä sovelluksia. Käsinkosketeltavien käyttöliittymien kehittämistä helpottavien ohjelmistokehysten myötä nopeutuu sovellusten kehittäminen entisestään. Kuten tutkimuksen yhteydessä suoritettua testauksessa havaitsin, pelkkä ohjelmistojen kehittäminen ei kuitenkaan riitä, vaan myös testauksessa käytettävien fyysisten prototyyppien tulisi olla mahdollisimman toimivia. Tämä on erityisen tärkeää juuri käsinkosketeltavissa käyttöliittymissä, joissa mekaanisen toimivuuden rajoitteet voivat vaikuttaa testaustuloksiin.

Suoritettu tutkimus osoitti myös sen, että käsinkosketeltavien käyttöliittymien hyödyt eivät tule tehokkaasti esiin, mikäli käsinkosketeltavuuden mahdollistamia monipuolisia ominaisuuksia ei ole hyödynnetty tehokkaasti. Vielä ei ole kuitenkaan täysin selvää, mitkä kaikki ominaisuudet vaikuttavat käsinkosketeltavien käyttöliittymien tehokkuuteen. Näitä ominaisuuksia tulisi yrittää selvittää pitkäkestoisten, oikeassa käyttöympäristössä tehtyjen tarkkailututkimusten avulla. Kuten tässäkin tutkimuksessa suoritettu testaus, ovat useimmat käsinkosketeltavien käyttöliittymien tutkimukset suoritettu kontrolloiduissa olosuhteissa. Erilaisia menetelmiä käyttämällä kerätty tieto olisi kuitenkin luotettavampaa ja oikeassa ympäristössä tapahtuvat tutkimukset saattaisivat paljastaa käsinkosketeltavista käyttöliittymistä uusia piirteitä.

## Viiteluettelo

- [Andersson et al., 2002] Gerd Andersson, Kristina Höök, Dário Mourão, Ana Paiva and Marco Costa, Using a Wizard of Oz study to inform the design of SenToy. In: *Proc. of Conference on Designing Interactive Systems (2002)*, 349-355.
- [Avrahami and Hudson, 2002] Daniel Avrahami and Scott E. Hudson, Forming interactivity: a tool for rapid prototyping of physical interactive products. In: *Proc. of Conference on Designing Interactive Systems (2006)*, 141-146.
- [Ballagas et al., 2003] Rafael Ballagas, Meredith Ringel, Maureen Stone, and Jan Borchers, iStuff: a physical user interface toolkit for ubiquitous computing environments. In: *Proc. of Conference on Human Factors in Computing Systems (2003)*, 537-544.
- [Ballagas et al., 2004] Rafael Ballagas, Andy Szybalski, and Armando Fox, Patch panel: enabling control-flow interoperability in ubicomp environments. In: *Proc. of Conference on Pervasive Computing and Communications (2004)*, 241-252.
- [Beckmann and Dey, 2003] Chris Beckmann and Anind Dey, SiteView: tangibly programming active environments with predictive visualization. *Intel Research Tech Report IRB-TR-03-019 (2003)*.
- [Bellotti et al., 2002] Victoria Bellotti, Maribeth Back, W. Keith Edwards, Rebecca E. Grinter, Austin Henderson and Cristina Lopes, Making sense of sensing systems: five questions for designers and researchers. In: *Proc. of Conference on Human Factors in Computing Systems (2002)*, 415-422.
- [Benford et al., 2005] Steve Benford, Holger Schnadelbach, Boriana Koleva, Rob Anastasi, Chris Greenhalgh, Tom Rodden, Jonathan Green, Ahmed Ghali, Tony Pridmore, Bill Gaver, Andy Boucher, Brendan Walker, Sarah Pennington, Albrecht Schmidt, Hans Gellersen and Anthony Steed, Expected, sensed, and desired: a framework for designing sensing-based interaction. *Transactions on Computer-Human Interaction* **12**, 1 (2005), 3-30.
- [Betts et al., 1987] Bill Betts, David Burlingame, Gerhard Fischer, Jim Foley, Mark Green, David Kasik, Stephen T. Kerr, Dan Olsen, and James Thomas, Goals and objectives for user interface software. *Computer Graphics* **21**, 2 (1987), 73-78.
- [Billinghurst et al., 2002] Mark Billinghurst, Hirokazu Kato, Kiyoshi Kiyokawa, Daniel Belcher and Ivan Poupyrev, Experiments with face-to-face collaborative AR interfaces. *Virtual Reality* **6**, 3 (2002), 107-121.
- [Blackwell, 2003] Alan F. Blackwell, Cognitive dimensions of tangible programming techniques. In: *Proc. of First Joint Conference of EASE and PPIG (2003)*, 391-405.

