

# Tietokoneavusteinen yhteiskirjoittaminen

Petri Gerdt

12.11.2001

Joensuun yliopisto

Tietojenkäsittelytiede

Pro gradu -tutkielma

# Tiivistelmä

Nykyaikana suuri osa yrityksissä ja yliopistomaailmassa tuotettavista dokumenteista kirjoitetaan ryhmätyönä. Tietokoneavusteinen yhteiskirjoittaminen on tietokoneavusteisen ryhmätyön eräs muoto, joka pyrkii tukemaan yhteistä dokumenttia työstävää ryhmää. Tässä tutkielmassa tutustutaan tietokoneavusteiseen yhteiskirjoittamiseen kirjallisuuden pohjalta. Tutkielman tavoitteena on antaa lukijalle mielikuva yhteiskirjoittamisesta, yhteiskirjoitusvälineen toteuttamiseen liittyvistä asioista ja olemassa olevien yhteiskirjoitusvälineiden piirteistä.

*ACM-luokat* (ACM Computing Classification System, 1998 version): H.5.3, I.7.m, I.7.2, I.7.1, H.4.3, J.7

*Avainsanat:* tietokoneavusteinen yhteiskirjoittaminen, tietokoneavusteinen ryhmätyö, tekstinkäsittely

## **Esipuhe**

Aloitin tämän tutkielman kirjoittamisen 1.1.2001. Yksitoista kuukautta ja kaksitoista päivää myöhemmin tutkielmani on viimein valmis. Itse kirjoitusprosessi on ollut hyvin rikas kokemus, josta koen oppineeni paljon. Tutkielmani kirjoittamiseen olen saanut asiantuntevaa apua ja tukea ohjaajaltani Professori Marja Kuittiselta. Professori Erkki Sutinen on myös tukenut ja rohkaissut minua pitkän kirjoitustaipaleeni varrella. Olen erityisen kiitollinen siitä, että olen saanut työstää tutkielmaani osana työtehtäviäni työskennellessäni tutkijana Joensuun yliopiston Tietojenkäsittelytieteen laitoksella Ajattelun välineitä tietoverkkoon -projektissa Professori Erkki Sutisen alaisena.

Lopuksi haluan kiittää Marjaanaa hänen tuestaan ja ymmärryksestään välillä myrskyisänkin kirjoitusrupeamani aikana.

## Tutkielmassa esiintyvät lyhenteet

### Lyhenne

CSCW

URL

XML

WYSIWYG

WYSIWIS

WWW

### Selitys

Computer Supported Collaborative Work

Universal Resource Locator

eXtended Markup Language

What You See Is What You Get

What You See Is What I See

World Wide Web

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Yhteiskirjoittamisen perusteita</b>	<b>5</b>
2.1	Yhteinen dokumentti . . . . .	7
2.1.1	Perustekstinkäsittely . . . . .	8
2.1.2	Käyttöliittymän kytkentä . . . . .	10
2.1.3	Versionhallinta . . . . .	13
2.2	Koordinointi . . . . .	19
2.2.1	Roolit ja sosiaaliset protokollat . . . . .	20
2.2.2	Kirjoitusprosessi . . . . .	25
2.2.3	Kirjoitussessio . . . . .	28
2.3	Kommunikaatio . . . . .	30
2.3.1	Tekstipohjainen keskustelu . . . . .	34
2.3.2	Huomautukset ja kommentit . . . . .	39
2.3.3	Osoitinvälineet ryhmätyöohjelmissa . . . . .	41
2.4	Tietoisuus . . . . .	42
2.4.1	Tietoisuus ryhmätyöohjelmissa . . . . .	44
2.4.2	Käyttöliittymä tietoisuutta tukemassa . . . . .	45
2.5	Yhteen veto . . . . .	48
<b>3</b>	<b>Yhteiskirjoitusvälineen toteuttaminen</b>	<b>52</b>
3.1	Järjestelmäarkkitehtuuri . . . . .	53
3.1.1	Keskitetty arkkitehtuuri . . . . .	56
3.1.2	Toistettu arkkitehtuuri . . . . .	59
3.1.3	Hybridiarkkitehtuuri ja muita arkkitehtuuriratkaisuja . . . . .	62
3.1.4	Vetoketjumalli . . . . .	66
3.2	Yhtenäisyyden hallinta . . . . .	70
3.2.1	Lukitseminen . . . . .	73
3.2.2	Sarjoittaminen . . . . .	75
3.2.3	Operationaalinen transformaatio . . . . .	76

3.3	Ryhmätyöohjelmien kehitystyökalut . . . . .	77
3.3.1	Suite . . . . .	79
3.3.2	Rendezvous . . . . .	81
3.3.3	GroupKit . . . . .	82
3.4	Yhteenveto . . . . .	85
<b>4</b>	<b>Esimerkkijärjestelmiä</b>	<b>89</b>
4.1	PREP . . . . .	91
4.1.1	Yhteinen dokumentti . . . . .	93
4.1.2	Koordinointi . . . . .	95
4.1.3	Kommunikointi . . . . .	96
4.1.4	Tietoisuus . . . . .	97
4.1.5	Järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta . . . . .	99
4.2	Calliope . . . . .	99
4.2.1	Yhteinen dokumentti . . . . .	101
4.2.2	Koordinointi . . . . .	101
4.2.3	Kommunikointi . . . . .	102
4.2.4	Tietoisuus . . . . .	104
4.2.5	Järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta . . . . .	105
4.3	Muita järjestelmiä . . . . .	107
4.3.1	DCWA-yhteiskirjoitusvälineen dokumentin esitys . . . . .	107
4.3.2	Quiltin roolihierarkkia . . . . .	108
4.4	Yhteenveto . . . . .	109
<b>5</b>	<b>Tietokoneavusteisen yhteiskirjoittamisen tulevaisuus</b>	<b>112</b>
5.1	Tulevaisuuden yhteiskirjoitusväline . . . . .	112
5.1.1	Yhteinen dokumentti . . . . .	114
5.1.2	Kommunikaatio . . . . .	116
5.1.3	Koordinaatio . . . . .	119
5.1.4	Tietoisuus . . . . .	119
5.1.5	Järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta . . . . .	120
5.2	Tutkimuskohteita tietokoneavusteisen yhteiskirjoittamisen parissa	122

<b>6 Loppupäätelmät</b>	<b>125</b>
<b>Viitteet</b>	<b>127</b>
<b>Liite 1: Sanasto</b>	<b>136</b>

# 1 Johdanto

Nykyisin suuri osa tekstidokumenteista kirjoitetaan tekstinkäsittelyohjelmilla. Globalisaation myötä erilaiset organisaatiot ovat yhä useammin hajasijoitettuja ympäri maailmaa. Yrityksissä ja yliopistomaailmassa tietyllä dokumentilla saattaa olla useita kirjoittajia, esimerkiksi yrityksen tilannetiedotuksessa tai akateemisessa artikkelissa. Fish *et al.* (1988) toteavat, että joillakin tieteellisillä aloilla jopa 65 % dokumenteista on yhteiskirjoitettuja ja Baecker *et al.* (1994) väittävät, että yrityksissä ja akateemisissa piireissä 85 % tuotetuista dokumenteista kirjoitetaan ryhmätyönä. Vaikka tietokoneita hyödynnetään yhteiskirjoitusprojekteissa, suurin osa kirjoitusryhmän kommunikaatiosta ja koordinaatiosta hoidetaan ilman tietokoneiden tukea kierrättämällä paperitulosteita dokumentin vedoksista ja pitämällä palavereja (Hofte, 1996).

Tietokoneavusteisen ryhmätyön tutkimuksen katsotaan alkaneen nykyisessä muodossaan vuonna 1984, jolloin Paul Cashman ja Irene Grief organisoivat CSCW'84-konferenssin, jossa lanseerattiin käsite computer supported collaborative work (CSCW) (Grudin, 1994). *Tietokoneavusteinen ryhmätyö* on monitieteinen tutkimusalue, jossa tutkitaan tietokoneavusteista yhteistyötä. Ellis *et al.* (1991) nimeävät viisi eri näkökulmaa, jotka on huomioitava tietokoneavusteisen ryhmätyön yhteydessä: hajautetut järjestelmät, kommunikaatio, ihmisen ja tietokoneen välinen vuorovaikutus, tekoäly ja sosiologia. Yhteistyöllä tarkoitetaan tässä yhteydessä sitä, että ryhmä ihmisiä työstää tuotosta tai palvelua yhdessä, heillä on yhteinen päämäärä, jonka eteen he tekevät työtä (Bannon ja Schmidt, 1991). *Ryhmätyöohjelmat* ovat ryhmätyötä tukevia tietokoneohjelmia, jotka toteuttavat tietokoneavusteisen ryhmätyön periaatteita. Tietokoneavusteisen ryhmätyön piirissä tutkitaan yleisesti useita erilaisia sovelluskohteita, kuten prosessin ohjausta ja hallintaa, ajoittamista ja virtuaalisten konferenssien tukemista.

Tässä tutkielmassa rajoitun erääseen tietokoneavusteisen ryhmätyön muotoon, tietokoneavusteiseen yhteiskirjoittamiseen. *Yhteiskirjoitusvälineet* ovat ryhmätyöohjelmia, jotka tukevat useasta henkilöstä koostuvaa ryhmää dokumenttia tuottaes-



sa (Hofte, 1996). Yhteiskirjoitusvälineistä käytetään myös nimitystä ryhmäeditori (Ellis *et al.*, 1991).

Ryhmätyöohjelmat jaetaan tyypillisesti niiden tukeman työskentelymallin mukaan kahteen kategoriaan, *synkronisiin*, joissa ryhmän jäsenet työskentelevät samanaikaisesti ja *asynkronisiin*, joissa ryhmän jäsenet työskentelevät eri aikoihin (Grudin, 1994). Useimmat nykyisistä yhteiskirjoitusvälineistä tukevat sekä synkronista että asynkronista työskentelyä. Yhteiskirjoittaminen edellyttää välillä ryhmän jäseniltä samanaikaista työskentelyä. Toisinaan on luontevaa työskennellä eri aikoihin, muista ryhmän jäsenistä riippumatta.

Ryhmätyöohjelmien yhteydessä käytetään usein termiä *kytkentä*, joka kuvaa kirjallisuudessa melko ristiriitaisesti käyttäjien välisen yhteistyön tukemisen astetta sekä ryhmätyöohjelmien komponenttien välistä suhdetta. Erottaakseni nämä kaksi eri asiaa toisistaan käytän tässä tutkielmassa termiä *käyttäjien välinen kytkentä* merkitsemään käyttäjien välisen yhteistyön tukemisen astetta. Termillä *komponenttien välinen kytkentä* tarkoitan ryhmätyöohjelman komponenttien välisiä yhteyksiä ja sidoksia.

Tässä tutkielmassa keskityn aitoihin yhteiskirjoitusvälineisiin, jotka ovat nimenomaan kirjoitusryhmän tarpeita silmällä pitäen suunniteltuja sovelluksia. Yhteiskirjoittaminen on eräs tutkituimmista tietokoneavusteisen ryhmätyön alueista, koska se tarjoaa hyvin määritellyn tapauksen. Yhteiskirjoittaminen on useimmiten selkeä yhteistyötapa, jossa suhteellisen pieni 2-5 henkilön ryhmä pyrkii samaan selkeästi määriteltyyn tavoitteeseen. Tyypillinen, usein kirjallisuudessa esiintyvä esimerkki yhteiskirjoitustapauksesta on tieteellisen artikkelin kirjoittaminen. Pieni ryhmä motivoituneita kirjoittajia kirjoittaa tietyllä aikataululla dokumenttia, jonka aihe ja sisältö on pääpiirteittäin selvillä alusta lähtien.

Pureudun tietokoneavusteiseen yhteiskirjoittamiseen kolmesta toisiaan tukevasta näkökulmasta. Aluksi käsittelen luvussa 2 tietokoneavusteista yhteiskirjoittamista käyttäjän tai yhteiskirjoitusvälineen suunnittelijan näkökulmasta. Luvun 2 tarkoituksena on antaa lukijalle mielikuva siitä, mitä tietokoneavusteisen yhteiskirjoit-

tamisen avulla pyritään saavuttamaan.

Tarkastelen tietokoneavusteista yhteiskirjoittamista toteutusnäkökulmasta luvussa 3. Johdattelen lukijaa yhteiskirjoitusvälineen toteuttamisen kannalta keskeisiin kysymyksiin. Tavoitteenani on antaa lukijalle näkemys niistä asioista, joihin yhteiskirjoitusvälineen toteuttajan on perehdyttävä. Luku 3 syventää luvussa 2 esitettyjä asioita täydentämällä siinä esitettyä käyttäjän näkökulmaa teknisemmällä näkemyksellä tietokoneavusteisesta yhteiskirjoittamisesta.

Luvussa 4 esittelen kaksi yhteiskirjoitusvälinettä, jotka edustavat hyvinkin erilaisia näkemyksiä yhteiskirjoitusvälineen toteuttamisesta. Käyn esimerkkijärjestelmistä läpi niitä asioita, joita olen käsitellyt luvuissa 2 ja 3. Luvun 4 tavoite on auttaa lukijaa sitomaan aiemmissa luvuissa esittelemäni teoreettisemmat kysymykset käytännön sovelluksiin. Luku 4 näyttää myös sen, kuinka eri lailla yhteiskirjoittamista on mahdollista tukea tietokoneitse.

Valitsemiani kolmea eri näkökulmaa voi luonnehtia myös yhteiskirjoitusvälineen suunnittelijan (luku 2), toteuttajan (luku 3) ja arvioijan (luku 4) näkökulmiksi tietokoneavusteisesta yhteiskirjoittamisesta. Tietojenkäsittelytieteessä kaikki edellä mainitsemani näkökulmat ovat tärkeitä tietyn tietojenkäsittelyn sovelluskohteen ymmärtämiseksi. Tutkielmani tarkoituksena on johdatella lukija tietokoneavusteisen yhteiskirjoittamisen saloihin tietojenkäsittelytieteen näkökulmasta.

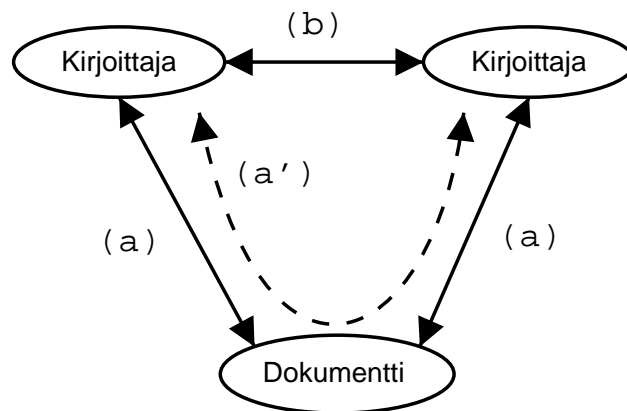
Luvussa 5 esitän omia ajatuksiani tietokoneavusteisen yhteiskirjoittamisen tulevaisuudesta. Pohdiskelen tulevaisuuden yhteiskirjoitusvälineen piirteitä ja esitän sekä kirjallisuudesta löytämiäni että itse keksimiäni jatkotutkimusaiheita tietokoneavusteisen yhteiskirjoittamisen liittyen. Tutkielmani päättää luvun 6 loppupäätelmät.

Tutkielman loppuun olen liittänyt kaksi liitettä: sanaston ja käyttämieni lyhenteiden selityksiä. Tietokoneavusteisesta yhteiskirjoittamisesta ei ole kirjoitettu juuri mitään suomeksi, ainoastaan yksi lähteistäni on suomenkielinen: Kari Kuutin luentomoniste (Kuutti, 1998). Olen poiminut Kuutin luentomonisteesta useita

suomennoksia englanninkielisille tietokoneavusteisen ryhmätyön ja yhteiskirjoittamisen yhteydessä käyttämistäni termeistä ja osalle termeistä olen kehittänyt itse suomennoksen. Juuri tästä syystä olen katsonut tarpeelliseksi antaa lukijalle mahdollisuuden tarkastaa liitteestä 1 vastineet osalle tutkielmassani esiintyvillä termeillä. Olen sijoittanut joidenkin termien englannin kielisiä vastineita niiden määritelmien yhteyteen, koska niillä ei ole merkitystä määrittelykontekstin ulkopuolella eikä niitä löydy yleisesti kirjallisuudesta.

## 2 Yhteiskirjoittamisen perusteita

Yhteiskirjoittaminen on tapahtuma, jossa usea kirjoittaja työstää samaa dokumenttia. Hofte (1996) esittää mallin yhteiskirjoittamiselle, jota kuva 1 esittää. Kuvassa suorat viivat kuvaavat vuorovaikutusta ja katkoviiva metatason tai epäsuoraa vuorovaikutusta. Ilmeisin yhteiskirjoittamisen yhteydessä esiintyvä vuorovai-  
kutustapahtuma toteutuu, kun kirjoittaja muokkaa dokumenttia (a). Kirjoittaja saa käyttämältään välineeltä palautetta, joka ilmaisee, kuinka hänen muutoksensa dokumentissa näkyvät, eli muokkaustapahtuma on kaksisuuntaista vuorovaikutusta. Toinen vuorovaikutustyyppi yhteiskirjoittamisessa on ryhmän jäsenten, kuvassa kirjoittajien, välinen keskustelu (b). Metatason vuorovaikutusta (a') tapahtuu silloin, kun kirjoitusryhmän jäsenet havainnoivat jonkun kirjoittajan tekemiä muutoksia dokumentissa. Muille kirjoittajille tietyn kirjoittajan muokkaustoimenpiteistä koituva palaute on epäsuoraa palautetta, joka on kohdistettu tietylle kirjoittajalle, mutta josta muutkin hyötyvät.

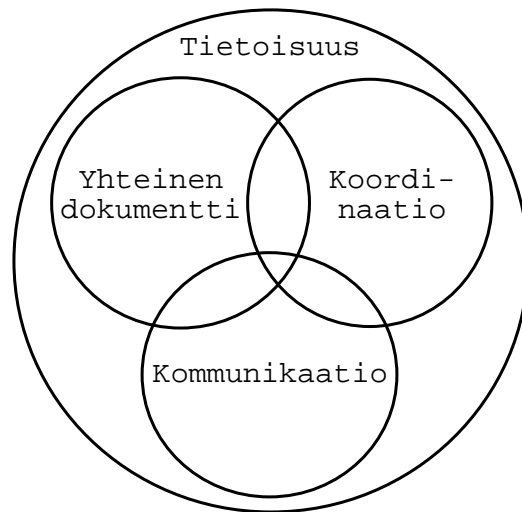


Kuva 1: Yhteiskirjoittamisen käsitteellinen malli (Hofte, 1996).

Kuvaa 1 tutkimalla selviää, että yhteiskirjoittamisessa tärkeässä asemassa on kirjoittajien kesken jaettu yhteinen dokumentti, jota kirjoitusryhmä työstää. Toinen

ilmeinen yhteiskirjoittamisessa esiintyvä asia on kirjoittajien välinen kommunikaatio, onhan kirjoitusryhmän päätettävä, mitä on tarkoitus tehdä ja miten tavoitteisiin pyritään. Jotta kirjoitusryhmän jäsenet voisivat huomioida toisiaan epäsuorasti, on heidän saatava tietoa muista ryhmän jäsenistä ja olla tietoisia kirjoitusryhmässä tapahtuvista asioista. Jos kirjoitusryhmän jäsenillä on mahdollisuus tavata kasvotusten, on näiden keskeisten tekijöiden, eli yhteisen dokumentin, kommunikaation, koordinaation ja ryhmän jäsenten tietoisuuden toisistaan, olemassaolo itsestäänselvyys.

Kuva 2 havainnollistaa edellisessä kappaleessa mainittujen tekijöiden suhdetta toisiinsa. Yhteisen dokumentin kirjoittaminen vaatii ryhmältä työn koordinoimista. Toisaalta ryhmä koordinoi tekemisiään keskustelemalla ja sopimalla asioista eli kommunikoimalla. Ryhmän voi olla vaikeaa kommunikoida, elleivät he pääse käsiksi yhteiseen dokumenttiin. Onhan dokumentti keskeinen tekijä yhteiskirjoitusprojektissa. Tietoisuus kattaa kuvassa 2 kaikki muut kategoriat, koska ryhmän jäsenten tulee olla tietoisia siitä, mitä yhteiskirjoitushankkeessa tapahtuu. Heidän on tunnettava koordinaatioseikat sekä tiedostettava dokumentin tila ja kommunikaatio, jota ryhmässä tapahtuu.



Kuva 2: Neljä yhteiskirjoittamisen tekijää, joita yhteiskirjoitusvälineen on tuettava.