- [Blackwell et al., 2005] Alan F. Blackwell, Darren Edge, Lorisa Dubuc, Jennifer A. Rode, Mark Stringer and Eleanor Toye, Using Solid Diagrams for Tangible Interface Prototyping. *Pervasive Computing* 4, 4 (2005), 74-77.
- [Boess et al., 2007] Stella Boess, Daniel Saakes and Caroline Hummels, When is role playing really experiential? case studies. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 279-282.
- [Brereton and McGarry, 2000] Margot Brereton and Ben McGarry, An observational study of how objects support engineering design thinking and communication: implications for the design of tangible media. In: *Proc. of Conference on Human Factors in Computing Systems* (2000), 217-224.
- [Buschmann et al., 1996] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad and Michael Stal, *Pattern-Oriented Software Architecture: A System of Patterns*. Wiley, 1996.
- [Buur et al., 2004] Jacob Buur, Mads Vedel Jensen and Tom Djajadiningrat, Hands-Only Scenarios and Video Action Walls—novel methods for tangible user interaction design. In: *Proc. of Designing Interactive Systems* (2004), 185-192.
- [Buxton, 1990] William A. S. Buxton, A three-state model of graphical input. In: *Proc. of IFIP INTERACT'90: Human-Computer Interaction* (1990), 449-456.
- [Buxton and Myers, 1986] William Buxton and Brad A. Myers, A study in two-handed input. In: *Proc. of Conference on Human Factors in Computing Systems* (1986), 321-326.
- [Calvillo-Gómez et al., 2003] Eduardo H. Calvillo-Gómez, Nancy Leland, Orit Shaer and Robert J. K. Jacob, The TAC paradigm: unified conceptual framework to represent tangible user interfaces. In: *Proc. of CLIHC 2003, Latin American Conference on Human-Computer Interaction* (2003), 9-15.
- [Carvey et al., 2006] Andrew Carvey, Jim Gouldstone, Pallavi Vedurumudi, Adam Whiton and Hiroshi Ishii, Rubber shark as user interface. In: *Proc. of Conference on Human Factors in Computing Systems* (2006), 634-639.
- [Chang et al., 2007] Angela Chang, James Gouldstone, Jamie Zigelbaum and Hiroshi Ishii, Simplicity in interaction design. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 135-138.
- [Decker et al., 2005] Christian Decker, Albert Krohn, Michael Beigl and Tobias Zimmer, The particle computer system. In: *Proc. of Conference on Information Processing in Sensor Networks* (2005).
- [Djajadiningrat et al., 2002] Tom Djajadiningrat, Kees Overbeeke and Stephan Wensveen, But how, Donald, tell us how?: on the creation of meaning in interaction design through feedforward and inherent feedback. In: *Proc. of Conference on Designing Interactive Systems* (2002), 285-291.