Kirjoitusryhmä voi hyödyntää yhteiskirjoitusvälinettä, jos he eivät jostain syystä voi työskennellä kasvotusten. Yhteiskirjoitusväline voi tukea kirjoitusryhmää kaikkien kuvan 2 tekijöiden suhteen. Käytän tässä luvussa edellä mainittuja tekijöitä jakaakseni yhteiskirjoitusvälineiden toiminnot ja piirteet neljään eri kategoriaan, joita ovat yhteinen dokumentti, kommunikointi, koordinointi ja tietoisuus. Vastaavanlaisia yhteiskirjoitusvälineiden luokitteluja löytyy myös kirjallisuudesta: Hofte (1996) puhuu jaetusta dokumentista, keskustelutilasta, koordinaatiosta ja (kirjoitus-) konferenssin hallinnasta. Sohlenkamp ja Chwelos (1994) puolestaan jakavat yhteiskirjoitusvälineen piirteet kolmeen kategoriaan: kommunikaatioon, yhteistyön tukemiseen ja tietoisuuden tukemiseen. Tässä luvussa käyttämäni yhteiskirjoittamisen jako neljään kategoriaan perustuu pitkälti hieman mukailtuun Hoften (1996) käyttämään jakoon, jota on terästetty Sohlenkampin ja Chwelosin (1994) tietoisuuskategoriolla. Seuraavaksi tutustumme jokaiseen kategoriaan tarkemmin niille omistetuissa alakohdissa.

## 2.1 Yhteinen dokumentti

Yhteinen dokumentti on yhteiskirjoittamisen perusta. Yhteiskirjoitusvälineen tulee mahdollistaa tavallisista tekstinkäsittelyohjelmista tutut tekstin muokkaustoimenpiteet, eli kirjoittajan ja dokumentin välinen vuorovaikutus (a kuvassa 1). Tutustumme yhteiskirjoitusvälineeltä vaadittaviin perustekstinkäsittelyvalmiuksiin alakohdassa 2.1.1. Ellis *et al.* (1991) määrittelevät ryhmätyöohjelmat tietokonejärjestelmiksi, jotka tukevat yhteisen päämäärän tai tuotoksen parissa työskentelevää ryhmää tarjoamalla heille yhteisen rajapinnan jaettuun ympäristöön. Yhteinen rajapinta jaettuun ympäristöön, yhteiskirjoittamisen tapauksessa yhteiseen dokumenttiin, tarkoittaa käytännössä ryhmätyöohjelman käyttöliittymää. Tarkastelemme alakohdassa 2.1.2, kuinka yhteiskirjoitusvälineen käyttöliittymä poikkeaa tavallisen tekstinkäsittelyohjelman käyttöliittymästä. Versionhallinnan merkitys tuotoksen hallinnassa korostuu usean henkilön työstäessä tuotosta. Alakohdassa 2.1.3 perehdymme versionhallintaan, jolle asetetaan käyttöliittymän tavoin yhteiskirjoitusvälineessä uusia, yhden käyttäjän järjestelmissä tuntemattomia haas-

teita. Lopuksi esitän luvun sisällöstä yhteenvedon alakohdassa 2.5.

### **2.1.1 Perustekstinkäsittely**

Yhteiskirjoitusvälineen perusvaatimuksena on luonnollisesti hyvät tekstinkäsittelyvalmiudet. Eräs syy tietokoneavusteisten ryhmätyövälineiden käyttämättömyyteen on niiden kankeus verrattuna henkilökohtaisiin, yksin käytettäviin ohjelmistoihin (Grudin, 1988). Neuwirth *et al.* (1990) huomioivat, että vaikeakäyttöinen ja huonosti yhteensopiva yhteiskäyttöinen kirjoitusohjelma johtaa usein siihen, että yksi teknisesti kyvykäs ryhmän jäsen käyttää tietokonevälinettä ja muut ryhmän jäsenet toimittavat hänelle tuotoksiaan järjestelmän ulkopuolisin keinoin. Tässä tilanteessa tietokoneen käyttö ei helpota ryhmätyötä vaan pikemminkin lisää työmäärää ja kangistaa koko prosessia. Yhteiskirjoitusjärjestelmän tulisikin olla yhteensopiva käytössä olevan tietokoneympäristön kanssa. Myös Baecker *et al.* (1993) ovat samaa mieltä. Heidän mielestään yhteiskirjoitusvälineen vähimmäisvaatimuksena on tavallisten tekstinkäsittelyohjelmien kanssa yhteensopivien dokumenttien tuottaminen. Yleisemmin yhteiskirjoitusvälineen tulee toimia saumattomasti muiden medioiden kanssa (Baecker *et al.*, 1993) eikä pelkkä dokumenttien yhteensopivuus välineiden välillä riitä. Myös tulostuksen, leikepöytämekanismin käyttö ja grafiikan lisäämisen tekstiin tulee olla mahdollista yhteiskirjoitusvälineellä. Taulukossa 1 on yhteenvedo yhteiskirjoitusvälineen, myös yhteiskirjoitusvälineen, perustoiminnoista käyttäjän kannalta katsottuna.

Taulukko 1: Tekstinkäsittelyjärjestelmän perustoimintoja.

<b>Toiminto</b>	<b>Huomioita ja selityksiä</b>
kopioi	Käyttäjä voi kopioida valitsemansa osan dokumentista leikepöydälle.
leikkaa	Käyttäjä voi leikata valitsemansa osan dokumentista leikepöydälle.
liimaa	Käyttäjä voi liimata leikepöydän sisällön haluamaansa kohtaan dokumentissa.
peru	Käyttäjä voi perua viimeisimmät suorittamansa toiminnot.
fontin valitseminen	Käyttäjä voi valita useasta fontista mieleisensä.
tekstin korostus	Käyttäjä voi korostaa tekstiä kursivoimalla, lihavoimalla ja alleviivaamalla.
talletus, avaus	Järjestelmä tukee useita eri tiedostomuotoja, varsinkin yleisesti käytössä olevia yleisformaatteja.
tulostus	Järjestelmällä voi tulostaa dokumentin yleisesti käytössä olevilla tulostimilla.

Neuwirth *et al.* (1990) nimeävät permeabiliteetin yhteensopivuuden ohella erääksi yhteiskirjoitusvälineen keskeiseksi vaatimukseksi. Permeabiliteetillä he tarkoittavat järjestelmän kykyä siirtää tuotos sähköisestä muodosta tulosteeseen ja päinvastoin. Neuwirth *et al.* perustelevat permeabiliteetin vaatimustaan huomiolla siitä, että järjestelmä, joka vaatii käyttäjiltä ylimääräistä työtä, jää käyttämättä (Grudin, 1988). Käyttäjä saattaa esimerkiksi tulostaa junamatkalle dokumentin arvioitavaksi ja haluaa työpaikalle palatessaan saattaa sähköisen version matkalla muokkaamansa tulosteen tasalle minimityöllä. Neuwirth *et al.*:n mielestä yhteiskirjoitusjärjestelmän tulee toimia myös siinä tilanteessa, että joku ryhmän jäsenistä ei voi itse käyttää järjestelmää vaan hänelle välitetään ryhmän tuotos vaihtoehtoisella medially, kuten paperilla tai vaikka ääniviestinä, johon hän vastaa muulla kuin



sähköisellä medialla.

Yhteiskirjoitusvälineen tulee tarvittaessa toimia kaikin puolin samoin kuin tavallinen tekstinkäsittelyohjelma (Neuwirth *et al.*, 1990; Baecker *et al.*, 1993; Koch, 1995). Vaikka yhteiskirjoitussovellus tukeekin dokumentin kirjoittamista ryhmätyönä, on sen kuitenkin toimittava myös yhden käyttäjän järjestelmänä. Ryhmätyöskentelyssä on hetkiä, jolloin ryhmän jäsenet toimivat itsenäisesti (katso alakohta 2.2.2). Jos ryhmäeditori on huonompi vaihtoehto tekstinkäsittelyyn kuin tavallinen tekstinkäsittelyohjelma, saattaa syntyä tilanne, jossa kirjoittajat kirjoittavat tekstinsä muilla ohjelmilla. Yhteiskirjoitusväline jää silloin ainoastaan tuotetun tekstin jakamisvälineeksi. Tässä tilanteessa ryhmäeditorin käyttö ei lainkaan ole perusteltua eikä ryhmä saa lisäarvoa työskentelyynsä siitä.

### **2.1.2 Käyttöliittymän kytkentä**

Yhteiskirjoitusvälineen käyttöliittymä muistuttaa ensisilmäykseltä tavallisen tekstinkäsittelyohjelman käyttöliittymää. Yhteiskirjoitusvälineen on kuitenkin tuettava myös ryhmätyöskentelyä, jolloin käyttöliittymälle asetetaan joukko uusia vaatimuksia. Ellis *et al.* (1991) määrittelevät ryhmätyövälineen tietokoneavusteiseksi rajapinnaksi yhteiseen tuotokseen. On selvää, että kaikkien ryhmätyöhön osallistuvien henkilöiden on pystyttävä käsittelemään samaa ryhmän kesken jaettua tietoa heidän rooliensa puitteissa (rooleista lisää alakohdassa 2.2.1) . Mutta pitäisikö käyttäjien jakaa myös täsmälleen sama näkymä? Vastaus on kyllä ja ei. Tietyissä ryhmätyön vaiheissa käyttäjien saattaa olla tarpeellista nähdä täsmälleen sama näkymä, kun taas toisissa vaiheissa heidän työskentelynsä edellyttää henkilökohtaisia näkymiä yhteiseen tuotokseen. Esimerkiksi alakohdassa 2.2.2 käsittelemässäni yhteiskirjoittamisprosessissa työskentelyn intensiteetti vaihtelee tiiviistä ryhmätyöskentelystä itsenäiseen, muista riippumattomaan työskentelyyn.

Erityisesti tavallisten tekstinkäsittelyohjelmien parissa käytetään yleisesti termiä What You See Is What You Get (WYSIWYG), joka merkitsee vapaamuotoisesti suomennettuna “mitä näet, sen saat”. Tekstinkäsittelykontekstissa WYSIWYG

merkitsee tietokoneen ruudulla näkyvän dokumentin tulostumista täsmälleen saman näköisenä paperille. Stefik *et al.* (1987) määrittelevät ryhmätyöohjelmille WYSIWYG-käsitettä vastaavan käsitteen What You See Is What I See (WYSIWIS), jonka vapaamuotoinen suomennos on “sen minkä minä näen sinäkin näet”. WYSIWIS-käyttöliittymä ylläpitää kaikille käyttäjille illuusiota täsmälleen samasta näkymästä yhteiseen tuotokseen välittämällä kaikki käyttöliittymätapahtumat kaikkien käyttäjien käyttöliittymälle. WYSIWIS-tilanteessa käyttöliittymät on kytketty toisiinsa mahdollisimman tiukasti ja niiden välillä on niin vähän eroavaisuuksia kuin mahdollista.

WYSIWIS ei kuitenkaan aina välttämättä ole tarpeellista yhteiskirjoittamisessa ja ryhmätyössä yleensä. Haake ja Wilson (1992) esittelivät SEPIA-järjestelmän yhteydessä uuden tavan kytkeä käyttöliittymät. SEPIA-yhteiskirjoitusjärjestelmän käyttäjät voivat valita tiukan tai löyhän käyttöliittymien kytkennän. Tiukka kytkentä vastaa WYSIWIS-käyttöliittymää, mutta löyhä kytkentä mahdollistaa käyttäjien käyttöliittymien eroavuuksia. Löyhästi kytketyillä käyttöliittymillä näkyy ainoastaan osa käyttöliittymätapahtumista, jolloin käyttäjän on helpompi syventyä itsenäiseen työskentelyyn. Baecker *et al.* (1993) puhuvat periferisestä ja fokusoidusta tietoisuudesta viitattaessaan tiukempaan ja löyhempään käyttöliittymien kytkentään. Järjestelmän käyttäjän ei tarvitse aina olla tietoinen kaikesta, mitä muut järjestelmän puitteissa tekevät, koska liika informaatio saattaa haitata työskentelyä. Toisaalta joskus kaikki mahdollinen informaatio on tarpeen. Käsittelen ryhmän jäsenten tietoisuutta toisistaan tarkemmin kohdassa 2.4.

Kytkenän muuttamista tiukasta WYSIWIS:stä höllempään voidaan myös luonnehtia käyttöliittymän rentouttamiseksi (Stefik *et al.*, 1987; Hofte, 1996). Stefik *et al.* (1987) ehdottavat neljää eri ulottuvuutta, joissa WYSIWIS-käyttöliittymää on mahdollista rentouttaa: tila, aika, populaatio ja kongruenssi. Yhteisen tuotoksen kaikki objektit, yhteiskirjoitusvälineessä esimerkiksi dokumentti ja siihen liittyvät kommentit, näkyvät tiukassa WYSIWIS-käyttöliittymässä samalla tavalla kaikille käyttäjille. Käyttöliittymän rentouttaminen tilan suhteen antaa käyttäjille mahdollisuuden ottaa esiin vain tarvitsemansa osat yhteisestä tuotoksesta, esi-

merkiksi dokumentin ensimmäisen puoliskon ja tietyn käyttäjän kommentit koko dokumentin ja kaikkien käyttäjien kommenttien sijasta.

Tiukan WYSIWIS:in tapauksessa kaikki muutokset dokumenttiin näkyvät heti muille käyttäjille. Jos käyttöliittymää rentoutetaan ajan suhteen, voi järjestelmä lähettää palautetta muille käyttäjille tietyin väliajoin (Stefik *et al.*, 1987). Ajan suhteen rentouttaminen saattaa olla hyödyllistä silloin, kun käyttäjien on oltava muista tietoisia, mutta he eivät tarvitse tarkkaa tietoa. Jos ryhmätyöohjelmaa rentoutetaan aikaulottuvuuden suhteen äärimmilleen, syntyy täysin asynkroninen järjestelmä (Hofte, 1996). Käsittelen alakohdassa 4.1.2 PREP-järjestelmän vuorovaiutusparametrejä, jotka antavat käyttäjien rentouttaa WYSIWIS-näkymää.

Tiukassa WYSIWIS:ssä kaikki käyttäjät näkevät saman jaetun näkymän. Tätä rajoitusta on mahdollista höllentää antamalla käyttäjien näkymien poiketa toisistaan. Stefik *et al.* (1987) nimeävät tämän rentouttamisulottuvuuden populaatioksi. Populaatioulottuvuuden rentouttaminen antaa mahdollisuuden esimerkiksi käyttäjien jakamisen pienempiin työryhmiin, jotka työskentelevät omien tehtäviensä parissa. Viimeinen ulottuvuus, kongruenssi, liittyy tiedon esittämiseen (Stefik *et al.*, 1987). WYSIWIS ei salli erilaisia tiedon esityksiä eri käyttäjille. Joskus on kuitenkin hyödyllistä sallia eri käyttäjille toisistaan poikkeavia näkymiä, esimerkiksi käyttäjän roolien perusteella, mitä käsittelen alakohdassa 2.2.1.

Käyttöliittymän rentouttamiseen liittyy riski siitä, että ryhmän yhteistyökyky heikkenee ryhmän jäsenten menettäessä yhteyden toisiinsa (Begole, 1998). Esimerkiksi alakohdassa 4.1.2 käsittelemäni PREP-yhteiskirjoitusvälineen piirre antaa käyttäjälleen mahdollisuuden määritellä, mitä tietoa hän vastaanottaa muilta ja mitä tietoa muut saavat hänen toiminnastaan. Väärin käytettynä edellä mainittu toiminto johtaa siihen, ettei ryhmä saa toistensa tekemisistä ryhmätyön kannalta välttämätöntä tietoa.

Hofte (1998) luettelee kolme erilaista tapaa, joilla yhteiskirjoitusväline voi suhtautua käyttöliittymien kytkentään. Kytkentä voi olla sisäänrakennettu järjestelmään, jolloin käyttäjät eivät voi vaikuttaa siihen, vaan heidän on tyydyttävä siihen

informaatiomäärään, minkä järjestelmä heille välittää. Järjestelmä voi mahdollistaa sessiokohtaisen (2.2.3) kytkennän asteen määrittelemisen, jolloin on kyseessä staattinen kytkentä. Staattisen kytkennän tapauksessa käyttäjät voivat määritellä kytkennän asteen sessiokohtaisesti sopivaksi. Dynaaminen kytkennän käsittely antaa käyttäjälle mahdollisuuden muuttaa kytkennän astetta tarvittaessa työskentelyn intensiteetin mukaisesti.

### 2.1.3 Versionhallinta

Ryhmätyöohjelmissa versionhallintaa tarvitaan hallitsemaan rinnakkaisesta työskentelystä aiheutuvia, osittain päällekkäisiäkin tuotosten versioita (Haake ja Haake, 1993; Neuwirth *et al.*, 1992). Itsenäisesti työskentelevä ryhmän jäsen voi hyödyntää versioiden vertailutyökalua nähdäkseen, miten dokumentti on muuttunut hänen edellisen työskentelyjaksonsa jälkeen. Versiohistorian tuntemus edistää myös ryhmän jäsenen tietoutta oman työnsä ja ryhmän työn kontekstista (Haake ja Haake, 1993).

Käyttäjän kannalta katsottuna versionhallinta on eräs keino hallita kehittyvää tuotosta. Kun tuotosta on kehitetty tiettyyn pisteeseen asti, se määritellään jollain tapaa nimetyksi versioksi, jota on mahdollista verrata aiempiin ja tuleviin versioihin. Tuotoksen uusin versio on tyypillisesti se, jota muokataan. Aiemmat versiot ovat jäädytettyjä, eli niitä ei voi enää muuttaa. Dewan ja Munson (1995) toteavat, että ryhmätyöohjelman, joka tukee useita versioita, on tuettava version ottamista muokattavaksi (check-out), version palauttamista versionhallinnan piiriin (check-in) sekä kahden version sulauttamista yhdeksi versioksi (merge). Mainituista operaatioista kaksi ensimmäistä löytyy kaikista versionhallintaohjelmista ja yhdessä ne mahdollistavat alkeellisen versionhallinnan. Versioiden sulauttaminen on jo erikoisempi toiminto, joka on tärkeä ryhmätyöohjelmistojen versionhallinnassa.

Versioiden sulauttamisella tarkoitetaan usean dokumentin version yhdistämistä uudeksi versioksi (Lee *et al.*, 1998). Uusi versio on aiemmissa versioissa tehtyjen muokkaustoimenpiteiden unioni (Neuwirth *et al.*, 1992). Jos useassa sulautetta-

vassa versiossa on muokattu samaa kohtaa, tapahtuu yhteentörmäys, konfliktitilanne, jossa on ratkaistava, mitkä muutokset uuteen versioon siirretään. Konfliktitilanteissa useimmat järjestelmät antavat päätösvallan käyttäjille, koska järjestelmän on vaikea päättää, mikä on paras mahdollinen ratkaisu. Perinteinen ratkaisu versioiden sulauttamiskonflikteihin on antaa ainoastaan yhden käyttäjän ottaa tuotoksen versio muokattavaksi kerrallaan (Persson, 1998). Tämä piirre estää konfliktitilanteet, mutta tekee usean käyttäjän samanaikaisen työskentelyn vaikeaksi, ellei mahdottomaksi.

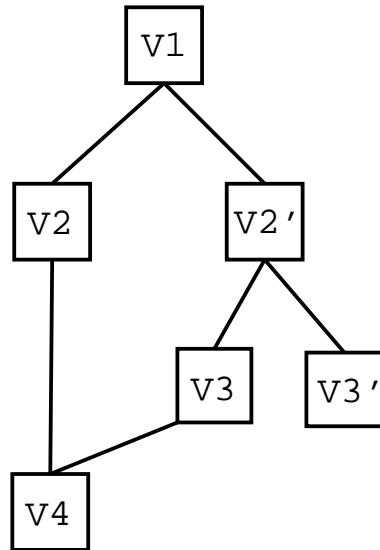
Versioiden sulauttaminen saattaa mutkistua yhteiskirjoitusvälineen tapauksessa, jos tuotettuun tekstiin liittyy sosiaalista tai kognitiivista tietoa (Lee *et al.*, 1998). Tekstikappaleeseen saattaa esimerkiksi olla sidottu tietoa tekijästä ja hänen roolistaan. Tulisiko järjestelmän huomioida roolihierarkkia versioita sulautettaessa ja voiko hierarkiassa ylempänä olevan käyttäjän tuotoksia suosia alempana olevaan nähden? Näihin kysymyksiin ei ole yksiselitteistä vastausta, mutta on selvää, että järjestelmän versioiden sulauttamismekanismiin pitää olla käyttäjille selvä ja uudesta muodostetusta versiosta on käytävä ilmi, kuinka se on muodostettu. Munson ja Dewan (1994) esittävät erään ratkaisun ryhmätyöohjelmissa sulauttamisen yhteydessä syntyviin konfliktitilanteisiin. Taulukko 2 on sulauttamismatriisi, jonka sarakkeet ja rivit edustavat kumpikin yhden käyttäjän mahdollisia muokkaustoimenpiteitä yhteiseen tuotokseen. Jos esimerkiksi käyttäjä A päättää tuhota tietyn elementin dokumentista ja käyttäjä B tahtoo muokata samaa elementtiä, syntyy konfliktitilanne.

Taulukko 2: Sulauttamismatriisi (Munson ja Dewan, 1994).

Toiminto	Lisää elementti	Poista elementti	Korvaa elementti
Lisää elementti	molemmat operaatiot	molemmat operaatiot	molemmat operaatiot
Poista elementti	molemmat operaatiot	rivin operaatio	konflikti
Korvaa elementti	molemmat operaatiot	konflikti	konflikti

Usean version sulauttaminen on haasteellinen tehtävä, sillä jos esimerkiksi sulautetaan  $n$  versiota, joihin on tehty  $m$  erilaista operaatiota, muodostuu sulauttamismatriisin tilakompleksisuudeksi  $O(m^n)$  (Munson ja Dewan, 1994). Tällaisen matriisin täyttäminen ja käsittely saattaa olla hidasta.

Versioinnin käyttökelpoisuutta lisää versioiden selaus ja versioiden vertailu (differentiate, diff.) (Lee *et al.*, 1998). Kun dokumenttia versioidaan, syntyy versiopuu, jossa solmut ovat tiettyjä versioita ja kaaret edustavat näiden välisiä riippuvuuksia. Joissakin tapauksissa versiopuusta voi muodostua myös verkko, jos versionhallintajärjestelmä sallii kahden version yhdistämisen sulauttamalla tai muulla tavalla. Versiopuu tai -verkko kuvaa dokumentin versioiden johdoshistoriaa, joka muodostuu ensimmäisestä dokumentin versiosta ja sen väliversioista (Haake ja Haake, 1993). Kuvassa 3 on esimerkki tuotoksen johdoshistoriasta.



Kuva 3: Esimerkki johdoshistoriasta.

Kuvan 3 versiohistoriaa luetaan ylhäältä alas. V1 on tuotoksen ensimmäinen versio ja versiot V2 ja V2' sekä V3 ja V3' ovat keskenään rinnakkaisia versioita, jotka saattavat sisältää ristiriitaista tietoa. Versio V4 on versioiden V2 ja V3 yhdistelmä. Järjestelmä voi perustaa versioiden selaus- tai navigointitoiminnon graafiseen esitykseen, jossa versiohistoria kuvataan kuvan 3 esittämällä tavalla. Versioiden nimet voivat olla järjestelmän automaattisesti generoimia, käyttäjien syöttämiä tai näiden yhdistelmiä.

Kahden version vertailu on tärkeä työkalu versionhallintaa hyödynnettäessä. Vertailutyökalu ilmoittaa tyypillisesti kahden version väliset erot tietyllä tarkkuudella, esimerkiksi kaikki muuttuneet kappaleet tai jopa kaikki muuttuneet sanat. Neuwirth *et al.* (1992) ovat sitä mieltä, että yhteiskirjoitusväline vaatii joustavan vertailutyökalun. He perustelevat vaatimustaan kirjoitusryhmän ja yksittäisten kirjoittajien tarpeiden vaihtelulla ja erilaisten kirjoitusryhmien olemassaololla. Kirjoitusryhmä ei välttämättä tarvitse kovin tarkkaa vertailutyökalua silloin, kun ryhmän jäsenet työstävät omia osiaan dokumentista. Mutta kun ryhmän jäsenten kirjoittamat tekstinpätkät yhdistetään suuremmaksi kokonaisuudeksi, saattavat ryh-

män jäsenet tarvita keinoja tutkia, mitä muut ryhmän jäsenet oikein ovat lisäilleet tuotokseen.

Eri roolien (rooleista lisää alakohdassa 2.2.1) edellyttämät tehtävät vaativat erilaisia vertailuja. Esimerkiksi toimittajana toimiva ryhmän jäsen tarvitsee keinon tutkia, ovatko kirjoittajat tehneet hänen ehdottamansa muutokset. Lakitieteellistä tekstiä tuottava ryhmä tarvitsee tarkkaa vertailutyökalua, koska sanamuodon muutokset saattavat muuttaa merkittävästi tuotoksen sanomaa. Toisaalta kirjaa yhdessä työstävä ryhmä on kiinnostuneempi yleisvaikutelmasta kuin yksittäisistä sanoista tai lauseista. Neuwirth *et al.* (1992) huomioivat, että versioita vertaillessa käyttäjälle tulisi aina tarjota sellainen määrä informaatiota, että se tukee hänen senhetkistä tehtäväänsä eikä kuormita hänen kognitiivisia voimavarojaan liian suurella tai pienellä informaatiomäärällä.

Neuwirth *et al.* (1992) esittelevät joustavan erottelun, menetelmän, jossa vertailua säädetään useiden parametrien avulla. Karkeusparametri ilmoittaa, millä tarkkuudella eroja esitetään. Parametrin arvona voi olla kirjaimen, sanan, lauseen tai kappaleen tarkkuus. Toinen parametri määrittelee pienimmän tarkasteltavan eron koon. Se voi olla kirjain, sana, lause tai kappale. Taulukossa 3 kuvattu tarkan erotelutarkkuuden eroraportti (c) syntyy silloin, kun käyttäjä määrittelee sekä karkeusparametrin että pienimmän tarkasteltavan eron sanaksi. Silloin jokainen lisätty sana on alleviivattu ja vastaavasti poistetut sanat kursivoituja. Taulukon 3 kohdan e karkean erotelutarkkuuden eroraportti puolestaan muodostuu, kun käyttäjä määrittelee karkeusparametrin sanan tasolle ja pienimmän tarkasteltavan eron lauseeksi. Tällöin eroraporttiin sisällytetään jokainen muuttunut sana, mutta muutos kuvataan sen lauseen muutoksena, johon sana kuuluu erottelematta muuttunut sana.



Taulukko 3: Versioiden vertailun erottelutarkkuus (Neuwirth *et al.*, 1992).

a) alkuperäinen teksti	Oli kylmää. Vikkelä ruskea kettu hyppää laiskan koiran yli. Hänen kuppinsa on tuolla, auton luona.
b) muutettu teksti	Oli aamu. Oli kylmää. Hidas ruskea koira hyppää laiskan kissan yli. Hänen kuppinsa on tuolla, mopon luona.
c) tarkka erottelutarkkuus	<u>Oli aamu.</u> Oli kylmää. <i>Vikkelä</i> <u>Hidas</u> ruskea <i>kettu</i> <u>koira</u> hyppää laiskan <i>koiran</i> <u>kissan</u> yli. Hänen kuppinsa on tuolla, <i>auton</i> <u>mopon</u> luona.
d) keskitasoinen erottelutarkkuus	<u>Oli aamu.</u> Oli kylmää. <i>Vikkelä</i> <i>ruskea</i> <i>kettu</i> <u>Hidas ruskea koira</u> hyppää laiskan <i>koiran</i> <u>kissan</u> yli. Hänen kuppinsa on tuolla, <i>auton</i> <u>mopon</u> luona.
e) karkea erottelutarkkuus	<u>Oli aamu.</u> Oli kylmää. <i>Vikkelä</i> <i>ruskea</i> <i>kettu</i> <i>hyppää</i> <i>laiskan</i> <i>koiran</i> <i>yli.</i> <u>Hidas ruskea koira hyppää laiskan kissan yli.</u> <i>Hänen kuppinsa on tuolla, auton luona.</i> <u>Hänen kuppinsa on tuolla, mopon luona.</u>

Taulukon 3 kohdan d keskitason erottelutarkkuuden mukainen muutosraportti syntyy, kun järjestelmä antaa käyttäjän määritellä sen, yhdistetäänkö lähellä toisiinsa sijaitsevat muutokset yhdeksi muutokseksi raportissa. Esimerkkimme tapauksessa käyttäjä on asettanut etäisyyden, jolla muutokset yhdistetään neljään sanaan sekä karkeustason ja pienimmän raportoitavan muutoksen sanan tasolle.

Usean käyttäjän järjestelmissä *version näkyvyys* voi vaihdella (Lee *et al.*, 1998). Dokumentin globaali versio on kaikkien käyttäjien ulottuvilla. Paikallisen version näkyvyys on rajallisempi, sen näkee vain tietty käyttäjä. Erityisesti sellaiset jär-

jestelmät, jotka tukevat itsenäistä työskentelyä tai asynkronista työskentelyä hyötyvät paikallisen versioinnin mahdollisuudesta. Kun käyttäjä työskentelee tilapäisesti muusta ryhmästä erillään, hän saattaa haluta versioida työstämäänsä tuotosta. Myöhemmin hän saattaa sulauttaa tekemänsä tuotoksen version ryhmän päätuotokseen tai kenties ainoastaan tehdä paikallisesta versiosta koko ryhmälle näkyvän globaalin version keskustelua varten. Järjestelmän tulee myös tukea paikallisen ja globaalin version vertailua (Haake ja Haake, 1993).

## 2.2 Koordinointi

Ryhmätyöskentelyä on koordinoitava tavalla tai toisella, jotta ryhmän jäsenet toimisivat yhteisymmärryksessä yhteisen tavoitteen saavuttaakseen. Ellisin *et al.* (1991) mukaan Singh (1989) määrittelee koordinaation keskeiseksi tehtäväksi ryhmän jäsenten henkilökohtaisten työpanosten kohdistamisen harmonisesti suuremman tavoitteen aikaansaamiseksi. Yhteiskirjoitusväline voi auttaa ryhmää tukemalla erilaisia koordinoitirakenteita ja -menettelytapoja, joiden varaan ryhmätyö voidaan perustaa (Baecker *et al.*, 1993; Neuwirth *et al.*, 2001). Hoften (1996) mielestä keskeisiä kysymyksiä koordinoinnissa ovat: kuka määrittelee rakenteet ja menettelytavat, joita ryhmä käyttää, ja pakottaako vai rohkaiseeko yhteiskirjoitusväline noudattamaan tiettyjä rakenteita ja menettelytapoja? Nämä filosofisilta kalskahtavat kysymykset sopivat hyvin johdattelemaan lukijaa yhteiskirjoittamisen koordinointiin, koska asia ei ole yksinkertainen eikä varsinkaan yksiselitteinen. Ongelmana on ihmisen ja erityisesti ihmisryhmän heuristinen luonne. Tehokkaassa ryhmätyöskentelyssä ryhmän jäsenet sopeutuvat muuttuviin tilanteisiin muuttamalla menettelytapojaan. Toisaalta ryhmätyöskentely on mahdotonta, elleivät ryhmän jäsenet ota vastuuta eri tehtävistä tai jos ryhmällä ei ole mielikuvaa siitä, mitä seuraavaksi pitäisi tehdä. Myös käytännön järjestelyt, kuten työtila ja kokoukset, ovat tärkeitä koordinoinnin kannalta.

Rao *et al.* (1996) jakavat tietokoneavusteiset yhteistoiminnallisuutta tukevat välineet korkean rakenteen (high-structure) ja matalan rakenteen (low-structure) jär-

jestelmiin. Korkean rakenteen välineet on suunniteltu ja toteutettu melko jäykästi; ne pyrkivät tarkoituksella tai vahingossa muuttamaan niitä käyttävän ryhmän rakennetta ja työskentelytapoja. Matalan rakenteen ryhmätyöskentelyvälineet muokautuvat paremmin käyttäjiensä tarpeisiin ja ovat tässä suhteessa korkean rakenteen välineiden vastakohtia. Hofte (1998) käyttää korkean rakenteen järjestelmästä nimeä rajoittava järjestelmä ja vastaavasti matalan rakenteen järjestelmästä termiä salliva järjestelmä. Järjestelmä voi asennoitua ryhmätyöhön joko rajoittavasti tai sallivasti. Rajoittava järjestelmä asettaa sääntöjä ja rajoja, joiden on tarkoitus ohjata työtä. Salliva järjestelmä puolestaan mahdollistaa ryhmätyöskentelyä tarjoamalla erilaisia palveluja ja tukitoimia.

Tässä luvussa käsittelen keinoja, joilla yhteiskirjoitusvälineet pyrkivät koordinoimaan tai antamaan käyttäjilleen keinoja koordinoida ryhmätyöskentelyä. Alakohdassa 2.2.1 tutustumme roolien ja sosiaalisten protokollien tukemiseen yhteiskirjoitusvälineissä, mikä on yksi keino antaa käyttäjille keinoja hallita työskentelyään ja määrittellä ryhmätyön kannalta tärkeitä sosiaalisia rakenteita. Roolien analysoiminen on tärkeää myös siksi, että ne kertovat mitä käyttäjät järjestelmällä tekevät. Tämä tieto on yhteiskirjoitusvälineen suunnittelijoille tärkeää.

Toinen tapa antaa ryhmätyölle selkärankaa on työskentelyprosessin määrittäminen ja noudattaminen. Yhteiskirjoitusprosessia on tutkittu paljon ja on luonnollista, että yhteiskirjoitusväline tukee jossain määrin hyvin tunnettua prosessia. Tarkastelemme yhteiskirjoittamista prosessina alakohdassa 2.2.2. Lopuksi tutustumme alakohdassa 2.2.3 käytännönläheisiin koordinaatiokysymyksiin: ryhmän muodostamiseen, koolle kutsumiseen ja hallinnoimiseen, joita yhteiskirjoitusvälineen tapauksessa luonnehditaan kirjoitussession hallinnalla.

### **2.2.1 Roolit ja sosiaaliset protokollat**

Sosiaalisessa vuorovaikutuksessa omaksumme tyypillisesti erilaisia rooleja. Roolit liittyvät läheisesti sosiaalisiin protokolliin, jotka hallitsevat ihmisten välistä vuorovaikutusta. Smith *et al.* (1998) määrittelevät roolin inhimilliseksi konstruk-

tioksi, joka syntyy ja jota pitää yllä ihmismielien välinen vuorovaikutus. Roolit kertovat meille vastuumme ja velvollisuutemme sekä paikkamme yhteisössä. Lapsen ja vanhemman suhde on eräs perustavanlaatuinen esimerkki rooleista ja niihin liittyvistä protokollista. Vanhemmilla on kasvattajan ja huoltajan rooli, joka sanelee kuinka heidän tulee suhtautua lapseen. Lapsien tulee yleisesti vallitsevien sosiaalisten protokollien mukaan suhtautua kunnioittavasti ja tottelevasti vanhempiinsa. Lapset tosin usein tuovat esiin erään roolien luonteenomaisen piirteen: roolit eivät välttämättä ole tarkasti määriteltyjä ja niihin liittyy paljon tulkinnanvaraisia ja kulttuurisidonnaisia tekijöitä. Näissä tilanteissa ihmiset kokeilevat ja etsivät rooliensa rajoja, jolloin roolit saattavat jopa muuttua. Esimerkiksi vanhempansa menettäneiden sisarusten tapauksessa on esimerkkejä vanhimman lapsen roolin muuttumisesta isoveljestä tai -siskosta vanhemmaksi.

Sosiaaliset roolit ja protokollat ovat tärkeässä roolissa myös ryhmätyöskentelyssä, eikä niitä voi sivuuttaa ryhmätyöohjelmissa. Grudin (1988) toteaa tietokoneavusteista ryhmätyötä kriittisesti tarkastelevassa artikkelissaan, että jos ryhmätyöohjelma ei huomioi sosiaalisia protokollia, käyttäjät hylkäävät sen. Grudin määrittelee ryhmätyöohjelman valmisohjelmaksi, joka on suunniteltu tukemaan työtä sen sosiaalisessa kontekstissa. Tämä niin kutsuttu sosiaalinen komponentti (Kuutti, 1998) on yllättävän hankala asia ryhmätyöohjelmistojen suunnittelijoille, joille ei välttämättä ole intuitiivisesti selvää minkälaisia vuorovaikutussuhteita ja rooleja välineen käyttöympäristössä on (Grudin, 1988). Yhteiskirjoitusvälineissä on huomioitu Grudinin opit. Yhteiskirjoittamisen olemusta on tutkittu laajalti epistemologisin keinoin (Posner ja Baecker, 1992; Baecker *et al.*, 1993), sekä prototyyppijärjestelmien perustavanlaatuisen evaluointien (Neuwirth *et al.*, 1994a; Mitchell *et al.*, 1995) kautta.

Baecker *et al.* (1993) seurasivat yhteiskirjoitusryhmiä ja erittelivät neljä roolia, joita esiintyy yhteiskirjoittamisessa. Taulukkoon 4 on Baecker *et al.*:n roolien lisäksi poimittu mukaan rooli edistäjä lähteestä (Mitchell, 1996). Yhteiskirjoitusryhmässä esiintyvät roolit ovat tiukasti sidoksissa yhteiskirjoittamisprosessiin ja siinä esiintyviin tehtäviin, joista lisää alakohdassa 2.2.2. Tietyn roolin omaava

henkilö voi esimerkiksi vastata tietyn tehtävän suorittamisesta.

Taulukko 4: Roolit yhteiskirjoitusprosessissa (Baecker *et al.*, 1993; Mitchell, 1996).

<b>Rooli</b>	<b>Tehtävä</b>
kirjoittaja	Tuottaa aktiivisesti tekstiä.
toimittaja	Muokkaa ja korjaa muiden (kirjoittajien) kirjoittamaa materiaalia.
tarkastaja	Arvioi ja kommentoi tuotettua tekstiä.
konsultti	Osallistuu aktiivisesti yhteiskirjoitushankkeeseen, mutta ei kirjoita tekstiä.
edistäjä	Jakaa roolit ja organisoii kirjoitusprosessia, mutta ei kirjoita tekstiä.

Guzdial *et al.* (2001) ottavat kokonaisvaltaisemman kannan rooleihin ryhmätyöohjelmissa. He erittelevät kahdeksan roolia, joita esiintyy ryhmätyöohjelmia käytettäessä. Taulukossa 5 on yhteenveto Guzdial *et al.* (2001) löytämistä rooleista selityksineen. Rooleista kirjoittajat, käyttötoimitsijat ja keskeiset käyttäjät muodostavat ydinjoukon, josta yhteistyötä tekevä ryhmä pääasiassa koostuu. Muut roolit toimivat ryhmätyön tukijoina ja rikastuttajina.

Taulukko 5: Rooleja ryhmätyöohjelmissa, mukaelma taulukosta lähteessä Guzdial *et al.* (2001).

<b>Rooli</b>	<b>Tehtävä</b>
kirjoittaja	Lisää materiaalia, etsii uutta materiaalia, yhdistää materiaalia ja huolehtii materiaalin saatavuudesta.
käyttötoimitsija	Rohkaisee ympäristön oikeaan käyttöön ja määrittelee käyttökontekstin.
keskeinen käyttäjä	Huolehtii yhteiseen työtilaan kootun materiaalin rakenteesta.
toisarvoinen käyttäjä	Kommentoi ja esittävää huomautuksia materiaaliin liittyen.
järjestelmän suunnittelija	Muuttaa loppukäyttäjän käyttöliittymän käyttötun- tumaa.
järjestelmän kehittäjä	Hyödyntää järjestelmän tarjoamaa teknologiaa li- säten jopa uusia piirteitä.
ylläpitäjä	Ylläpitää järjestelmää tuntematta tarkemmin tekni- siä yksityiskohtia.
tukihenkilö	Ylläpitää ja huoltaa laitteistoa.

On mielenkiintoista huomata, että siinä missä Baecker *et al.* löysivät ja tunnistivat tietokoneista riippumattomia rooleja, tutkivat Guzdial *et al.* nimenomaan tietokoneen ja sen ryhmätyöskentelykäytön synnyttämiä ja mahdollistamia rooleja. Taulukoita 4 ja 5 vertaillen on huomioitava eri lähtökohdat, Baecker *et al.* (1993) tarkastelevat yhteiskirjoittamista, kun taas Guzdial *et al.* (2001) tutkivat tietokoneavusteista yhteistyötä käyttäen tutkimuskohteenaan ryhmätyötä tukevaa avointa oppimisympäristöä. Mielestäni roolimääritelmät tukevat toisiaan eri lähtökohdista huolimatta tai paremminkin eri lähtökohtiensa vuoksi. Baecker *et al.* (1993) määrittelevät tarkasti tietokoneavusteisen ryhmätyön yhteiskirjoituskontekstin roolit ja Guzdial *et al.* (2001) puolestaan kartoittavat tietokoneavusteisessa ryhmätyössä esiintyvät roolit. Yhdessä roolimääritelmiä voidaan käyttää muodostettaessa

mielikuvaa nimenomaan tietokoneavusteisessa yhteiskirjoittamisessa esiintyvistä rooleista.