- [Djajadiningrat et al., 2004] Tom Djajadiningrat, Stephan Wensveen, Joep W. Frens, Kees C. J. Overbeeke, Tangible products: redressing the balance between appearance and action. *Personal and Ubiquitous Computing* **8**, 5 (2004), 294-309.
- [Dourish, 2001] Paul Dourish, *Where the action is: the foundations of embodied interaction*. MIT Press, 2001.
- [Edge and Blackwell, 2006] Darren Edge and Alan Blackwell, Correlates of the cognitive dimensions for tangible user interface. *Journal of Visual Languages and Computing* **17**, 4 (2006), 366-394.
- [Fishkin, 2004] Kenneth P. Fishkin, A taxonomy for and analysis of tangible interfaces. *Personal and Ubiquitous Computing* **8**, 5 (2004), 347-358.
- [Fitzmaurice et al., 1995] George W. Fitzmaurice, Hiroshi Ishii and William Buxton, Bricks: laying the foundations for graspable user interfaces. In: *Proc. of Conference on Human Factors in Computing Systems* (1995), 442-449.
- [Fitzmaurice and Buxton, 1997] George W. Fitzmaurice and William Buxton, An empirical evaluation of graspable user interfaces: towards specialized, space-multiplexed input. In: *Proc. of Conference on Human Factors in Computing Systems* (1997), 43-50.
- [Fjeld et al., 1998] Morten Fjeld, Martin Bichsel and Matthias Rauterberg, BUILD-IT: an intuitive design tool based on direct object manipulation. In: I. Wachsmut & M. Frölich (eds.), *Lecture Notes in Computer Science* **1371**. Springer, 1998, 297-308.
- [Fjeld et al., 2002] Morten Fjeld, Guttormsen Schär, Domenico Signorello and Helmut Krueger, Alternative tools for tangible interaction: a usability evaluation. In: *Proc. of Symposium on Mixed and Augmented Reality* (2002), 157-318.
- [Gamma et al., 1995] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object Oriented Software*. Addison-Wesley, 1995.
- [Gellersen, 2005] Hans Gellersen, Smart-Its: computers for artifacts in the physical world. *Communications of the ACM* **48**, 3 (2005), 66.
- [Gellersen et al., 2004] Hans Gellersen, Gerd Kortuem, Albrecht Schmidt and Michael Beigl, Physical prototyping with Smart-Its. *Pervasive Computing* **3**, 3 (2004), 74-82.
- [Greenberg and Fitchett, 2001] Saul Greenberg and Chester Fitchett, Phidgets: easy development of physical interfaces through physical widgets. In: *Proc. of Symposium on User Interface Software and Technology* (2001), 209-218.
- [Hartmann, 2006] Björn Hartmann, Scott R. Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher and Jennifer Gee, Reflective physical prototyping through integrated design, test, and analysis. In: *Proc. of Symposium on User Interface Software and Technology* (2006), 299-308.

- [Holmquist, 2006] Lars Erik Holmquist, Sketching in hardware. *Interactions* **13**, 1 (2006), 47-60.
- [Holmquist et al., 1999] Lars Erik Holmquist, Johan Redström and Peter Ljungstrand, Token-based access to digital information. In: *Proc. of Handheld and Ubiquitous Computing: First International Symposium* (1999), 234-245.
- [Hornecker, 2005] Hornecker, Eva, A design theme for tangible interaction: embodied facilitation. In: *Proc. of Conference on Computer-Supported Cooperative Work* (2005), 23-43.
- [Hornecker and Buur, 2006] Eva Hornecker and Jacob Buur, Getting a grip on tangible interaction: a framework on physical space and social interaction. In: *Proc. of Conference on Human Factors in Computing Systems* (2006), 437-446.
- [Hornecker and Psik, 2005] Eva Hornecker and Thomas Psik, Using ARToolKit markers to build tangible prototypes and simulate other technologies. In: *Proc. IFIP INTERACT'05: Human-Computer Interaction* (2005), 30-42.
- [van den Hoven and Eggen, 2004] Elise van den Hoven and Berry Eggen, Tangible computing in everyday life: extending current frameworks for tangible user interfaces with personal objects. In: *Proc. of Ambient Intelligence: Second European Symposium* (2004) 230-242.
- [van den Hoven et al., 2007] Elise van den Hoven, Joep Frens, Dima Aliakseyeu, Jean-Bernard Martens, Kees Overbeeke and Peter Peters, Design research & tangible interaction. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 109-115.
- [Huang, 2004] Chen-Je Huang, Not just intuitive: examining the basic manipulation of tangible user interfaces. In: *Proc. of Conference on Human Factors in Computing Systems* (2004), 1387-1390.
- [Hudson and Mankoff, 2006] Scott E. Hudson and Jennifer Mankoff, Rapid construction of functioning physical interfaces from cardboard, thumbtacks, tin foil and masking tape. In: *Proc. of Symposium on User Interface Software and Technology* (2006), 289-298.
- [Hurtienne and Israel, 2007] Jörn Hurtienne and Johann Habakuk Israel, Image schemas and their metaphorical extensions: intuitive patterns for tangible interaction. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 127-134.
- [Ishii and Ullmer, 1997] Hiroshi Ishii and Brygg Ullmer, Tangible Bits: towards seamless interfaces between people, bits and atoms. In: *Proc. of Conference on Human Factors in Computing Systems* (1997), 234-241.
- [Jacob et al., 1999] Robert J. K. Jacob, Leonidas Deligiannidis and Stephen Morrison, A software model and specification language for non-WIMP user interfaces. *Transactions on Computer-Human Interaction* **6**, 1 (1999), 1-46.