Roolit ja sosiaaliset protokollat ovat tärkeitä, mutta kuinka niitä voidaan tukea tietokoneavusteisessa yhteiskirjoittamisessa? Ongelmaksi muodostuu ihmisten välisen vuorovaikutuksen epädeterministinen luonne. Ihmisille monimutkaiset, osittain tiedostamattomat kulttuurisidonnaiset vuorovaikutussuhteet ovat pitkälti itseäänselvyyksiä samoin kuin ryhmän dynaamisuus. Neuwirth *et al.* (2001) toteavat, että jos roolit ja sosiaaliset protokollat ovat joustavia, niin silloin yhteiskirjoitusvälineiden tiedonsiirtoprotokollienkin on oltava sellaisia. Sekä Neuwirth *et al.* (1990) että Baecker *et al.* (1993) huomioivat liian kankean roolien määrittelyn haittaavan ryhmätyöskentelyä. Pahimmillaan roolit kääntyvät tarkoitustaan vastaan tukahduttaen yksilön luovuuden ja sotkien ryhmän sisäisen luonnollisen vuorovaikutuksen. Toisaalta välineen on tuettava rooleja ja sosiaalisia protokollia jollain tavalla ollakseen käyttökelpoinen, kuten Grudin (1988) huomauttaa.

Kirjallisuudesta löytyy useita esimerkkejä erilaisista lähestymistavoista rooliky-symykseen. Neuwirth *et al.* (1990) ovat sitä mieltä, että rooleja ei tule määrittellä tarkasti yhteiskirjoitusvälineessä, vaan ne tulee jättää käyttäjäkunnan vastuulle. Neuwirth *et al.* (1994a) vahvistivat käytännön kokeiden kautta, ettei luonnollisia sosiaalisia vuorovaikutussuhteita saa sotkea pakotetuilla malleilla. Neuwirth *et al.* (1994b) hylkäsivät eksplisiittiset roolit ja korvasivat ne vuorovaikutusparametreillä. Vuorovaikutusparametrejä säätämällä käyttäjä voi säätää vuorovaikutustaan muiden käyttäjien kanssa päättäen mitä, missä, milloin ja kuka hänestä näkee ja kuulee. Yhdistettynä muokkaus- ja lukuoikeuksien määrittelyyn vuorovaikutusparametreillä voidaan simuloida erilaisia vuorovaikutussuhteita ja tätä kautta myös rooleja. Fish *et al.* (1988) ottavat päinvastaisen kannan roolien määrittelylle. Heidän Quilt-järjestelmässään on määritelty valmiiksi joitakin rooleja, jotka samalla sanelevat muokkaus- ja lukuoikeudet haltijoilleen. Lisäksi Quilt tukee käyttäjien määrittelemiä rooleja, joiden avulla on mahdollista räätälöidä tietyille ryhmälle sopivia rooli- ja käyttöoikeusyhdistelmiä (katso alakohta 4.3.2).

## 2.2.2 Kirjoitusprosessi

Ryhmätyöprosessi koostuu vaiheista, joiden aikana tuotosta kehitetään. Prosessin työvaiheet mielletään usein tietokoneavusteisessa ryhmätyössä tehtäviksi, jotka on suoritettava tietyssä järjestyksessä, jotta tuotos valmistuisi. Yhteiskirjoittaminen ei tässä suhteessa ole poikkeus. Neuwirth *et al.* (2001) ovat tutkineet yhteiskirjoittamista ja päätyneet jakamaan yhteiskirjoitusprosessin kolmeen päävaiheeseen: suunnitteluun, luonnosteluun ja tarkastamiseen. Päävaiheet jakautuvat edelleen alitehtäviin. Baecker *et al.* (1993) ovat myös tutkineet yhteiskirjoittamista keskittyen erityisesti aktiviteetteihin tai alitehtäviin, joita yhteiskirjoittamisen yhteydessä esiintyy. Luonnostelen seuraavaksi yhteiskirjoitusprosessia käyttäen molempia edellä mainitsemiani lähteitä. On mielenkiintoista havaita, että sekä Neuwirth *et al.* (2001) että Baecker *et al.* (1993) ovat päätyneet pääpiirteittäin samoihin tuloksiin toisistaan hieman poikkeavista näkökulmista huolimatta.

Suunnitteluvaihe aloittaa kirjoitusprosessin. Sen aikana tehdään suunnitelmia sekä tuotettavan dokumentin sisällöstä että varsinaisesta työskentelystä (Neuwirth *et al.*, 2001). Aluksi ryhmän on päästävä yhteisymmärrykseen tekstin tarkoituksesta ja siitä, kenelle se on suunnattu. Seuraavaksi ryhmä ideoi sisältöä, Baecker *et al.* (1993) puhuvat tässä yhteydessä aivoriieheilystä. Kun ideoita alkaa olla kassassa, joutuu ryhmä suunnittelemaan, kuinka dokumentin sisältö järjestetään. Samalla ryhmän on tehtävä suunnitelmia prosessin jatkamisesta, varsinaisesta työskentelystä, joka johtaa tehtyjen suunnitelmien perustalta valmiiseen tuotokseen. Suunnitteluvaiheessa ryhmä määrittelee tavoitteensa sekä sen, kuinka se ne aikoo saavuttaa. Baecker *et al.* (1993) sisällyttävät suunnitteluaktiviteetteihin myös tutkimisen, jolla he tarkoittavat ryhmän ulkopuolisten lähteiden tutkimista tuotoksen kannalta tärkeän tiedon kartuttamiseksi.

Suunnitteluvaiheen aikana ryhmän työskentely on hyvin intensiivistä, koska ryhmä keskustelee ja työstää yhdessä ajatuksiaan (Neuwirth *et al.*, 2001). Suunnitteluvaiheen aikana korostuu yhteiskirjoitusvälineen keinot mahdollistaa ryhmän jäsenten välinen keskustelu reaaliaikaisesti, synkronisesti (Baecker *et al.*, 1993).



Kasvotusten kohtaava ryhmä käyttää puheen lisäksi kommunikaatiokeinoinaan eleitä, ilmeitä ja erilaisia improvisoituja demonstraatiovälineitä, esimerkiksi näinäinaa sisältöluonnoksen pohjana. Neuwirth *et al.* (2001) ovat sitä mieltä, että yhteiskirjoitusvälineelle on erityisen tärkeää tukea puheen lisäksi myös muita kommunikaatiokeinoja. Käsittelen tarkemmin yhteiskirjoitusvälineen keinoja tukea ryhmän sisäistä kommunikaatiota kohdassa 2.3.

Luonnosteluvaiheessa ryhmä tuottaa tekstiä ideoittensa perusteella. Neuwirth *et al.* (2001) huomioivat, että suunnitelmiin kirjatut ajatukset selkenevät usein vasta varsinaisen tekstin tuottamisen yhteydessä. Tästä syystä toisistaan erillään kirjoittavat ryhmän jäsenet saattavat käsittää aiemmat yhdessä suunnitellut suuntaviivat hyvinkin eri tavoin. Ryhmän käsityksien on tietenkin täsmättävä ja suunnitelmia tarkastettava, jotta tuotos syntyy. Luonnosteluvaiheelle tunnusomaista on kehittyvän dokumentin rooli osana työympäristöä (Neuwirth *et al.*, 2001). Dokumentti ei ole pelkkä lopputulos vaan jaettu resurssi, jota koko ryhmä käyttää koko prosessin ajan. Dokumentin saatavuus on tärkeää myös siksi, että luonnosteluvaiheessa ryhmän jäsenet tyypillisesti työstävät omia osuuksiaan erillään, jolloin järjestelmän tuki asynkroniselle työskentelylle korostuu (Baecker *et al.*, 1993).

Baecker *et al.* (1993) ja Neuwirth *et al.* (2001) huomioivat, että yhteiskirjoitusryhmän jäsenten työskentelytottumuksissa voi olla suuria eroja. Toiset henkilöt haluavat työstää omaa osuuttaan dokumentista julkisesti, kaikkien ryhmän jäsenten näkyvissä saaden kenties palautetta keskeneräiselle tekeleelleen (Baecker *et al.*, 1993). Joku muu ryhmän jäsenistä saattaa haluta työstää kappalettaan kaikessa rauhassa ja näyttää muulle ryhmälle vasta viimeistellyn ja tarkastetun lopputuloksen (Neuwirth *et al.*, 2001). Joskus työskentelytyyliin voi vaikuttaa myös olosuhteet, kuten aikaero, jolloin reaaliaikainen työskentely ei ole mahdollista lainkaan. Yhteiskirjoitusvälineen on tuettava useita lähestymistapoja, jotka korostuvat vapaamuotoisen luonnosteluvaiheen aikana. Baecker *et al.* (1993) ovat sitä mieltä että yhteiskirjoitusvälineen on juuri tästä syystä tuettava sekä asynkronista, että synkronista työskentelyä. Lisäksi dokumentin hallinnan on oltava joustava, jotta yksityisyyttä kaipaavat ryhmän jäsenet voivat määrittellä missä määrin heidän

työskentelystään näkyy muille.

Tarkistamisvaiheessa kirjoitusryhmä arvioi ja tarkastaa tuottamaansa tekstiä. Tekstiä arvioitaessa ryhmän jäsenet kommentoivat tekstiä eri tavoin (Baecker *et al.*, 1993; Neuwirth *et al.*, 2001). Neuwirth *et al.* (2001) pitävät tärkeänä sitä, että ryhmä pystyy keskustelemaan kommentista ja myös kommentoimaan kommentteja. He ovat päätyneet tähän tulokseen huomioituaan, että kirjoitusryhmän kokoontuessa arviointipalaveriin keskustelu käydään usein tekstiin kirjattuja kommentteja selvittäen. Kommentit muodostavat arvioinnin rungon eivätkä välttämättä itsessään muodosta tyhjentävää arviointia vaan toimivat ikään kuin huomautuksina ja muistilappuina arvioijille. Käsittelen kommentteja tarkemmin alakohdassa 2.3.2. Tekstin tarkastamisella tarkoitetaan ryhmän hyväksymien korjausten ja muutosten tekemistä tekstiin. Koska muutettu dokumentti on jälleen arvioitava, syntyy arvioinnin ja tarkastamisen kesken sykli, joka jatkuu ja toistuu, kunnes dokumentti arvioidaan valmiiksi. Taulussa 6 on yhteenveto yhteiskirjoitusprosessista tehtävineen.

Taulukko 6: Yhteenveto yhteiskirjoitusprosessista (Neuwirth *et al.*, 2001; Baecker *et al.*, 1993).

Vaihe	Tehtävä
suunnittelu	dokumentin suunnittelu
	työskentelyn suunnittelu
	sisältöideointi
	sisällön organisointi
luonnostelu	dokumentin kirjoittaminen
tarkistus	arviointi
	tarkastaminen

Kirjoitusprosessia luonnehtii vaihteleva työskentelyn intensiteetti. Osallistujat voivat myös olla eri vaiheissa, jolloin ryhmän sisällä on erilaisia työskentelymuotoja tarvitsevia henkilöitä (Baecker *et al.*, 1993). Siksi on tärkeää, että yhteiskirjoitusväline mahdollistaa erilaisia työskentelytapoja, esimerkiksi alakohdassa

2.1.2 mainitulla käyttöliittymän rentouttamisen avulla. Myös sekä asynkronisen että synkronisen työn tukeminen on eräs yhteiskirjoitusvälineen perusominaisuuksista, jotta eri kirjoitusvaiheiden erilaiset työskentelymuodot olisivat mahdollisia.

Kirjoitusprosessin huomioiminen yhteiskirjoitusvälineessä on projektinhallintaa. Kirjoitushankkeen jakaminen alitavoitteisiin, joilla on tietty kesto ja sijainti kokonaisuudessa selkeyttää työskentelyä. Näin ollen yhteiskirjoitusväline voi tukea ryhmätyöskentelyä myös tarjoamalla projektinhallintatoimintoja, joita voivat olla kirjoitusprosessin seuranta ja aikarajojen asettaminen. Esimerkiksi yhteiskirjoitusvälineet PREP (katso kohta 4.1.2) ja Quilt antavat käyttäjiensä asettaa dokumenttiin kahdenlaisia aikarajoja: pehmeitä ja kovia (Fish *et al.*, 1988; Leland *et al.*, 1988; Neuwirth *et al.*, 1994b). Pehmeät aikarajat toimivat vain huomautuksina kirjoittajille, kovat aikarajat sulkevat muokattavan objektin määräaikana, jonka jälkeen sitä ei voi enää muokata. Koch (1995) on sitä mieltä, että yhteiskirjoitusvälineen projektinhallinta pitäisi hoitaa ulkoisen tarkoitusta varten suunnitellun sovelluksen avulla.

Tilanne on hieman paradoksaalinen: toisaalta järjestelmältä odotetaan selkeää mallia yhteiskirjoitusprosessille, mutta toimiakseen mallin tulee olla hyvin joustava. Yhteiskirjoitusvälineen suunnittelijoiden ja käyttäjien on tyydyttävä jonkinlaiseen kompromissiin prosessin suhteen. Baecker *et al.* (1993) toteavat yhteiskirjoittamisen edellyttävän henkilökohtaisten mieltymysten uhraamista ryhmän harmonian puolesta. Galegher ja Kraut (1992) huomioivat kokeelliseen aineistoon tukeutuen, että ryhmä voi muuntaa työskentelyprosessiaan käytettävälle välineelle sopivammaksi, mutta se vaatii paljon ylimääräistä työtä. Grudin (1988) muistuttaa, ettei ryhmätyöväline saa edellyttää keneltäkään ylimääräistä työtä sen tuottamaan hyötyyn nähden tai se jää käyttämättä.

### **2.2.3 Kirjoitussessio**

Tietokoneavusteisessa ryhmätyössä työympäristö rakennetaan tyypillisesti jonkin reaali maailmasta löydetyn metaforan varaan. Jaettu työtila kuvataan spatiaalisena

tilana, kuten virtuaalisena toimistorakennuksena, jossa yhteistyötä tekevät henkilöt liikkuvat. Virtuaalisessa tilassa järjestetään konferensseja, joiden Hofte (1996) määrittelee koostuvan tietystä määrästä käyttäjiä, jaettuja dokumentteja ja keskustelutiloja, joita hallitaan koordinaatiomekanismeilla. Konferenssit luodaan tiettyä tarkoitusta varten ja niillä on tietty elinikä ja tila. Useimmat yhteiskirjoitusvälineet käyttävät sessiokäsitettä konferenssin sijaan (Hofte, 1996). *Sessio* on käsitteellisesti rajoitetumpi kuin konferenssi ja kuulostaa epävirallisemmalta konferenssiin verrattuna. Sessio tai istunto on myös tuttu erilaisista työasemista, joihin kirjaututaan sisään työskentelyn alussa ja ulos työskentelyistunnon päättyessä.

Sessionhallinnan tehtävät voidaan jakaa ylläpitoon ja käyttöön. Ylläpitoon liittyviä sessionhallinnan tehtäviä ovat: uuden session luominen ja nimeäminen, session lopettaminen, käyttäjien oikeuksien määrittely sessiossa sekä session keston ja pysyvyyden määrittely (Roseman ja Greenberg, 1994; Greenberg ja Roseman, 1999). Session pysyvyydellä tarkoitetaan sitä, häviääkö sessio kaikkien käyttäjien poistuttua siitä vai jääkö se odottamaan käyttäjien paluuta. Käytön kannalta sessionhallinnan on huolehdittava siitä, että käyttäjä voi löytää session ja liittyä siihen sekä lähteä siitä halutessaan pois. Sessionhallinnan tehtävä, joka liittyy sekä ylläpitoon että tavalliseen käyttöön, on tiedon välittäminen session osallistujista.

Tutustumme seuraavaksi Greenbergin ja Rosemanin (1998) ajatuksiin huonemetaforasta session peruskonseptina. Huone on rajoitettu tila, se voi sisältää esineitä ja ihmisiä ja sillä on tietty sijainti. Lisäksi ihmiset työskentelevät ja vaikuttavat huoneissa. Tietokoneavusteinen ryhmätyö on mahdollista järjestää huonemetaforan varaan, jolloin järjestelmän käyttäjät voivat tukeutua heille jokapäiväisestä työskentelystä tuttuihin asioihin ja niiden abstraktioihin. Huone on rajoitettu tila, jossa voi olla esineitä ja ihmisiä ja jota voi tutkia kurkistamalla vaikkapa ikkunasta tai pistäytymällä siinä. Huoneeseen jätetyt esineet säilyvät huoneessa, yleensä huoneen omistaja järjestee tavaransa tietyllä häntä miellyttävällä tavalla. Joihinkin huoneisiin saattaa olla rajoitettu pääsy lukkomekanismin tai kulunvalvonnan kautta. Huonemetaforaa käyttävässä ryhmätyöohjelmassa käyttäjät voivat olla tietyssä huoneessa, jolloin he näkevät huoneen sisällön. Pysyvyyttä ryhmätyöohjelmassa

edustaa huoneisiin jätetyt objektit, jotka myös mahdollistavat asynkronisen työskentelyn esimerkiksi vuorotyönä. Käyttäjät myös tietävät, missä huoneissa muut työskentelevät, ja järjestelmä voi säätää käyttäjien välistä kytkentää (katso alakohdasta 2.1.2) heidän sijaintinsa mukaan: samassa huoneessa olevien välinen kytkentä on tiukempi kuin eri huoneissa olevien.

Huonemetaforaa yksinkertaisempi metafora on yhteiskirjoittamisen tapauksessa dokumentti. Huonetta yksinkertaisempana käsitteenä se ei kata läheskään kaikkia niitä asioita, joita ryhmätyössä tarvitaan. Yhteiskirjoitusvälineessä dokumentti on kokoelma objekteja, esimerkiksi vedoksia ja kommentteja, itse päätuotoksen lisäksi. Hyperavaruus on dokumenttia laajempi käsite, joka on objekteja sisältävä virtuaalinen avaruus. Hyperavaruudessa voidaan myös puhua tilasta ja etäisyydestä, jolloin huonemetaforan yhteydessä mainittuja piirteitä on mahdollista hyödyntää. Hyperavaruus on kuitenkin useimmille ihmisille huoneita vaikeampi käsite, se saattaa tuntua käyttäjistä hieman sekavalta ja hämmentävältä.

Sessioon liittyy läheisesti käyttäjien hallinta. Alakohdassa 2.2.1 käsittelemäni roolit ja erilaiset sosiaaliset protokollat on jotenkin mahdollistettava yhteiskirjoitusvälineessä ja oikeuksien antaminen ja rajoittaminen on eräs keino tehdä tämä. Käyttäjä, joka alustaa session, määrittelee, ketkä pääsevät mukaan ja mitkä heidän oikeutensa on. Järjestelmä voi sisältää valmiiksi määriteltäviä rooleja, jotka mielletään sessionhallinnan kannalta erilaisiksi kokoelmiksi oikeuksia. Esimerkiksi yhteiskirjoitusväline Quilt sisältää joitakin valmiiksi määriteltäviä rooleja (katso alakohta 4.3.2), sekä mahdollisuuden kehittää uusia rooleja oikeuksia määrittellen (Leland *et al.*, 1988; Fish *et al.*, 1988).

## **2.3 Kommunikaatio**

Mitä paremmin ryhmän sisäinen kommunikaatio toimii, sitä paremmin ryhmätyöskentely sujuu ja jäsenten välinen luottamus kasvaa todetaan sosiologisissa tutkimuksissa (Jensen *et al.*, 2000). Fussell *et al.* (1998) määrittelevät kommunikaatio-

tion ryhmätyön kontekstissa: kommunikaatiolla tarkoitetaan kokouksia kasvotusten, sähköpostia, tiedostojen vaihtoa ja muita sellaisia keinoja, joita ryhmä käyttää suoriutuakseen tehtävistä, kuten tavoitteiden sopimisesta, päätösten tekemisestä ja tilannetietojen vaihtamisesta ryhmän jäsenten kesken. Baecker *et al.* (1993) tarkentavat määritelmää nimeämällä kolme kommunikaatiokategoriaa. Substanssista kommunikoimisella he tarkoittavat ryhmän viestintää tuotokseen liittyen. Kun ryhmä vaihtaa keskenään kysymyksiä, huomautuksia ja dokumentin versioita, on kyseessä huomautuksista ja selityksistä kommunikoiminen. Menettelytavoista kommunikoimisella ryhmä pyrkii yhteisymmärrykseen suunnitelmistaan työskentelyprosessin ja työtehtäviin nähden päästäkseen tavoitteeseensa.