- [Jacob et al., 2002] Robert J. K. Jacob, Hiroshi Ishii, Gian Pangaro and James Patten, A tangible interface for organizing information using a grid. In: *Proc. of Conference on Human Factors in Computing Systems* (2002), 339-346.
- [Jensen et al., 2005] Mads Vedel Jensen, Jacob Buur and Tom Djajadiningrat, Designing the user actions in tangible interaction. In: *Proc. of Conference on Critical Computing* (2005), 9-18.
- [Johanson and Fox, 2002] Brad Johanson and Armando Fox, The event heap: a coordination infrastructure for interactive workspaces. In: *Proc. of 4th IEEE Workshop on Mobile Computing Systems and Applications* (2002), 83-93.
- [Järvinen ja Järvinen, 2004] Pertti Järvinen ja Annikki Järvinen, *Tutkimustyön metodeista*. Tampereen yliopistopaino, Tampere, 2004.
- [Kaltenbrunner et al., 2004] Martin Kaltenbrunner, Sile O'Modhrain and Enrico Costanza, Object design considerations for tangible musical interfaces. In: *Proc. of COST287-ConGAS Symposium on Gesture Interfaces for Multimedia Systems* (2004).
- [Kimura et al., 2006] Hiroaki Kimura, Eiji Tokunaga, Yohei Okuda and Tatsuo Nakajima, CookieFlavors: easy building blocks for wireless tangible input. In: *Proc. of Conference on Human Factors in Computing Systems* (2006), 965-970.
- [Kirsh and Maglio, 1994] David Kirsh and Paul P. Maglio, On distinguishing epistemic from pragmatic action. *Cognitive Science* **18**, 4 (1994), 513-549.
- [Kirsh, 1995] David Kirsh, Complementary strategies: Why we use our hands when we think. In: *Proc. of Conference of the Cognitive Science Society* (1995), 212-217.
- [Klemmer, 2003] Scott Klemmer, Papier-Mâché: Toolkit support for tangible interaction. In: *Proc. of ACM Symposium on User Interface Software and Technology: Doctoral Consortium* (2003).
- [Klemmer et al., 2004] Scott R. Klemmer, Jack Li, James Lin, and James A. Landay, Papier-Mâché: toolkit support for tangible input. In: *Proc. of Conference on Human Factors in Computing Systems* (2004), 399-406.
- [Klemmer et al., 2006] Scott R. Klemmer, Björn Hartmann and Leila Takayama, How bodies matter: five themes for interaction design. In: *Proc. of Conference on Designing Interactive Systems* (2006), 140-149.
- [Koleva et al., 2003] Boriana Koleva, Steve Benford, Kher Hui Ng and Tom Rodden, A framework for tangible user interfaces. In: *Proc. of Mobile HCI Conference* (2003), 46-50.
- [Kranz and Schmidt, 2005] Matthias Kranz and Albrecht Schmidt, Prototyping smart objects for ubiquitous computing. In: *Proc. of International Workshop on Smart Object Systems in Conjunction with the Seventh International Conference on Ubiquitous Computing* (2005).