Kommunikaatio on hyvin tärkeää tietoisuuden (katso kohta 2.4) kannalta. Ihmisille on luonnollista kysellä heitä askarruttavista seikoista. Myös ryhmätyöskentelyssä suuri osa tietoisuusinformaatiosta välitetään kyselemällä ja keskustelemalla asioista. Kommunikaatio on tärkeässä asemassa ryhmän luodessa yhteistä pohjaa, yhtenäistä ymmärrystä ryhmään ja ryhmätyöhön liittyvistä asioista (Mitchell, 1996). Yhteisen pohjan kehittyminen ja ylläpitäminen on jatkuva prosessi, jossa ryhmän jäsenet jatkuvasti tutkivat ja korjaavat jaettua sosiaalista todellisuuttaan. Kommunikaatiota ohjaa vaihtelevassa määrin sosiaaliset roolit ja protokollat, joita käsittelen alakohdassa 2.2.1.

Fussell *et al.* (1998) näkevät kommunikaatiokeinojen ja ryhmän rakenteen sekä työn jaon (katso kohta 2.2) toimivan osittain toistensa korvikkeina. Ryhmän on käytettävä enemmän aikaa kommunikaatioon, jos ryhmän rakenne ja työskentelyprosessi on epämääräinen. Toisaalta ryhmä voi korvata puutteelliset kommunikaatiokeinot tarkemmin määritellyllä työskentelymallilla. Tarkka työnjakokaan ei kuitenkaan voi korvata puuttuvia kommunikaatiokeinoja, jos ryhmän työskentelyyn liittyy epävarmuustekijöitä

Yhteiskirjoitusryhmä tarvitsee erilaisia kommunikaatiokeinoja yhteiskirjoitusprosessin eri vaiheissa (katso kohta 2.2.2). Yleistäen voidaan todeta, että kommunikaatiotarve kasvaa työn intensiteetin kasvaessa. Kommunikaationäkökulmasta katsottuna yhteiskirjoitusvälineen kommunikaatiotuki voidaan jakaa synkronisen

ja asynkronisen viestinnän tukemiseen. Synkroninen eli reaaliaikainen viestintä on välitöntä, hyvin vuorovaikutteista toimintaa, josta kasvotusten käytävä keskustelu on hyvä esimerkki. Asynkronista viestintää luonnehtii höllempi suhde aikaan. Kirjeenvaihto on esimerkki asynkronisesta viestinnästä, johon tyypillisesti liittyy tarkemmin määrittelemättömän pituisia taukoja.

Kommunikaatiota voidaan luokitella myös modaliteetin eli viestintämuodon mukaan (Jensen *et al.*, 2000; Neuwirth *et al.*, 1994a). Erilaisia viestintämuotoja ovat esimerkiksi puhe, teksti, piirroksiset ja sanaton viestintä. Jensen *et al.* (2000) toteavat, että puhe on välittömämpi viestintämuoto kuin esimerkiksi erilaiset kirjoitettuun tekstiin perustuvat viestintämuodot, koska se vaatii suurempaa tarkkaavaisuutta osapuolilta ja edistää näin sosiaalista läsnäoloa. Ryhmätyöskentelyssä välittömämpi viestintämuoto edistää ryhmätyötä paremmin kuin vähemmän välitön viestintämuoto. Yhteiskirjoittamisen yhteydessä on kuitenkin muistettava, että kirjoitusprosessin eri vaiheissa kommunikaatiotarpeet ovat erilaisia (katso kohta 2.2.2). Jensen *et al.* (2000) vetävät johtopäätöksensä reaaliaikaisen ryhmätyöskentelyn parissa tehdyn tutkimuksen perusteella. Itsekseen omaa tekstiosuuttaan viime tingassa väkertävä kirjoittaja ei välttämättä ole kiinnostunut sosiaalisesta läsnäolosta. Jensen *et al.* (2000) huomioivatkin, että oikean kommunikointimodaliteetin valitseminen on ratkaisevaa, kun tavoitteena on rohkaista ja tukea jotain tiettyä toimintaa.

Taulukossa 7 on kuvattu yhteiskirjoitusvälineissä käytettäviä synkronisia ja asynkronisia viestintämuotoja.

Taulukko 7: Yhteiskirjoitusvälineissä käytettäviä viestintämuotoja.

<b>Viestintämuoto</b>	<b>Synkroninen</b>	<b>Asynkroninen</b>
puhe	puhekanava, puhelin	äänikommentit
teksti	tekstipohjainen keskusteluohjelma	huomautukset ja kommentit
sanaton viestintä	hiirellä osoittaminen ja maalaaminen	
kuva	luonnostelu	työkaavio

Vaikuttaa intuitiivisesti selvältä, että kasvotusten tapahtuva keskustelu, tai ainakin reaaliaikainen keskusteluyhteys, on paras kommunikointikeino ryhmätyökontekstissa. Fussell *et al.* (1998) raportoivat kuitenkin kokeellisista tuloksistaan, joissa sähköposti osoittautui ryhmän koordinaation kannalta parhaaksi keinoksi viestiä. Tämä selittyy sillä, että ryhmät käyttivät kokeessa huomattavan osan sähköpostiviestinnästä reaaliaikaisten palaverien järjestelykysymysten ratkaisemiseen. Palavereiden vastaava hyöty ei vastannut niihin käytettyä energiaa, koska ne olivat liian lyhyitä ja niitä oli harvoin. Päätelen edellä mainituista tuloksista sen, että kirjoitusryhmän saattaa kannattaa käyttää heikompa kommunikointikeinoa, jos paremman kommunikointikeinon käyttökustannukset ovat huomattavasti suuremmat. Tilanne ei tällöin suinkaan ole optimaalinen, mutta käytännössä on usein valittava pienempi kahdesta pahasta.

Yhteiskirjoitusvälineiden suunnittelijat olettavat usein, että synkronisen välineen käyttäjillä on käytössään jonkinlainen keskustelukanava (Ellis *et al.*, 1991; Baecker *et al.*, 1994). Joissakin yhteiskirjoitusvälineissä on jopa sisäänrakennettu äänikanava (Haake ja Wilson, 1992). Nykyään on puhelimen lisäksi olemassa useita erilaisia ääntä välittäviä verkkosovelluksia, joten äänikanavan toteuttaminen yhteiskirjoitusvälineessä ei välttämättä ole kannattavaa. Sivuutan tästä syystä puheen välittämisen synkronisissa ympäristöissä. Sivuan aihetta muiden kommunikointikeinojen yhteydessä, koska esimerkiksi sanaton viestintä liittyy usein läheisesti puheeseen, esimerkiksi kirjoittaja voi sanoa “Tätä kappaletta tarkoitin” ja osoit-



taa kyseessä olevaa kappaletta. Toinen esimerkki on kirjoittaja, joka kysyy, mitä kohtaa kommentaattori tarkalleen kritisoi, jolloin kommentaattori voi maalata kritiikin kohteena olevan tekstinpätkän.

Neuwirth *et al.* (1990) huomioivat, että yhteiskirjoitusprosessin aikana kehittyvä dokumentti on tärkeä mutta epätäydellinen kirjoitusryhmän kommunikaatiota mahdollistava objekti. Dokumentin tuottaminen on eräänlaista ryhmän jäsenten välistä vuoropuhelua. Johdonmukaisesti yhteiskirjoittamisjärjestelmien yhteydessä käytettävät kommunikaatiokeinot on tarkoitettu tukemaan nimen omaan tuotettavasta dokumentista kommunikoimiseen.

Tarkastelen seuraavaksi kolmea esimerkkiä eri viestintämuodoista. Tekstipohjainen keskusteluohjelma on esimerkki tyypillisesti synkronisesta ja melko välittömästä viestintämuodosta. Tutustumme tekstipohjaiseen keskusteluun alakohdassa 2.3.1. Alakohdassa 2.3.2 tutustumme yhteiskirjoittamiselle tärkeiden huomautusten ja kommenttien käyttöön asynkronisena viestintänä. Sanattoman synkronisen viestinnän muotoihin perehdymme alakohdassa 2.3.3.

### **2.3.1 Tekstipohjainen keskustelu**

Keskusteluohjelmat ovat tuttuja suurelle osalle tietokoneen käyttäjistä ja niiden suosio on suuri. Esimerkiksi pelkästään suosittua ICQ-keskusteluohjelmaa käytti 1990-luvun loppupuolella päivittäin parhaimmillaan yli sata tuhatta käyttäjää (ICQ Incorporated, 2001). Keskusteluohjelma on suosionsa ja helpon käytön takia eräs hyvin tyypillinen valinta kommunikaatiokeinoksi yhteiskirjoitusvälineissä ja ryhmätyövälineissä yleensä. Esimerkiksi 1990-2000 voimakkaasti yleistyneisiin verkkopohjaisiin moninpeleihin, kuten suosittuun Quake-sarjaan (Id Software Corporation, 2001), kuuluu tekstipohjainen keskustelutoiminto. Kuvassa 4 on ruudunkaappaus tyypillisestä peruskeskusteluohjelmasta.



Kuva 4: Ruudunkaappaus Woven Stories II -järjestelmän keskustelutoiminnosta (Gerdt, 2001).

Vaikka keskusteluohjelmat ovat helppokäyttöisiä ja suosittuja, liittyy niihin kuitenkin ongelmia, jotka korostuvat tietokoneavusteisen ryhmätyön yhteydessä kommunikoidessa. Smith *et al.* (2000) luettelee viisi epäkohtaa, joita sisältyy keskusteluohjelmien käyttöön kommunikaation välineenä.

(a) Keskustelijoita voi olla vaikea erottaa toisistaan keskusteluohjelmassa (Smith *et al.*, 2000). Toisin kuin kasvotusten käytävässä keskustelussa tyypillinen keskusteluohjelma näyttää varsin vähän tietoa siitä, kuka viestii. Kuvan 4 ruudunkaappauksessa jokaista viestiä edeltää viestin kirjoittajan nimi tai nimimerkki. Tämä on ainoa vihje viestijän henkilöllisyydestä. Jos keskusteluohjelmaa käyttää usea henkilö samanaikaisesti, niin viestit voivat vilistää sellaista vauhtia, ettei viestejä lukeva henkilö enää ehdi lukea kaikkea tarkkaan. Silloin voi myös syntyä epäselvyyttä, kuka tietyn viestin on kirjoittanut, mihin viesti liittyy ja ketkä yleensäkin osallistuvat keskusteluun.

(b) Keskustelijaohjelman käyttäjä ei tiedä, kuunnellaanko häntä (Smith *et al.*,

2000). Kasvotusten käytävässä keskustelussa keskustelukumppanista näkee, kuunteleeko hän puhujaa. Pienet eleet ja ilmeet antavat puhujalle vihjeitä kuuntelijoiden tunnoista. Keskustelun osapuolet tuntevat sosiaalista läsnäoloa, jossa keskustelijoiden käyttäytymistä ohjaavat hienovaraiset sosiaaliset protokollat. Keskusteluohjelmassa kaikki tämä informaatio karsiutuu pois ja sosiaalisen läsnäolon tunne hälvenee. Viestin kirjoittaja ei tiedä, kuinka muut käyttäjät reagoivat, elleivät nämä kirjoita viestejä, jotka kuvailevat heidän tuntojaan. Pahimmassa tapauksessa kirjoitusohjelman käyttäjä ei tiedä, huomioiko kukaan hänen viestejään.

(c) Keskusteluohjelmassa ei näe, valmisteleeko joku puheenvuoroa (Smith *et al.*, 2000). Ajoitus on tärkeää kasvotusten käytävässä keskustelussa, kun taas keskusteluohjelmassa useat eri tekijät, kuten kirjoitusvaikeudet ja verkkoviiveet, särkevät ajoituksen. Lisäksi käyttäjillä on havaittu taipumusta kirjoittaa lisää viestejä, jos hänelle ei vastata, jolloin keskustelutilanne muuttuu yhä sekavammaksi.

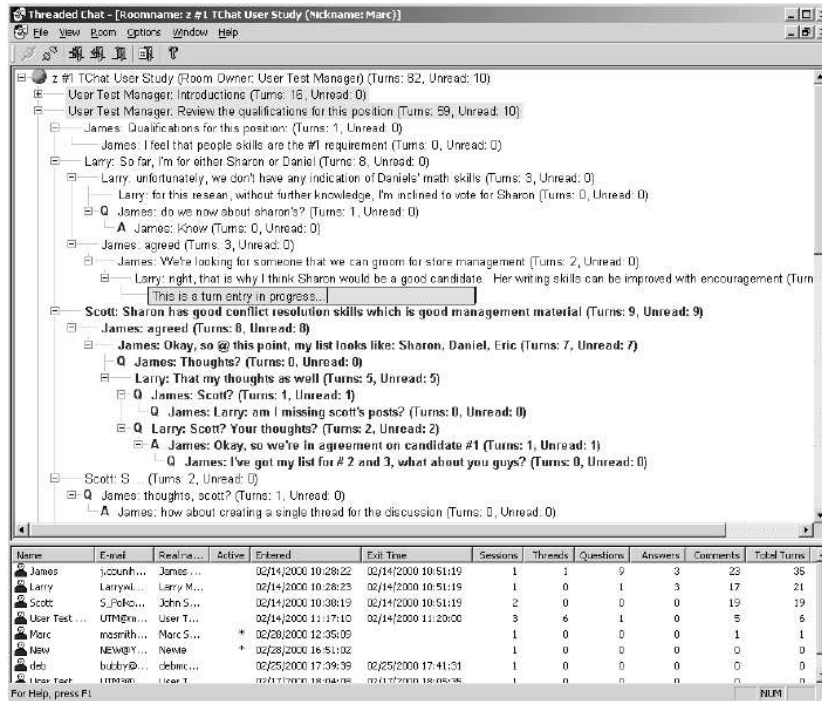
(d) Keskusteluohjelmassa ei ole puheenvuoroja (Smith *et al.*, 2000). Sosiologisissa keskusteluanalyysitutkimuksissa on havaittu, että ihmiset käyttävät kehittyntä vuoropohjaista mekanismeista keskustellakseen johdonmukaisesti ja ymmärrettävästi. Keskustelun osanottajat puhuvat vuorollaan, jolloin keskustelu etenee luontevasti. Perinteinen keskusteluohjelma ei kuitenkaan noudata vuoropohjaista järjestelmää, vaan kirjoitetut viestit näkyvät näytöllä siinä järjestyksessä kuin ne ilmaantuvat keskusteluohjelman palvelimelle. Kasvotusten käytävän keskustelun hienovarainen vuoromekanismi rikkoontuu ja keskusteluohjelman välittämää keskustelua voi olla vaikea seurata.

(e) Sekä Smith *et al.* (2000) että Churchill *et al.* (2000) huomioivat, että keskusteluohjelma ei talleta keskusteluja käyttökelpoisesti. Useat keskusteluohjelmat säilyttävät jonkinlaisen lokitiedoston viesteistä, mutta lokitiedostoja on vaikea tulkita jäljestäpäin niiden sekavuuden vuoksi. Tämän vuoksi keskusteluohjelma ei tuota mitään pysyvää ja käyttökelpoista tuotosta, kuten esimerkiksi sähköpostin vaihto tekee viestejä tallettaessa. Näin syystä niiden käyttökelpoisuus helposti rajoittuu ainoastaan vapaamuotoiseen ”jutusteluun”, koska keskusteluun ei käytännössä voi mitenkään viitata jäljestä päin. Lisäksi keskusteluohjelman käyttökelpoi-

suus asynkronisen työn yhteydessä on huono edellä mainituista syistä (Churchill *et al.*, 2000).

Churchill *et al.* (2000) lisää tärkeän epäkohdan keskusteluohjelmissa: keskusteluohjelman välityksellä käyty keskustelu on vaikea sijoittaa tiettyyn kontekstiin. Jos kaksi henkilöä keskustelee dokumentista keskusteluohjelman välityksellä, joutuvat he käyttämään huomattavan osan puheenvuoroista taustatietoihin luodakseen keskustelulle kontekstin. Olettaen esimerkiksi, että kaikilla keskustelijoilla on sama versio dokumentista, josta keskustellaan, on ryhmän käytettävä puheenvuoroja määritelläkseen, mitä dokumentin kohtaa he käsittelevät. Tilanne mutkistuu entisestään, jos kaikilla ei ole dokumenttia tarkasteltavana. Silloin keskustelijoiden on välitettävä keskusteluohjelman kautta myös osia dokumentista.

Smith *et al.* (2000) ovat kehittäneet säikeistetyn keskusteluohjelman, joka koettaa ratkaista osan edellä mainituista epäkohdista. Säikeistetty keskusteluohjelma rakentuu keskustelusäikeiden varaan. Nämä säikeet koostuvat vuoroista, jotka puolestaan koostuvat jokaisen keskusteluun osallistuvan mahdollisesta viestistä. Kuvassa 5 näkyy, kuinka vuoron aikana syötetyt viestit sijoittuvat hierarkkisesti aina alemmas oikealla säikeen aloittavaan edellisen vuoron viestiin nähden. Ohjelma korostaa uudet viestit lihavoimalla ne. Ajan mittaan viestien teksti muuttuu tavallisemmaksi ja ne sulautuvat aiemmin syötettyihin viesteihin. Tämä piirre auttaa käyttäjiä näkemään, mitkä viestit on syötetty viimeksi, mikä onkin tärkeää, koska viestejä voi ilmestyä mihin tahansa säikeissä.



Kuva 5: Säikeistetty keskusteluohjelma Smith *et al.* (2000).

Säikeiden muodostaminen helpottaa keskustelun seuraamista, koska toisiinsa liittyvät viestit löytyvät helposti. Vuorot toimivat kuten puheenvuorot kasvotusten käytävässä keskustelussa ja jokainen säikeistettyyn keskusteluun osallistuva saa mahdollisuuden syöttää oman viestinsä, joka liittyy tiettyyn viestiin. Kun käyttäjä alkaa syöttää viestiä, ilmestyy asianomaisen säikeen ja vuoron kohdalle rivi, joka ilmoittaa käyttäjän valmistelevan viestiä. Näin muut käyttäjät tietävät, että puheenvuoro on valmisteilla. Säikeiden hierarkkinen rakenne tekee keskusteluun osallistujille selväksi, mihin viestit liittyvät. Tietty kirjoittaja näkee konkreettisesti, vastaako kukaan hänen viestiinsä, koska vastaus ilmestyy sisennettynä hänen viestinsä alle. Säikeistetyssä keskusteluohjelmassa pyritään ylläpitämään sosiaalista kontekstia ja läsnäoloa näyttämällä käyttöliittymän alaosassa tietoa jokaisen käyttäjän aktiivisuudesta ja toiminnasta.

### 2.3.2 Huomautukset ja kommentit

Tuotetun tekstin kommentointi on yhteiskirjoittamisen ja sitä tukevien tietokonevälineiden kannalta erittäin tärkeä kommunikaatiokeino (Neuwirth *et al.*, 2001). Tekstin kommentointi on tyypillisesti asynkronista viestintää, jossa joku henkilö tutkii jonkun muun henkilön kirjoittamaa tekstiä ja liittää siihen erilaisia huomioita eli kommentteja.

Jotkut yhteiskirjoittamisen yhteydessä esiintyvät roolit, kuten alakohdassa 2.2.1 käsitellyt tarkastaja ja konsultti, edellyttävät yhteiskirjoitusvälineeltä hyviä kommentointimahdollisuuksia ollakseen mahdollisia. Toisaalta yhteiskirjoitusprosessin tarkistusvaiheen aikana tekstin kommentointi arvioimis- ja tarkastamismielessä on kommentteja vaativaa puuhaa (kirjoitusprosessista tarkemmin alakohdassa 2.2.2). Neuwirth *et al.* (2001) raportoivat tutkimuksistaan (Cavalier *et al.* (1991), Neuwirth *et al.* (1990), Neuwirth *et al.* (1994a)), joissa he lähestyvät yhteiskirjoittamista kommentointinäkökulmasta. Heidän lähtökohtansa yhteiskirjoittamisen kommentointitarpeiden tutkimiseen on erityisen kiinnostava: munkkien 1100-luvulta alkaen käyttämät käsikirjoitukset ja munkkien kehittämä tapa tehdä huomautuksia teksteihin (Cavalier *et al.*, 1991) (aiheesta lisää kohdassa 4.1). Tutkimustensa perusteella Neuwirth *et al.* (2001) ovat päätyneet neljään yhteiskirjoitusvälineen kommentointitoiminnoille asetettaviin vaatimuksiin.

Dokumentin päätekstin tulee erottua selvästi kommenteista. Kommentoidun dokumentin lukijalle ei missään vaiheessa saa olla epäselvää, mikä osa näkyvästä tekstistä on kommentoivaa tekstiä ja mikä puolestaan kommentoinnin kohteena olevaa tekstiä. Lukijan on helppo perehtyä pelkästään tekstiin huomioimatta kommentteja, jos ne erottuvat selvästi tekstistä. Toisaalta lukija löytää helposti tekstiin liittyvät kommentit halutessaan tutkia niitä, jos kommentit erottuvat selvästi päätekstistä. Yhteiskirjoitusväline voi erottaa kommentit ja tekstiin kohdistuvat huomautukset varsinaisesta tekstistä esimerkiksi korostamalla niitä eri väreillä tai eri kokoisella ja näköisellä fontilla kuin tekstiä, johon ne liittyvät.