- [Lee et al., 2004] Johnny C. Lee, Daniel Avrahami, Scott E. Hudson, Jodi Forlizzi, Paul H. Dietz, and Darren Leigh, The Calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In: *Proc. Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (2004), 167-175.
- [MacLean et al., 2000] Karon E. MacLean, Scott S. Snibbe and Golan Levin, Tagged handles: merging discrete and continuous manual control. In: *Proc. of Conference Human Factors in Computing Systems* (2000), 225-232.
- [Maher and Kim, 2005] Mary Lou Maher and Mi Jeong Kim, Do tangible user interfaces impact spatial cognition in collaborative design? In: *Proc. of Cooperative Design, Visualization, and Engineering* (2005), 30-41.
- [Marquardt and Greenberg, 2007] Nicolai Marquardt and Saul Greenberg, Distributed physical interfaces with shared phidgets. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 13-20.
- [Marshall, 2007] Paul Marshall, Do tangible interfaces enhance learning? In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 163-170.
- [Mazé and Bueno, 2002] Ramia Mazé and Monica Bueno, Mixers: a participatory approach to design prototyping. In: *Proc. of Conference on Designing Interactive Systems* (2002), 341-344.
- [Mazzone et al., 2004] Andrea Mazzone, Christian P. Spagno and Andreas M. Kunz, The HoverMesh: a deformable structure based on vacuum cells. In: *Proc. of International Conference on Advances in Computer Entertainment Technology* (2004), 187-193.
- [Merrill et al., 2007] David Merrill, Jeevan Kalanithi and Pattie Maes, Siftables: towards sensor network user interfaces. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 75-78.
- [MIAMI, 1995] Lambert Schomaker, Jan Nijtmans, Antonio Camurri, Fabio Lavagetto, Piero Morasso, Christian Benoît, Thierry Guiard-Marigny, Bertrand Le Goff, Jordi Robert-Ribes, Ali Adjoudani, Irek Defée, Stefan Münch, Klaus Hartung, Jens Blauert, A Taxonomy of Multimodal Interaction in the Human Information Processing System. *A report of the Esprit Basic Research Action 8579 MIAMI*, 1995.
- [Michelitsch et al., 2004] Georg Michelitsch, Jason Williams, Martin Osen, Beatriz Jimenez, Stefan Rapp, Haptic chameleon: a new concept of shape-changing user interface controls with force feedback. In: *Proc. of Conference on Human Factors in Computing Systems* (2004), 1305 -1308.
- [Ng et al., 2005] Kher Hui Ng, Steve Benford and Boriana Koleva, PINS push in and POUTS pop out: creating a tangible pin-board that ejects physical documents. In: *Proc. of Conference on Human Factors in Computing Systems* (2005), 1981-1984.



- [Norman, 1990] Donald A. Norman, *The design of everyday things*. Doubleday, 1990.
- [Norman, 1999] Donald A. Norman, Affordance, conventions, and design. *Interactions* **6**, 3 (1999), 38-43.
- [Oppl et al., 2006] Stefan Oppl, Christian Stary and Andreas Auinger, Towards Tangible Work Modeling. In: *Proc. of Mensch Computer* (2006).
- [Oppl, 2006] Stefan Oppl, Towards Intuitive Work Modeling with a Tangible Collaboration Interface Approach. In: *Proc. of International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises* (2006).
- [Pangaro et al., 2002] Gian Pangaro, Dan Maynes-Aminzade and Hiroshi Ishii. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In: *Proc. of Symposium on User Interface Software and Technology* (2002), 181-190.
- [Patten and Ishii, 2000] James Patten and Hiroshi Ishii, A comparison of spatial organization strategies in graphical and tangible user interfaces. In: *Proc. of Conference on Designing Augmented Reality Environments* (2000), 41-50.
- [Poupyrev et al., 2007] Ivan Poupyrev, Tatsushi Nashida and Makoto Okabe, Actuation and tangible user interfaces: the Vaucanson duck, robots, and shape displays. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 205-212.
- [Poynor, 1995] Rick Poynor, The Hand That Rocks the Cradle. *The International Design Magazine* (May/June 2005), 60-65.
- [Pangaro et al., 2002] Gian Pangaro, Dan Maynes-Aminzade and Hiroshi Ishii, The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In: *Proc. of ACM Symposium on User Interface Software and Technology* (2002), 181-190.
- [Sandor and Klinker, 2005] Christian Sandor and Gudrun Klinker, A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality. *Personal and Ubiquitous Computing* **9**, 3 (2005), 169-185.
- [Schäfer et al., 1997] Kai Schäfer, Volker Brauer and Willi Bruns, A New Approach to human-computer interaction—synchronous modelling in real and virtual spaces. In: *Proc. of Conference on Designing Interactive Systems* (1997), 335-344.
- [Shaer, 2005] Orit Shaer, A framework for building reality-based interfaces for wireless-grid applications. In: *Proc. of Conference on Human Factors in Computing Systems, Extended Abstracts* (2005), 1128-1129.
- [Shaer et al., 2004] Orit Shaer, Nancy Leland, Eduardo H. Calvillo-Gamez and Robert J. K. Jacob, The TAC paradigm: specifying tangible user interfaces. *Personal and Ubiquitous Computing* **8**, 5 (2004), 359-369.
- [Sharlin et al., 2002] Ehud Sharlin, Yuichi Itoh, Benjamin Watson, Yoshifumi Kitamura, Steve Sutphen and Lili Liu, Cognitive cubes: a tangible user interface