Neuwirth *et al.* (2001) korostavat, että kommenttien on oltava näkyvillä pääteks-

tiä luettaessa. Silloin lukija voi helposti vilkaista tiettyyn tekstin kohtaan liittyvän kommentin sisällön minimaalisella vaivannäöllä. Yhteiskirjoitusvälineille on tyyppillistä, että kommentit liitetään erilaisin hyperlinkein päätekstiin, jolloin lukija näkee tekstiin liittyvän huomautuksen vasta aktivoituaan linkin. Neuwirth *et al.* (2001) ovat sitä mieltä, että linkkien seuraaminen aiheuttaa liikaa kognitiivista rasitusta, mikä osaltaan heikentää yhteiskirjoitusvälineen käyttökelpoisuutta.

Sen lisäksi, että kommentit ovat aina näkyvissä, on kommenttien ja päätekstin suhde oltava selkeästi lukijan havaittavissa. Näin lukija näkee heti, mihin päätekstin osaan tietty kommentti viittaa. Yhteiskirjoitusprosessin edetessä kommenttien määrä voi kasvaa suureksikin. Tuotos kokonaisuudessaan muuttuu hankalaksi käsitellä, jos se sisältää paljon vaikeasti jäljitettävissä olevia kommentteja. Neuwirth *et al.* (2001) mainitsevat kommenttien ja päätekstin suhteen ilmaisemisen toteuttamiseksi erilaisten graafisten symbolien käytön ja kommenttien sijoittamisen tietyllä johdonmukaisella tavalla päätekstiin nähden.

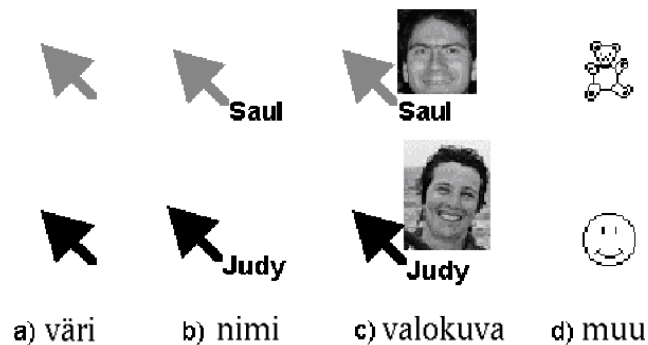
Viimeinen neljästä kommentoinnille asetettavasta vaatimuksesta on se, että kommentin kirjoittaja selviää helposti (Neuwirth *et al.*, 2001). Lukijan on mieluiten nähtävä yhdellä silmäyksellä, kuka tietyn kommentin on kirjoittanut. Dokumentin lukija saattaa olla kiinnostunut jonkun tietyn henkilön tuottamista kommentteista, tai hänen on vastattava kommentaattorille henkilökohtaisesti. Yhteiskirjoitusprosessin jäsenillä on rooliensa (2.2.1) mukaisesti erilainen painoarvo päätöksiä tehdessä ja tuotoksia arvioitaessa, joten kommenttien käyttökelpoisuutta vesittää epätietoisuus niiden kirjoittajasta.

Baecker *et al.* (1993) ovat sitä mieltä, että huomautuksia ja kommentteja pitäisi pystyä tallettamaan tekstinä, piirrustuksina ja ääniviesteinä. Monimuotoisten kommenttien tukemista puoltaa tutkimus, jossa havaittiin, että tekstikommenteilla kommentoitiin enemmän sisällöllisiä asioita (substanssia), kun taas äänikommenteilla oli taipumus käsitellä asioita, jotka liittyvät kohdeyleisöön ja kirjoituksen tarkoitukseen ja päämäärään (Neuwirth *et al.*, 1994a).

### 2.3.3 Osoitinvälineet ryhmätyöohjelmissa

Käytämme runsaasti sanatonta viestintää kasvotusten tapahtuvassa kommunikointiossa. Kiivastunut puhuja voi tehostaa sanomaansa irvistämällä tai heristämällä sormiaan. Yhteiskirjottamiskontekstissa sanatonta viestintää edustaa esimerkiksi tiettyä dokumentin kohtaa osoittaminen. Sanaton viestintä tukee usein puheviestintää, kirjoittaja voi esimerkiksi sanoa “tuossa on virhe” ja osoittaa kyseistä kohtaa tekstissä. Yhteiskirjoitusväline voi tukea sanatonta viestintää kaikille käyttäjille näkyvien osoitinvälineiden, kuten hiiren kursorin, avulla.

Osoitinvälineillä on mahdollista välittää ryhmälle tietoa tietyn käyttäjän identiteetistä, toimista ja eleistä (Greenberg *et al.*, 1996). Kuvassa 6 on esimerkkejä siitä, kuinka osoitin voi välittää tietoa käyttäjän identiteetistä. Kuvan 6 osoittimet ovat alakohdassa 3.3.3 käsittelemäni GroupKit-kehitystyökalun komponentteja (Greenberg *et al.*, 1996).



Kuva 6: Käyttäjän identiteetin välittäviä osoitinratkaisuja, mukaelma lähteestä (Greenberg *et al.*, 1996).

Begole (1998) antaa esimerkin yksinkertaisesta tavasta välittää tietoa tietyn käyttäjän toiminnasta. Kuvassa 7 (a) näkyy hiiren osoitin, joka kuuluu Laura-nimiselle käyttäjälle. Kun Laura napsauttaa hiirtään ryhmätyöohjelmassa, ilmestyy osoittimen viereen hetkellisesti teksti “click!” sopivasti erottuvalla värillä (b). Muut käyttäjät näkevät Lauran toiminnon hiiren käytön tunnusomaisen napsahdusää-



nen kuulemisen sijasta.



Kuva 7: Esimerkki käyttäjän toiminnan välittämisestä osoittimen avulla (Begole, 1998).

## 2.4 Tietoisuus

Tietoisuustutkimuksella on juuret sotilasilmailussa, jonka parista se on levinnyt muille aloille, joissa ihmiset ovat vuorovaikutuksessa hyvin dynaamisten ja informaattiorikkaiden ympäristöjen kanssa (Gutwin ja Greenberg, 1999). Tietoisuus on esillä jokapäiväisissä toimissamme, me esimerkiksi ylläpidämme mielikuvaa ympäröivästä liikenteestä ajaessamme autolla. Toimiessamme ryhmässä meillä on automaattisesti mielikuva siitä, ketkä ovat paikalla, kuka tekee mitäkin ja niin edelleen. Dourish ja Bellotti (1992) määrittelevät ryhmätietoisuuden omalle työskentelylle kontekstin antavaksi ymmärrykseksi muiden ryhmän jäsenten toiminnasta. Ryhmän jäsenet käyttävät tietoisuutta toisistaan koordinoidessaan toimiaan, kommunikoidessaan toisten kanssa ja työskennellessään tietyn tehtävän parissa. Gutwin *et al.* (1995) määrittelevät ryhmätietoisuuden olevan osa sitä synergiaa, joka antaa ryhmille mahdollisuuden olla tehokkaampia kuin yksilön. Jos yhteistyötä tekevän ryhmän jäsenet eivät ole täysin tietoisia muista, on ryhmän työskentelyä vaikea koordinoita.

Gutwin ja Greenberg (1999) mainitsevat kolme syytä, jotka vaikeuttavat ryhmän tietoisuuden muodostumista tietokoneavusteisessa ryhmätyöskentelyssä: Ensimmäiseksi vuorovaikutus ryhmätyöohjelman käyttöliittymän kautta karsii pois suuren osan siitä tiedosta, jota kasvotusten käytävissä keskustelussa esiintyy. Toiseksi käyttäjän toiminta ryhmätyöohjelman jaetussa työympäristössä ei tuota kaikkea

sitä informaatiota, jota syntyy tavallisessa työympäristössä toimittaessa. Kolmanneksi ryhmätyöohjelmat laiminlyövät usein senkin tietoisuusinformaation välittämisen käyttäjälle, jota järjestelmän kautta olisi mahdollista tuottaa. Näiden seikkojen vuoksi tietokoneitse tehtävän ryhmätyöskentelyn tapauksessa ryhmän tietoisuus tilastaan ei ole itsestään selvä, vaan tietoisuutta on tuettava tarkoitusta varten suunnitelluilla keinoilla.

Gutwin ja Greenberg (1999) tarkentavat tietoisuuden käsitettä määrittelemällä keskeisiä piirteitä, jotka vetävät yhteen tietoisuustutkimuksen pääkohdat. Tietoisuus on aikaan ja tilaan sidottua tietämystä tietyn ympäristön tilasta. Tietoisuutta on ylläpidettävä jatkuvasti, koska ympäristö muuttuu ajan kuluessa. Ihmiset tutkivat ympäristöä ja ovat vuorovaikutuksessa sen kanssa, jolloin he ylläpitävät tietoisuuttaan tämän toiminnan kautta. Vaikka tietoisuus on välttämätön asia ryhmätyön kannalta, se on kuitenkin toisarvoinen tavoite, koska ryhmätyön ensisijainen tavoite on suorittaa tietty tehtävä ympäristössä.

Liika tietoisuusinformaatio voi kääntyä itseään vastaan, koska korostunut tietämys muiden ryhmän jäsenten tekemisistä vaikeuttaa keskittymistä (Gutwin *et al.*, 1995). Tästä syystä on hyvä antaa käyttäjille mahdollisuus vaikuttaa heidän vastaanottamaansa tietoisuusinformaation määrään. Tietoisuutta voidaan rentouttaa ryhmän jäsenten kognitiivisen kuormituksen välttämiseksi (Fussell *et al.*, 1998). Koska suuri osa tietoisuusinformaatiota välittyy ryhmätyöohjelmissa käyttöliittymän kautta, voidaan tietoisuutta rentouttaa samoin keinoin kuin käyttöliittymää (katso alakohta 2.1.2).

Alakohdassa 2.4.1 tarkastelen tietoisuuden eri lajeja keskittyen erityisesti ryhmätyöohjelmien kannalta tärkeään työtilatietoisuuteen. Alakohdassa 2.4.2 tutustumme joihinkin esimerkkeihin siitä, kuinka käyttöliittymä voi välittää tietoisuusinformaatiota yhteiskirjoitusvälineessä.

### 2.4.1 Tietoisuus ryhmätyöohjelmissa

Dourish ja Bellotti (1992) jakavat ryhmätyöohjelmistoissa esiintyvät tietoisuutta tukevat mekanismit informatiivisiin, roolien kautta rajoittaviin ja jaettuun palautteeseen perustuviin. Ryhmätyöohjelmien suunnittelijoiden on otettava kantaa kolmeen pääkysymykseen tietoisuutta tukevaa ympäristöä suunnitellessaan (Gutwin ja Greenberg, 1999). Aluksi on päätettävä, mitä informaatiota yleensäkin ympäristöstä kerätään ja välitetään ryhmän jäsenille. Toiseksi on päätettävä, mitä tietoisuusinformaatiota toiminta ympäristössä vaatii ja milloin se on esitettävä. Kolmanneksi on päätettävä, kuinka ryhmätyöohjelma esittää tietoisuusinformaation käyttäjilleen.

Gutwin *et al.* (1995) jakavat ryhmän tietoisuuden neljään kategoriaan: vapaamuotoiseen, rakenteelliseen, sosiaaliseen ja työtilaan rajoittuvaan tietoisuuteen. Vapaamuotoinen tietoisuus on henkilön tietämys siitä, ketä muita on paikalla samassa tilassa ja missä he ovat suhteessa henkilöön. Begole (1998) nimittää vapaamuotoista tietoisuutta tietoisuudeksi läsnäolosta. Rakenteellinen tietämys viittaa jäsenen tietoisuuteen ryhmän rakenteista, kuten rooleista, asemasta, vastuista, arvoasteesta ja ryhmässä tapahtuvista prosesseista. Rakenteellinen tietämys on oleellista ryhmätyön onnistumisen kannalta. Sosiaalinen tietoisuus on tietämystä muiden ryhmäläisten sitoutumisesta yhteiseen työskentelyyn. Esimerkiksi ryhmän jäsenen kiinnostus asioihin, huomion kohde ja tunteet on sosiaalista tietoisuutta, jota välittävät eleet, katse ja kehonkieli. Begole (1998) on kehittänyt oman nimen myös sosiaaliselle tietoisuudelle, hän käyttää nimitystä tietoisuus sitoutumisesta.

Kaikkia edellä mainittuja tietoisuuden lajeja esiintyy missä tahansa kasvotusten tapahtuvassa ryhmätilanteessa. Nämä tietoisuuden lajit yhdistyvät ryhmätyötilanteessa muodostaen työtilatietoisuuden, jonka Gutwin ja Greenberg (1996) sanovat olevan tietämystä, jota henkilö tarvitsee muiden ryhmän jäsenten vuorovaikutuksesta jaetusta työtilasta tehdäkseen heidän kanssaan yhteistyötä. Työtilatietoisuus määrittelee sen tiedon, joka on välttämätöntä ryhmätyöohjelmassa ryhmätyön onnistumiseksi. Taulukossa 8 on lueteltu asioita, jotka ovat osia työtilatietoisuudesta

sekä kysymyksiä, joita ryhmätyöhön osallistuva saattaa pohtia. Taulukon 8 esittämät kysymykset on kirjattu samanaikaiseen työskentelyyn sopiviksi. Kysymysten muotoileminen imperfektiin tuottaa vastaavat kysymykset eriaikaiseen työskentelyyn nähden.

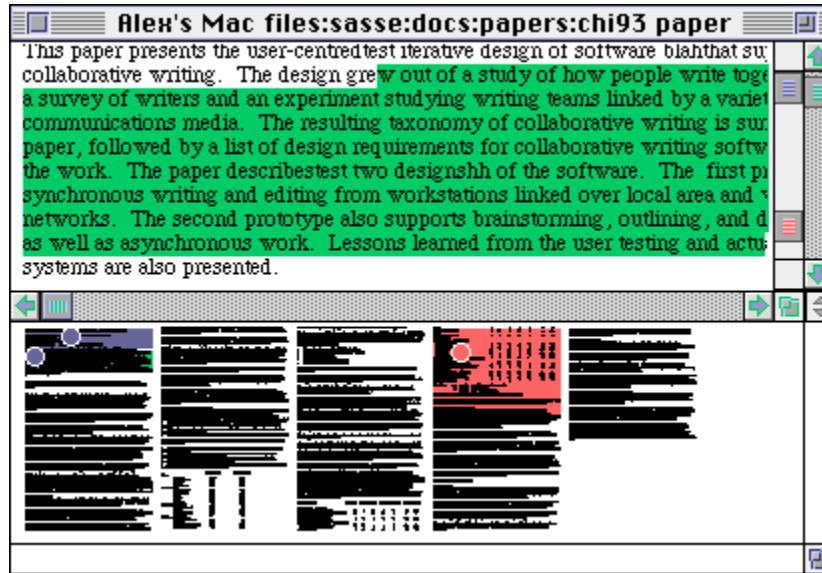
Taulukko 8: Ryhmätyövälineiden suunnittelussa huomioitavia työtilatietoisuustekijöitä (Gutwin ja Greenberg, 1996).

<b>Asia</b>	<b>Kysymys</b>
identiteetti	Kuka ottaa osaa toimintaan?
sijainti	Missä he ovat?
aktiivisuustaso	Ovatko he aktiivisia työtilassa? Kuinka nopeasti he työskentelevät?
aikomukset	Mitä he tekevät? Mitkä ovat heidän tämän hetkiset aktiviteettinsa ja tehtävänsä?
muutokset	Mitä muutoksia he tekevät? Missä muutoksia tehdään?
objektit	Mitä objekteja he käyttävät?
ulottuvuus	Mitä he näkevät?
kyvyt	Mitä he voivat tehdä?
vaikutuspiiri	Missä he voivat vaikuttaa?
odotukset	Mitä he haluavat minun tekevän seuraavaksi?

#### 2.4.2 Käyttöliittymä tietoisuutta tukemassa

Alakohdassa 2.3.3 tutustuimme osoitinvälineisiin, jotka välittävät tietoa käyttäjien toimista ja identiteetistä. Jaetut osoitinvälineet voidaan mieltää kommunikatiovälineiden lisäksi myös käyttöliittymän tietoisuutta tukeviksi komponenteiksi. Tässä alakohdassa esittelen muita käyttöliittymien tietoisuusmekanismeja, joiden ensisijainen tehtävä on välittää tietoisuusinformaatiota. SASSE (Synchronous Asynchronous Structured Shared Editor) on yhteiskirjoitusväline, joka kehitettiin

alunperin tukemaan nimenomaan hajautettua reaaliaikaista työskentelyä (Baecker *et al.*, 1993, 1994). Kuvassa 8 on SASSE-järjestelmän ruudunkaappaus, jossa näkyy kaksi toisiaan tukevaa tietoisuusinformaatiota välittävää mekanismia: dokumentin hahmonäkymä ja usean käyttäjän vierityspalkit.



Kuva 8: Ruudunkaappaus SASSE-yhteiskirjoitusvälineestä (Baecker *et al.*, 1993).

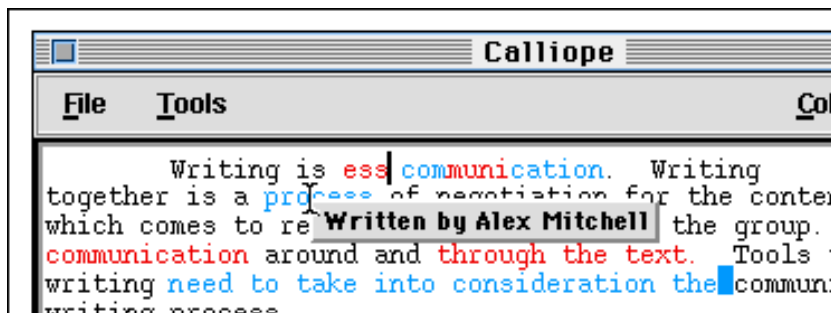
SASSE:n käyttöliittymä välittää tietoa monista työtilatietoisuuden tekijöistä, joihin tutustuimme alakohdassa 2.4.1. Hahmonäkymä välittää tietoa muiden käyttäjien sijainnista dokumentissa näyttämällä koko dokumentin hahmon pienoiskoos- sa ja merkitsemällä jokaisen käyttäjän näkymän dokumentista. SASSE:n hahmonäkymän tapauksessa kirjoittajien sijainti ilmaistaan värittämällä se osa hahmonäkymästä, jonka käyttäjät näkevät. Lisäksi käyttäjien kursorin sijainti näkyy hahmonäkymässä pienenä väritettynä ympyränä. Hahmonäkymää kutsutaan myös tutkanäkymäksi (Gutwin ja Greenberg, 1996; Begole, 1998), koska sen voidaan ajatella olevan lennonjohdon tutkaruutua vastaava näkymä käyttäjistä siinä avaruudesta, jossa he työskentelevät.

Usean käyttäjän vierityspalkit välittävät samaa informaatiota kuin hahmonäkymä-

kin. Kuvan ruudunkaappauksessa 8 näkyy kaksi vierityspalkkia, joilla tavallisen tekstinkäsittelyohjelman tapauksessa hallitaan sitä, mikä osa dokumentista näkyy. Oikeanpuoleisin vierityspalkki onkin tavallinen, paikalliselle käyttäjälle tarkoitettu dokumentinhallintavierityspalkki. Sen sijaan vasemmanpuoleinen vierityspalkki on indikaattori muiden kirjoittajien näkymästä. Kuvassa 8 vasemmanpuoleisessa pystyvierityspalkissa näkyy kahden käyttäjän sijainnit suhteessa paikalliseen käyttäjään. Paikallinen käyttäjä voi nopeasti säätää näkemänsä dokumentin osan vastaamaan toisen käyttäjän näkymää siirtämällä oman vierityspalkkinsa säätönappi samaan kohtaan kuin toisen käyttäjän näkymä.

Käyttäjä voi saada tietoisuusinformaatiota muuttamalla käyttöliittymän kytkentää. Kohdassa 4.2 käsittelemäni yhteiskirjoitusväline Calliope antaa käyttäjälleen mahdollisuuden kytkeä käyttöliittymänsä tiukasti jonkun toisen käyttöliittymään, jolloin kytketyt käyttöliittymät ovat tiukassa WYSIWIS-tilassa (alakohta 2.1.2). Tietty käyttäjä voi saada tietoa jonkun muun käyttäjän tekemisistä esimerkiksi kytketyillä hetkeksi toisen käyttäjän käyttöliittymään. Toinen tapa hyödyntää kytkennän muuttamista tietoisuusinformaation määrän ja laadun säätämiseksi on kohdassa 4.1 tarkastelemani PREP-ryhmäeditorin vuorovaikutusparametrit (Neuwirth *et al.*, 1994b). PREP antaa käyttäjän säätää erityisten vuorovaikutusparametrien avulla kytkentää WYSIWIS-käyttöliittymää rentouttamiseksi. Tarkasteleminen PREP:in tätä piirrettä tarkemmin alakohdassa 4.1.2.