- for cognitive assessment. In: *Proc. of Conference on Human Factors in Computing Systems* (2002), 347-354.
- [Sharlin et al., 2004] Ehud Sharlin, Benjamin Watson, Yoshifumi Kitamura, Fumio Kishino and Yuichi Itoh. On tangible user interfaces, humans and spatiality. *Personal and Ubiquitous Computing* **8**, 5 (2004), 338-346.
- [Sharlin et al., 2005] Ehud Sharlin, Yuichi Itoh, Benjamin Watson, Yoshifumi Kitamura, Steve Sutphen, Lili Liu and Fumio Kishino, Spatial tangible user interfaces for cognitive assessment and training. In: Auke Jan Ijspeert, Masayuki Murata and Naoki Wakamiya (eds.), *Lecture Notes in Computer Science* **3141**. Springer, 2005, 137-152.
- [Shneiderman, 1983] Ben Shneiderman, Direct manipulation: a step beyond programming languages. *IEEE Computer* **16**, 8 (1983), 57-69.
- [Shneiderman, 1997] Ben Shneiderman, Direct manipulation for comprehensible, predictable and controllable user interfaces. In: *Proc. of International Conference on Intelligent User Interfaces* (1997), 33-39.
- [Spiessl et al., 2007] Wolfgang Spiessl, Nicolas Villar, Hans Gellersen and Albrecht Schmidt, VoodooFlash: authoring across physical and digital form. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 97-100.
- [Stringer et al., 2005] Mark Stringer, Jennifer A. Rode, Eleanor F. Toye, Alan F. Blackwell and Amanda R. Simpson, The Webkit tangible user interface: a case study of iterative prototyping. *IEEE Pervasive Computing* **4**, 4 (2005), 35-41.
- [Svanaes and Verplank, 2000] Dag Svanaes and William Verplank, In search of metaphors for tangible user interfaces. In: *Proc. of Conference on Designing Augmented Reality Environments* (2000), 121-129.
- [Ullmer, 1997] Brygg Ullmer, 3wish: distributed [incr Tcl] extensions for physical-world interfaces. In: *Proc. of Fifth Annual Tcl/Tk Workshop* (1997), USENIX, 169-170.
- [Ullmer and Ishii, 1997] Brygg Ullmer and Hiroshi Ishii, The metaDESK: models and prototypes for tangible user interfaces. In: *Proc. of ACM Symposium on User Interface Software and Technology* (1997), 223-232.
- [Ullmer and Ishii, 2000] Brygg Ullmer and Hiroshi Ishii, Emerging frameworks for tangible user interfaces. *IBM Systems Journal* **39**, 1 (2000), 915-931.
- [Ullmer and Ishii, 2001] Brygg Ullmer and Hiroshi Ishii, Emerging frameworks for tangible user interfaces. In: John M. Carrol (ed.), *Human-Computer Interaction in the New Millennium*. Addison-Wesley, 2001, 579-601.
- [Ullmer et al., 2005] Brygg Ullmer, Hiroshi Ishii and Robert J. K. Jacob, Token+constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction* **12**, 1 (2005), 81-118.