Eräs yhteiskirjoittamisryhmän jäsenille tarpeellisimmista tiedoista on tieto siitä, kuka mitäkin on kirjoittanut. Tietoisuus tekstin tuottajista on luonteeltaan pysyvää ja se vastaa lähinnä alakohdassa 2.4.1 esille tulleisiin työtilatietoisuuteen liittyviin kysymyksiin, kuten mitä muut ovat tehneet ja missä he ovat vaikuttaneet. Yhteiskirjoitusväline Calliopessa kirjoittajat erotellaan väreillä ja tietyn kirjoittajan teksti on mahdollista nähdä kirjoittajansa värissä. Kuvassa 9 näkyy, kuinka Calliope värittää tekstin kirjaimen tarkkuudella kirjoittajan mukaan. Lisäksi Calliope näyttää tietyn tekstinpätjän kirjoittajan henkilöllisyyden tarvittaessa.



Kuva 9: Esimerkki tekstin kirjoittajan ilmaisemisesta värein (Mitchell, 1996).

## 2.5 Yhteenveto

Tämän luvun johdannossa jaottelin tietokoneavusteiseen yhteiskirjoittamiseen liittyvät piirteet neljään kategoriaan: yhteinen dokumentti, koordinointi, kommunikointi ja tietoisuus. Jokainen kategorioista on yhteiskirjoittamisen kannalta välttämätön. Ilman ryhmän jäsenten kesken jaettua dokumenttia ei kirjoitusryhmä voi kirjoittaa yhdessä. Ellei ryhmätyötä koordinoida, on kysymys yksilöiden työskentelystä. Ryhmätyön koordinointi edellyttää kommunikointia jäsenten välillä ja ryhmän jäsenten on mahdotonta työskennellä, elleivät he ole tietoisia toistensa tekemisistä. Tässä kohdassa esitän edellä mainittuun jakoon perustuen yhteenvedon tietokoneavusteisen yhteiskirjoittamisen peruspiirteistä käyttäjän kannalta katsottuna.

**Yhteinen dokumentti.** Tietokoneavusteinen yhteiskirjoittaminen perustuu yhteisen dokumentin käsittelyyn ja ylläpitoon, LaMarca *et al.* (1999) nimittää yhteiskirjoittamista osuvasti dokumenttikeskeiseksi yhteistyöksi. Kohdassa 2.1 keski-tyin keinoihin, joilla kirjoitusryhmä yhteistä dokumenttia muokkaa ja hallitsee. Taulukossa 9 on yhteenveto kohdassa 2.1 käsittelemistäni asioista.

Taulukko 9: Keinoja hallita yhteistä dokumenttia tietokoneavusteisessa yhteiskirjoittamisessa kohdasta 2.1.

Asia	Selitys
perustekstinkäsittely	Yhteiskirjoitusvälineessä on perustekstinkäsittelyvalmiudet, sitä voi käyttää tavallisena tekstinkäsittelyohjelmana tarvittaessa.
käyttöliittymän kytkentä	Järjestelmä näyttää kaikille käyttäjilleen saman jaetun näkymän dokumenttiin, mutta mahdollistaa myös käyttäjien kesken erilaiset näkymät siihen.
versionhallinta	Käyttäjät voivat hyödyntää versionhallintaa dokumentin hallitsemiseksi.

**Koordinointi.** Ryhmätyöskentelyä on koordinoitava, jotta ryhmän jäsenet työskentelevät yhteisen päämäärän eteen. Käsittelin kohdassa 2.2 keinoja koordinoida yhteiskirjoitushanketta tietokoneavusteisen yhteiskirjoittamisen kontekstissa, joista olen tehnyt yhteenvedon taulukkoon 10.



Taulukko 10: Keinoja koordinoita tietokoneavusteista yhteiskirjoittamista kohdasta 2.2.

<b>Asia</b>	<b>Selitys</b>
roolit, sosiaaliset protokollat	Järjestelmä huomioi ja tukee kirjoitusprosessiin ja järjestelmän tekniseen tukeen liittyviä rooleja mahdollistaen näin relevantit sosiaaliset protokollat.
kirjoitusprosessi	Järjestelmä tukee kirjoitusryhmää tarjoamalla keinoja hallita kirjoitusprosessia ja mahdollistamalla useita erilaisia työmuotoja, kuten synkronisen ja asynkronisen työskentelyn.
kirjoitussessio	Järjestelmässä on mekanismit kirjoitusryhmän jäsenten oikeuksien ja tuotosten hallinnoimiseen sekä käyttäjien ulottuvilla oleva virtuaalinen työtila.

**Kommunikointi.** Yhteiseen tavoitteeseen pyrkivän ryhmän perustyökalu on kommunikointi. Ryhmän jäsenet keskustelevat keskenään päästäkseen yhteisymmärrykseen siitä, mitä on tehtävä, kuka mitäkin tekee ja millä aikataululla asiat tehdään. Taulukossa 11 on yhteenveto keinoista mahdollistaa kirjoitusryhmän jäsenten välistä kommunikaatiota kohdasta 2.3.

Taulukko 11: Keinoja mahdollistaa kommunikointi tietokoneavusteisessa yhteiskirjoittamisessa kohdasta 2.3.

<b>Asia</b>	<b>Selitys</b>
vapaamuotoinen keskustelu	Järjestelmä mahdollistaa kirjoittajien välisen synkronisen keskustelun dokumenttiin liittyen tekstipohjaisesti tai puheena.
huomautukset ja kommentit	Järjestelmä tukee asynkronista viestintää mahdollisuudella liittää huomautuksia ja kommentteja dokumenttiin.
sanaton viestintä	Yhteiskirjoitusväline mahdollistaa sanattoman viestinnän erilaisilla jaetuilla osoitinvälineillä.

**Tietoisuus.** Tietoisuus työympäristöstä ja siinä vaikuttavista asioista on välttämätöntä ryhmätyöhön osallistuvalla henkilöllä. Yhteiskirjoitusvälineen tulee välittää tietoisuusinformaatiota käyttäjilleen kohdassa 2.4 mainituilla tavoilla, jotta se tukisi ryhmän työskentelyä mahdollisimman hyvin. Taulukossa 12 on lyhyt yhteenveto kohdassa 2.4 käsitellyistä asioista.

Taulukko 12: Ryhmän tietoisuuden ylläpitäminen tietokoneavusteisessa yhteiskirjoittamisessa (alakohta 2.4).

<b>Asia</b>	<b>Selitys</b>
työtilatietoisuus	Järjestelmä välittää tietoa käyttäjilleen erilaisista tekijöistä, joista heidän on oltava tietoisia pystyäkseen työskentelemään yhteiskirjoitusvälineen avulla.
käyttöliittymä tukee tietoisuutta	Suuri osa järjestelmän välittämästä tietoisuusinformaatiosta välittyy käyttöliittymän kautta.

### 3 Yhteiskirjoitusvälineen toteuttaminen

Tässä luvussa tutustumme asioihin, joita yhteiskirjoitusvälineen toteuttamisesta kiinnostuneiden on hyvä tietää. Koska yhteiskirjoittaminen kuuluu tietokoneavusteisen ryhmätyön piiriin, käytän tässä luvussa usein termejä ryhmätyöohjelma ja usean käyttäjän järjestelmä termin yhteiskirjoitusjärjestelmä sijasta. Yleistetympään terminologian käyttö on perusteltua, koska kirjallisuudessa erotetaan lähes poikkeuksetta ryhmätyöohjelmien tarkemmat toiminnot ja sovelluskohteet teknisiä ratkaisuja kuvailtaessa.

Yhteiskirjoitusvälineen toteutus eroaa tavallisen yhden käyttäjän tekstinkäsittelyohjelman toteutuksesta siinä, että yhteiskirjoitusvälineen on tuettava ryhmätyötä. Tämä vaatimus tekee ryhmätyöohjelmista paljon monimutkaisempia kuin yhden käyttäjän ohjelmat ovat. Ryhmätyöohjelmien erityispiirteet näkyvät selvimmän järjestelmäarkkitehtuuria tutkittaessa. *Järjestelmäarkkitehtuuri* kuvaa, minäkalaisista komponenteista järjestelmä on toteutettu ja kuinka komponentit viestivät keskenään (Dewan, 1996). Ryhmätyöohjelmien arkkitehtuuriratkaisut ovat tyypillisesti *hajautettuja*, eli järjestelmän komponentit toimivat eri tietokoneissa kommunikoiden verkon välityksellä (Begole, 1998). Käsittelen ryhmätyöohjelmien erilaisia arkkitehtuuriratkaisuja Rothin ja Ungerin (2000) kehittämän luokittelumallin jakoa löyhästi noudatellen kohdassa 3.1.

Eräs tärkeimmistä tehtävistä, joita ryhmätyöohjelmalle asetetaan on yhteisen tuotoksen esittäminen kaikille ryhmän jäsenille yhtenäisessä esitysmuodossa (Dewan, 1996). *Yhtenäisyyden hallinnalla* tarkoitetaan ryhmätyöohjelmista puhuttaessa järjestelmän käsittelemän informaation yhtenäisyyden takaamista kaikille käyttäjille. Tutustumme yhtenäisyyden hallintaan kohdassa 3.2.

Luvun päätteeksi teen kohdassa 3.3 lyhyen katsauksen ryhmätyövälineiden kehitystyökaluihin. *Kehitystyökalujen* tarkoituksena on helpottaa ryhmätyöohjelmien tekemistä tarjoamalla niiden toteuttajille valmiita ryhmätyöpiirteitä tukevia komponentteja, kirjastoja ja sovelluskehyskiä. Kehitystyökalujen avulla ryhmätyöoh-

jelmien toteuttaminen ei välttämättä ole sen monimutkaisempi tehtävä kuin tavallisen yhden käyttäjän järjestelmän toteuttaminen. Luvun päättää yhteenveto ryhmätyöohjelman toteutukseen liittyvistä asioista alakohdassa 3.4.

### 3.1 Järjestelmäarkkitehtuuri

Tässä kohdassa käsittelen erilaisia arkkitehtuuriratkaisuja, joita käytetään ryhmätyöohjelmia, myös yhteiskirjoitusvälineitä, toteutettaessa. Järjestelmäarkkitehtuuri on kuvaus komponenteista, joista järjestelmä koostuu ja komponenttien välisistä suhteista (Dewan, 1996). Järjestelmäarkkitehtuuri kuvaa usein myös komponenttien sijoittumista eri tietokoneisiin, jolloin käytetään myös nimitystä *hajautusarkkitehtuuri* (Roth ja Unger, 2000). Ryhmätyöjärjestelmät ovat usean käyttäjän järjestelmiä, jotka ovat väistämättä monimutkaisempia kuin yhden käyttäjän järjestelmät. Ryhmätyöohjelmien arkkitehtuuriratkaisut ovat tyypillisesti hajautettuja, eli järjestelmän komponentit toimivat eri tietokoneissa kommunikoiden verkon välityksellä (Begole, 1998). Ryhmätyöohjelmien arkkitehtuuriratkaisujen kaksi ääripäätä ovat keskitetty ja toistettu arkkitehtuuri, näiden ääripäiden välimuotoja kutsutaan hybridiarkkitehtuureiksi (Dewan ja Choudhary, 1992; Begole, 1998; Hofte, 1998; Roth ja Unger, 2000). Käytän tässä kohdassa käsitettä *palvelin* merkitsemään tietokonetta, johon keskitetyt komponentit on sijoitettu.

Ryhmätyöohjelmat voivat olla *yhteistyötietoisia*, tai *yhteistyön suhteen läpinäkyviä* (Begole, 1998). Yhteistyötietoiset ryhmätyöohjelmat on alunperin suunniteltu ryhmätyöohjelmiksi, tukemaan käyttäjäryhmän työskentelyä. Yhteistyön suhteen läpinäkyvät ryhmätyöohjelmat ovat alunperin yhden käyttäjän järjestelmiä, joista jokin ulkoinen mekanismi tekee usean käyttäjän järjestelmiä. Yhteistyötietoisten ryhmätyöohjelmien piiriin kuuluvat sekä asynkroniset että synkroniset sovellukset, kun taas yhteistyön suhteen läpinäkyvät ryhmätyöohjelmat ovat tyypillisesti synkronisia.

Jos ryhmätyöohjelman toiminta perustuu siihen, että jokin sen komponenteista on

jaettu kaikkien muiden komponenttien kesken, sanotaan sen olevan *keskitetty arkkitehtuuriratkaisu*. Tyypillinen esimerkki keskitetystä arkkitehtuurista on palvelintietokoneella toimiva tietokantasovellus, johon asiakasohjelmat ovat yhteydessä verkon välityksellä. Tutustumme keskitettyyn arkkitehtuuriratkaisuun ryhmätyöohjelmien kannalta katsottuna alakohdassa 3.1.1.

*Toistettu arkkitehtuuri* on keskitetyn arkkitehtuurin vastakohta, koska sillä ei ole lainkaan keskitettyjä komponentteja. Sen sijaan toistettua arkkitehtuuria ilmentävä järjestelmä koostuu useista eri tietokoneissa toimivista järjestelmän ilmentymistä, *toisinoista*, jotka kommunikoivat keskenään. Toistettu järjestelmä on aina myös hajautettu, koska järjestelmä koostuu kokonaisuudessaan useista ilmentymistä, vaikka tietyssä tietokoneessa suoritettava ilmentymä sisältäisikin kaikki järjestelmän komponentit. Toistetun ryhmätyöohjelman toiminnan kannalta on oleellista, että hajautetut toisinnot ovat samassa tilassa eli käsittelevät samaa tietoa. Tarkastelemme toistettua arkkitehtuuria alakohdassa 3.1.2.

Käytännössä on harvinaista, että mikään ryhmätyöohjelma edustaa puhtaasti keskitettyä tai toistettua arkkitehtuuria. Useimmat nykyisistä järjestelmistä edustavat *hybridiarkkitehtuuria*, jossa on sekä toistettuja että keskitettyjä komponentteja (Roth ja Unger, 2000). Toinen käytännössä esiintyvä arkkitehtuuriratkaisuperhe on *asynkroniset arkkitehtuuriratkaisut*, joissa komponenttien hajautus ei ole symmetrinen. Edellä mainituista arkkitehtuuriratkaisuista poiketen *usean palvelimen arkkitehtuurissa* palvelinkomponentit on hajautettu useaan tietokoneeseen eli järjestelmä sisältää enemmän kuin yhden palvelimen. Tutustumme hybridiarkkitehtuureihin, asynkronisiin arkkitehtuureihin ja usean palvelimen arkkitehtuuriin alakohdassa 3.1.3.

Käytän tämän kohdan alakohdissa 3.1.1-3.1.3 Rothin ja Ungerin (2000) luokittelumallinsa yhteydessä esittelemää notaatiota arkkitehtuurin abstraktiolla. Rothin ja Ungerin (2000) luokittelumallissa kuvataan järjestelmä kolmen pääkomponentin kautta, jotka ovat sovelluksen ydin, ikkunointijärjestelmä ja koordinoitukomponentti. Sovelluksen ydin (application core) kuvaa järjestelmän toimintoja ja se jakautuu kahteen alakomponenttiin: toiminnallinen ydin (functional core) vastaa

toimintojen toteutuksesta ja esityskomponentti (presentation) vastaa sovelluksen ytimessä tiedon esittämisestä. Sovelluksen ydin hoitaa myös tiedon pysyväistal-  
tuksen esimerkiksi tietokantaa tai tiedostoja ylläpitämällä.

Rothin ja Ungerin (2000) luokittelumallissa ikkunointijärjestelmä (window sys-  
tem) huolehtii käyttöliittymästä, eli se esittää tietoa käyttäjälle ja huolehtii  
käyttäjän syötteiden käsittelystä. Ikkunointijärjestelmällä tarkoitetaan Unixin X-  
Windowsin kaltaista käyttöjärjestelmään integroitua järjestelmää. Järjestelmä on  
mahdollista toteuttaa ilman ikkunointijärjestelmää, jolloin käyttöliittymätoimin-  
not liitetään esityskomponenttiin tai siihen liittyvään muuhun käyttöliittymä-  
komponenttiin. Koordinaatikomponentti (coordination) on vastuussa järjestel-  
män suorittamisesta hajautetussa ympäristössä, se huolehtii yhtenäisyyden hal-  
linnasta (3.2) ja session hallinnasta (2.2.3). Ikkunointijärjestelmän tavoin erillistä  
koordinaatikomponenttia ei tarvita kaikissa arkkitehtuuriratkaisuissa, jolloin sen  
tehtävät hoitaa sovelluksen toiminnallinen ydin.

Alakohdassa 3.1.4 tutustumme Pattersonin (1995) kehittämään ryhmätyöohjel-  
man arkkitehtuuriratkaisun. Pattersonin niin kutsuttu vetoketjuarkkitehtuuri (Be-  
gole, 1998) on yksinkertainen mutta ilmaisuvoimainen malli kuvata ryhmätyöoh-  
jelmien arkkitehtuuria jakamalla järjestelmä neljään loogiseen tasoon niiden kä-  
sittelemän tiedon perusteella. Vetoketjumallia käytetään tietokoneavusteista ryh-  
mätyötä käsittelevässä kirjallisuudessa kuvaamaan järjestelmien perusratkaisu-  
ja. Pattersonin (1995) vetoketjumalli käyttää järjestelmän kuvauksessa korkeam-  
paa abstraktiotasoa kuin Rothin ja Ungerin (2000) luokittelumalli. Vetoketjumal-  
li myös korostaa järjestelmän tasojen välistä kytkentää. En tarkastele alakohdassa  
3.1.4 arkkitehtuuriratkaisujen etuja ja haittoja, koska käsittelen niitä tämän kohdan  
muissa alakohdissa, vaan keskityn tuomaan esiin vetoketjumallin näkökulman ja  
arkkitehtuurin kuvaustavan Rothin ja Ungerin (2000) luokittelumallille vaihtoeh-  
toisena tapana esittää ryhmätyöohjelman arkkitehtuurin kuvaus.

### 3.1.1 Keskitetty arkkitehtuuri

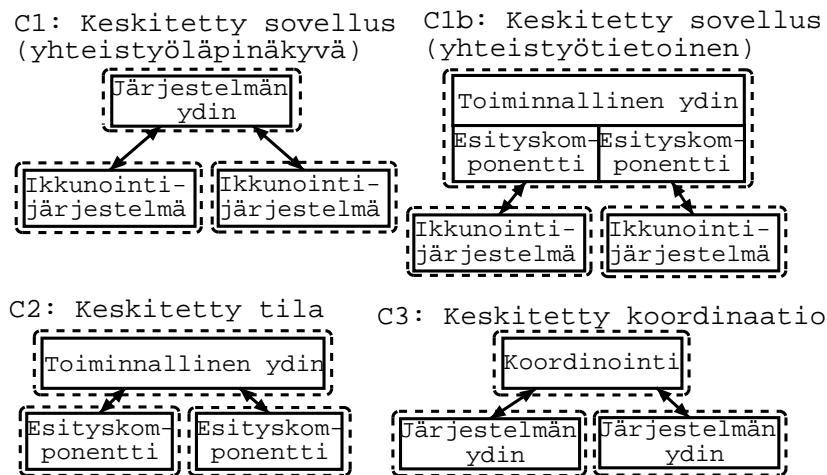
Keskitetty arkkitehtuuri perustuu yhteen tietyssä käyttöjärjestelmäprosessissa toimivaan prosessiin, johon usea käyttäjä on yhteydessä (Dewan ja Choudhary, 1992). Keskitettyä arkkitehtuuria noudattava järjestelmä säilyttää ja käsittelee käyttäjien kesken jaettua tietoa yhdessä paikassa (Begole, 1998). Keskitetyn arkkitehtuuriratkaisun järjestelmät muodostavat tähtimäisen verkkotopologian, jossa verkkoyhteyksiä on yhtä paljon kuin asiakkaitakin, koska kaikki tiedonkäsittelytoiminnot ja -tapahtumat kierrätetään keskitettyjen komponenttien kautta. Tähtitopologian vuoksi keskitetyn arkkitehtuurin järjestelmien tarve verkkoresursseille kasvaa lineaarisesti asiakkaiden lukumäärän suhteen.