- [Underkoffler and Ishii, 1999] John Underkoffler and Hiroshi Ishii, Urp: a luminous-tangible workbench for urban planning and design. In: *Proc. of Conference on Human Factors in Computing Systems* (1999), 386-393.
- [Waldner et al., 2006] Manuela Waldner, Jörg Hauber, Jürgen Zauner, Michael Haller and Mark Billingham, Tangible Tiles: design and evaluation of a tangible user interface in a collaborative tabletop setup. In: *Proc. of OZCHI* (2006).
- [Weghorst, 2003] Suzanne Weghorst, Augmented tangible molecular models. Presented at International Conference on Humans and Computers (2003), 347-354. Available at: <http://www.hitl.washington.edu/publications//r-2004-11/r-2004-11.pdf>.
- [Weiser, 1991] Mark Weiser, The computer for the 21st century. *Scientific American* **265**, 3 (1991), 94-104.
- [Wellner, 1993] Pierre Wellner, Interacting with paper on the DigitalDesk. *Communications of the ACM* **36**, 7 (1993), 87-97.
- [Wensveen et al., 2004] Stephan Wensveen, J. P. Djajadiningrat and Kees C. J. Overbeeke, Interaction frogger: a design framework to couple action and function through feedback and feedforward. In: *Proc. of Conference on Designing Interactive Systems* (2004), 177-184.
- [Villar and Gellersen, 2007] Nicolas Villar and Hans Gellersen, A malleable control structure for softwired user interfaces. In: *Proc. of Conference on Tangible and Embedded Interaction* (2007), 49-56.
- [Wren and Reynolds, 2004] Christopher R. Wren and Carson J. Reynolds, Minimalism in ubiquitous interface design. *Personal and Ubiquitous Computing* **8**, 5 (2004), 370-373.
- [Zhang, 1997] Jiajie Zhang, The nature of external representations in problem solving. *Cognitive Science* **21**, 2 (1997), 179-217.
- [Zuckerman and Resnick, 2003] Oren Zuckerman and Mitchel Resnick, A physical interface for system dynamics simulation. In: *Proc. of Conference on Human Factors in Computing Systems* (2003), 810-811.
- [Zuckerman and Resnick, 2004] Oren Zuckerman and Mitchel Resnick, Hands-on modeling and simulation of systems. In: *Proc. of Interaction Design and Children* (2004), 157-158.
- [Zuckerman et al., 2005] Oren Zuckerman, Saeed Arida and Mitchel Resnick, Extending tangible interfaces for education: digital montessori-inspired manipulatives. In: *Proc. of Conference on Human Factors in Computing Systems* (2005), 859-868.

**Kyselylomake käyttäjätesteihin osallistuneille**

Nimi: \_\_\_\_\_

Ikä: \_\_\_\_\_

Sukupuoli:

 mies nainen

Vahvempi käsi:

 oikea vasen

Tietokonekokemus:

[aloittelija] 1 2 3 4 5 [kokenut]

Siirtäminen hiirellä oli:

[helppoa] 1 2 3 4 5 [vaikeaa]

Siirtäminen liu'uttamalla oli:

[helppoa] 1 2 3 4 5 [vaikeaa]

Siirtäminen käsin oli:

[helppoa] 1 2 3 4 5 [vaikeaa]

Virheiden huomaaminen hiirellä siirtäessä oli:

[helppoa] 1 2 3 4 5 [vaikeaa]

Virheiden huomaaminen liu'uttamalla siirtäessä oli:

[helppoa] 1 2 3 4 5 [vaikeaa]

Virheiden huomaaminen käsin siirtäessä oli:

[helppoa] 1 2 3 4 5 [vaikeaa]

Mieluiten siirtäisin:

 hiirellä liu'uttamalla käsin

Mitä hyvää oli hiirellä siirtämisessä: \_\_\_\_\_

\_\_\_\_\_

Mitä huonoa oli hiirellä siirtämisessä:

\_\_\_\_\_

\_\_\_\_\_

Mitä hyvää oli liu'uttamalla siirtämisessä:

\_\_\_\_\_

\_\_\_\_\_

Mitä huonoa oli liu'uttamalla siirtämisessä:

\_\_\_\_\_

\_\_\_\_\_

Mitä hyvää oli käsin siirtämisessä:

\_\_\_\_\_

\_\_\_\_\_

Mitä huonoa oli käsin siirtämisessä: \_\_\_\_\_

\_\_\_\_\_