**Keskitetyn arkkitehtuurin etuja.** Keskitetty arkkitehtuuriratkaisun etuna on yksinkertaisuus ja jaetun tiedon helppo hallinta (Begole, 1998; Roth ja Unger, 2000). Keskitettyjä komponentteja käyttävä arkkitehtuuriratkaisu on yksinkertaisin vaihtoehto toteuttaa ryhmätyöohjelma, koska järjestelmä säilyttää tietoa vain yhdessä paikassa. Keskitetysti säilytettyä tietoa myös käsitellään keskitettyjen komponenttien kautta, jolloin tiedon yhtenäisyyden hallinta (katso kohta 3.2) on yksinkertaista. Session alustus (katso alakohta 2.2.3) on myös yksinkertaisempaa, koska sessioonkin liittyvä tieto säilytetään keskitetyissä komponenteissa. Tähtimäinen verkkotopologian etuna on lineaarinen kasvu käyttäjien määrän mukaisesti, koska jokainen käyttäjä lisää uuden kaksisuuntaisen verkkoyhteyden keskitettyjen ja käyttäjää vastaavien hajautettujen komponenttien välille. Keskitetty järjestelmä-arkkitehtuuri tukee myös ohjelmistokehityksen peruseriaatetta erottamalla käyttöliittymän ja sovelluksen ytimen toisistaan.

**Keskitetyn arkkitehtuurin huonot puolet.** Keskitetystä tiedon käsittelystä voi kuitenkin koitua myös haittaa, koska kaikki vuorovaikutus jaetun tiedon ja asiakkaiden välillä kulkee palvelimen kautta (Dewan ja Choudhary, 1992; Begole, 1998; Roth ja Unger, 2000). Huonot verkkoyhteydet ja palvelintietokoneen jumiuminen voivat saada koko järjestelmän takkuilemaan. Myös vasteajat käyttäjien toiminnoille saattavat venyä pitkiksi, koska kaikki tapahtumat kulkevat edestakai-

sin asiakaskomponenteilta keskitetylle komponentille, jossa varsinaiset tuotoksen muokkaustoimenpiteet tehdään. Keskitettyjen komponenttien vuoksi koko järjestelmällä on huono vikasietoisuus, koska se on täysin riippuvainen palvelimen toiminnasta. Hajautetut komponentit eivät voi toimia, jos verkkoyhteys keskitettyihin komponentteihin katkeaa tai palvelinkoneeseen tulee toimintahäiriö. Toimintahäiriön syntymistä edesauttaa myös kuormituksen keskittyminen samalle tietokoneelle kaikkien asiakaskomponenttien ollessa yhteydessä samaan palvelimeen.

**Erilaisia keskitettyjä arkkitehtuurivaihtoehtoja.** Kuvassa 10 on kuvattu keskitettyjen arkkitehtuuriratkaisujen varaan rakennettuja järjestelmäarkkitehtuureita Rothin ja Ungerin (2000) ryhmätyövälineiden arkkitehtuuriluokittelusta. Kuvassa komponentit on kuvattu laatikoina, joiden sisään on kirjoitettu komponentin nimi. Komponenttien välinen viestintä kuvataan kaksipäisinä nuolina ja katkoviivalla piirretyt neliöt komponenttien ympärillä kuvaavat yksittäisiä tietokoneita, joihin komponentit on sijoitettu.



Kuva 10: Keskitettyjä arkkitehtuuriratkaisuja, mukaelma lähteestä (Roth ja Unger, 2000).

Kuvassa 10 C1 ja C1b ovat molemmat keskitettyjä sillä erolla, että edellinen on yhteistyön suhteen läpinäkyvä ja jälkimäinen yhteistyötietoinen. Sekä C1 että C1b



toimivat ympäristöissä, jotka tukevat X-Window:sin kaltaista ikkunointijärjestelmää. Ikkunointijärjestelmä hoitaa käyttöliittymän ja keskitetyn sovelluksen ytimen välisen kommunikoinnin. Ratkaisu on yksinkertainen, koska ikkunointijärjestelmä hoitaa sovelluksen ytimen ja käyttöliittymän välisen kommunikaation, eikä järjestelmän toteuttajan tarvitse rakentaa vastaavaa mekanismia itse. Arkkitehtuurissa C1 ikkunointijärjestelmä ei ole tietoinen useasta käyttäjästä. Se käsittelee ryhmätyösovellusta tavallisen yhden käyttäjän ohjelman tavoin, mikä saattaa johtaa konfliktitilanteisiin esimerkiksi kahden käyttäjän siirtäessä vierityspalkkia yhtä aikaa eri suuntiin (Roth ja Unger, 2000). C1b ratkaisee konfliktitilanteet laajentamalla ikkunointijärjestelmää komponentilla, joka huomioi eri käyttäjät. C1b-arkkitehtuuriratkaisua on käytetty esimerkiksi Rendezvous-sovelluskehyksessä, jota tarkastelen lähemmin alakohdassa 3.3.2.

C2 edustaa ratkaisua, jossa keskitetty sovelluksen toiminnallinen ydin ja esityskomponentti on erotettu toisistaan (Roth ja Unger, 2000). C2 ei ole riippuvainen ikkunointijärjestelmästä, joten se on mahdollista toteuttaa ympäristöissä, joissa sellaista ei ole. Perusajatuksena C2:ssa on järjestelmän tilan keskittäminen, hajautetut esityskomponentit vain välittävät tilan käyttäjille. Kohdassa 4.1 käsittelemäni yhteiskirjoitusväline PREP edustaa C2-ratkaisun varaan rakennettua järjestelmää. Hajautusarkkitehtuureissa C1, C1b ja C2 ainakin toiminnallinen ydin sijaitsee palvelinkoneella, jolloin palvelimen kuormitus on suuri ja toimintahäiriöt kohtalokkaita koko järjestelmälle (Roth ja Unger, 2000; Begole, 1998).

C3 eroaa muista keskitetyistä ratkaisuista hajauttamalla ja toistamalla järjestelmän tilan, jolloin tarvitaan keskitetty koordinaatiokomponentti, jonka tehtävänä on synkronisoida tilat ja vastata niiden yhtenäisyydestä (Roth ja Unger, 2000). C3:ssa pyritään vähentämään palvelinkoneen kuormitusta ja verkon yli siirrettävää tietoa keskittämällä koordinaatiokomponentti, joka huolehtii ainoastaan järjestelmän tilan hajautettujen toisintojen yhtenäisyydestä. C3:n keskitetty komponentti ei käsittele niin suurta tietomäärää kuin muissa keskitetyissä ratkaisuissa. Toimintaedellytyksiä parantaa myös sovelluksen ytimen sijaitseminen jokaisen käyttäjän omalla tietokoneella, jolloin järjestelmään on mahdollista toteuttaa val-

mius toimia tilapäisesti itsenäisesti ilman yhteyttä koordinaatiokomponenttiin ja muihin käyttäjiin. Paremmasta vikatietyösuudesta huolimatta myös C3 on kuitenkin riippuvainen keskitetystä palvelimesta ja siten häiriöaltis sen toimintahäiriöille ja verkko-ongelmille.

### 3.1.2 Toistettu arkkitehtuuri

Toistettu arkkitehtuurin perusajatus on järjestelmän hajauttaminen toistamalla eli sijoittamalla järjestelmän kaikki komponentit usealle rinnakkaiselle tietokoneelle (Dewan ja Choudhary, 1992). Rinnakkaisia järjestelmän toisintoja nimitetään myös järjestelmän ilmentymiksi ja versioiksi (Dewan, 1996). Toistetun arkkitehtuurin ilmentymät säilyttävät ja käsittelevät jokainen omaa kopiotaan jaetusta tiedosta sekä täsmäävät tilansa kommunikoimalla keskenään verkon välityksellä (Begole, 1998).

Koska puhtaasti toistetussa arkkitehtuuriratkaisussa jokainen toisinto tarvitsee verkkoyhteyden muihin toisintoihin, tarvitsee koko järjestelmä  $n(n-1)/2$  verkkoyhteyttä, jotta täydellisen verkon kaltainen topologia syntyy (Begole, 1998). Vaikka verkkoyhteyksien määrä kasvaa, ei siirrettävän tiedon määrä välttämättä lisäänty, koska ajoaikainen toisinto säilyttää käsiteltävää tietoa paikallisesti. Verkkoyhteyksiä pitkin siirretään lähinnä tiedon yhtenäisyyden ylläpitämiseen liittyvää tietoa (katso kohtia 3.2 ja 2.3). Tästä johtuen toistetun arkkitehtuurin verkkoresursien kuormitus ei ole merkittävästi suurempi kuin keskitetyn arkkitehtuuriratkaisun (Dewan, 1996; Roth ja Unger, 2000).

Toistetun arkkitehtuurin session käsittely on yhteisen tuotoksen ohella monimutkaisempaa kuin keskitetyssä ratkaisussa (Hofte, 1996; Schlichter, 2001). Kun toisinto liittyy esimerkiksi kirjoitussessioon, on jonkin sessiossa mukana olevan toisinnon välitettävä sille session aiemmat tapahtumat, jotta toisinto osaa saattaa kopionsa tuotoksesta ajan tasalle. Toisinnon on päivitettävä tuotoksen versiohistoria (katso alakohta 2.1.3) ja muut järjestelmän käsittelemät tiedot, kuten esimerkiksi talletetut tekstipohjaiset keskustelut (katso alakohta 2.3.1), ajan tasalle. Session

alustus on tärkeää, koska muuten yhtenäisyyden hallinta on mahdotonta, sillä jaetusta tiedosta on jo valmiiksi rinnakkaisia ja ristiriitaisia versioita.

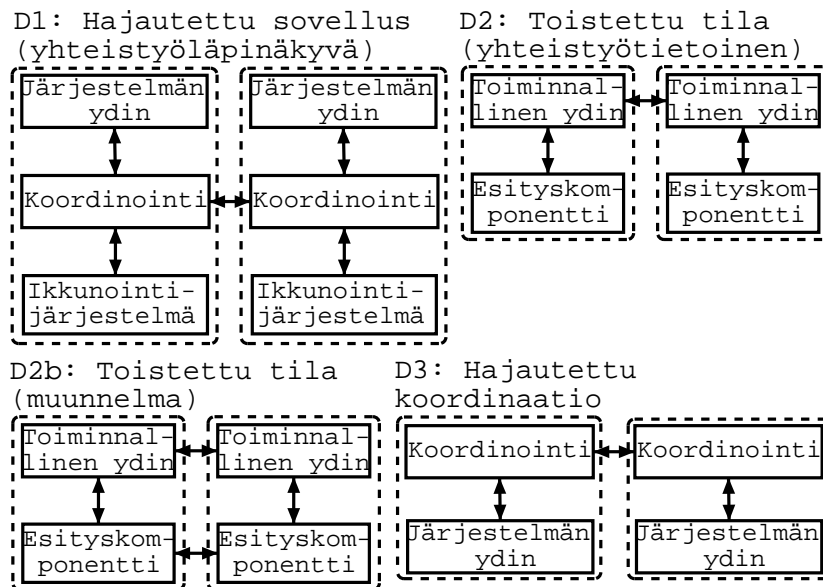
**Toistetun arkkitehtuurin etuja.** Toistetun järjestelmän vikasietoisuus on huomattavasti parempi kuin keskitetyn, koska jokainen toistettu sovellus toimii itsenäisenä kokonaisuutena (Begole, 1998; Roth ja Unger, 2000). Yhden toisinnon kaatuminen ei johda koko järjestelmän kaatumiseen, vaan toimivat sovellukset voivat jatkaa toimintaansa. Tietyn käyttäjän aikaansaamat paikalliset toiminnot toimivat toistetussa järjestelmässä nopeasti, koska kaikki järjestelmän komponentit sijaitsevat samalla tietokoneella (Dewan, 1996; Begole, 1998). Toistetussa järjestelmässä on myös helpompi toteuttaa alakohdassa 2.1.2 kuvailtuja erilaisia esitystapoja järjestelmän käsittelemälle yhteiselle tuotokselle sekä erilaisia käyttöliittymien kytkennän asteita, koska sovelluksen ytimeen kuuluva esityskomponentti ei ole keskitetty (Dewan, 1996).

**Toistetun arkkitehtuurin haittapuolia.** Kääntöpuolena paikallisen toiminnan sujuvuudelle toistetussa ratkaisussa on järjestelmän käsittelemän tuotoksen yhtenäisyyden hankala ylläpitäminen (Dewan, 1996; Begole, 1998). Yhtenäisyysongelmat ovat seurausta siitä, että järjestelmän toisinoilla on omat kopionsa tiedosta, joiden on täsmättävä monimutkaisien yhtenäisyyden hallintamekanismien avulla, esimerkiksi sarjoittamalla (3.2.2). Verkkoiviiveet saattavat aiheuttaa ajoitusongelmia kahden samanaikaisen tapahtuman näkyessä käyttäjille epäloogisessa järjestyksessä, kuten alakohdan 3.2 esimerkki havainnollistaa. Yhtenäisyyteen liittyvä epäkohta toistetussa arkkitehtuurissa on tässä alakohdassa edellä mainittu session alustuksen vaikeus.

**Erilaisia toistettuja arkkitehtuuriratkaisuja.** Roth ja Unger (2000) esittelevät neljä erilaista toistettua arkkitehtuurityyppiä, jotka on kuvattu kaaviomuodossa kuvassa 11. D1 on yhteistyön suhteen läpinäkyvä ratkaisu, jossa tavallisesta yhden käyttäjän järjestelmästä on tehty ryhmätyöohjelma lisäämällä mekanismi, joka koordinoi yksittäisten toisintojen tapahtumat. D1 on keskitetyn yhteistyön suhteen läpinäkyvän keskitetyn C1 hajautusarkkitehtuurin toistettu versio.

D2 on yhteistyötietoinen hajautusvaihtoehto, jossa toisinnot yhdenmukaistavat toimintaansa jakamalla tietoa sovelluksen toiminnallisen ytimen tilasta. D2:n etuna keskitettyyn arkkitehtuuriratkaisuihin (C1, C1b, C2) nähden on paikallisten toimintojen suorittaminen ilman verkkoliikennettä, koska toiminnallinen ydin sijaitsee paikallisella tietokoneella. Haittapuolena on monimutkaisen yhtenäisyyden ylläpitämismekanismien tarve jokaisessa järjestelmän toisinnossa.

D2b on variaatio D2:sta. D2b:ssä esityskomponentit keskustelevat keskenään esimerkiksi mahdollistaakseen tietyn käyttäjän operoiman hiiren liikkeen välittämisen usean käyttäjän näytölle. Hiiren liikkeitä ei kannata kierrättää toiminnallisen ytimen kautta, koska sen tulisi olla riippumaton käyttöliittymätapahtumista, jotka eivät johda toiminnallisen ytimen tilan muutokseen. Kehitystyökalu GroupKit (Roseman ja Greenberg, 1996), johon tutustumme alakohdassa 3.3.3, perustuu D2b-tyyppiseen arkkitehtuuriratkaisuun.



Kuva 11: Toistettuja arkkitehtuuriratkaisuja (Roth ja Unger, 2000).

Kuvassa 11 esitetty hajautusarkkitehtuuri D3 perustuu koordinaatiokomponenttien väliseen kommunikointiin ja se on hajautettu versio C3:sta (Roth ja Unger,

2000). Sovelluksen ydin on toistettuna kaikilla käyttäjillä ja koordinaatiokomponentit syöttävät tietoa tapahtumista toisilleen täsmäen sovellusytimen tilan toisnoissa. Ratkaisun etuna on C3:n keskitetyn kommunikaatiopalvelimen muodostaman pullonkaulan sivuuttaminen, toisaalta koordinaatioprotokolla muodostuu hyvin monimutkaiseksi.

### 3.1.3 Hybridiarkkitehtuuri ja muita arkkitehtuuriratkaisuja

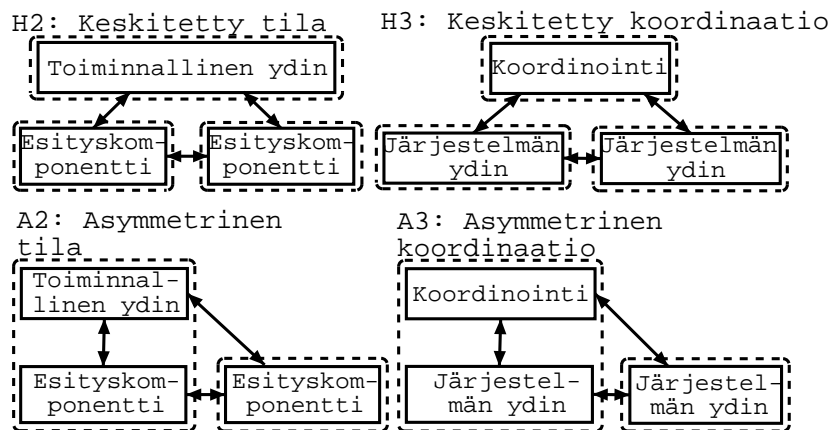
Hybridiarkkitehtuuri on yhdistelmä hajautettua ja keskitettyä arkkitehtuuria (Dewan, 1996; Begole, 1998; Roth ja Unger, 2000). Se sisältää sekä keskitettyjä että hajautettuja komponentteja pyrkien hyötymään molempien arkkitehtuurien hyödyistä välttämällä samalla haitat. Esimerkiksi tiedon yhdenmukaisuus voidaan varmistaa hybridiarkkitehtuurissa keskitetyllä palvelinkomponentilla ja tiedon talletus sekä käsittely tapahtuu hajautetuissa komponenteissa.

**Hybridiarkkitehtuurin etuja.** Verkkoliikenteen minimoiminen on eräs hybridiarkkitehtuurin merkittävimmistä saavutettavissa olevista eduista (Begole, 1998). Yhdistämällä esimerkiksi toistettujen komponenttien suora kommunikointi ja keskitetty koordinaatiokomponentti voidaan verkkoresurssien käyttö ja yksittäisten tietokoneiden kuormitus pitää minimissä suuria tietomääriä siirrettäessä. Keskitettyyn arkkitehtuurin verrattuna hybridiarkkitehtuuri on luotettavampi, koska keskitettyjä komponentteja ei kuormiteta liikaa. Hybridiarkkitehtuurin toimintavarmuus paranee hajautetusta ratkaisusta tutun tiedon hajauttamisen kautta. Toisaalta keskitettyjen komponenttien käyttö yksinkertaistaa toteutusta ja erityisesti yhtenäisyyden hallintaa toistettuihin ratkaisuihin verrattuna.

**Hybridiarkkitehtuurin haittoja.** Suunnittelijoiden, jotka aikovat käyttää hybridiarkkitehtuuria ryhmätyöohjelmassa, on punnittava tarkoin, mitkä komponentit hajautetaan ja vastaavasti toistetaan. Epäonnistuneen suunnittelun tuloksena saattaa syntyä vielä monimutkaisempi rakenne kuin hajautetussa arkkitehtuurissa, jossa yhdistyykin keskitetyn ja toistetun järjestelmän heikkoudet hyvien puolien sijasta. Hybridijärjestelmä saattaa olla suunnittelijan näkökulmasta toistettuakin

arkkitehtuuria vaikeaselkoisempi, koska se koostuu sekä hajautetuista että keskitetyistä komponenteista (Begole, 1998). Ryhmätyöohjelman suunnittelu ja toteutus vaikeutuu valmiin järjestelmän toimintavarmuuden ja nopeuden kustannuksella. Monimutkaisuudesta huolimatta nykyiset ryhmätyöohjelmat ovat tyypillisesti hybridijärjestelmiä (Roth ja Unger, 2000).

**Kaksi hybridijärjestelmäversiota.** Kuvassa 12 on kaksi Rothin ja Ungerin (2000) esittämää hybridiarkkitehtuuria. Hybridiarkkitehtuuri H2 perustuu järjestelmän tilan keskittämiseen sijoittamalla toiminnallinen ydin palvelinkoneelle, mikä painottaa keskitetyn arkkitehtuurin piirteitä. H2 pyrkii välttämään keskitetyn C2-arkkitehtuurin huonot puolet yhdistämällä siihen piirteitä toistetusta D2b-arkkitehtuurista: C2:n keskitetty tiedon käsittely yksinkertaistaa yhtenäisyyden hallintaa ja D2b:n mallin mukainen esityskomponenttien välinen kommunikointi vähentää keskitettyjen komponenttien kuormitusta. Eräs H2-tyyppistä arkkitehtuuriratkaisua käyttävistä ryhmätyösovelluksista on sovelluskehys Suite, jota käsittelem tarkemmin alakohdassa 3.3.1.



Kuva 12: Hybridiarkkitehtuureita ja asymmetrisia arkkitehtuuriratkaisuja (Roth ja Unger, 2000).

Kuvan 12 arkkitehtuuriratkaisu H3 on sekoitus keskitettyä C3:sta ja toistettua D3:sta jälkimmäistä painottaen. H3:ssa pyritään yksinkertaistamaan yhtenäisyy-

den hallintaa keskitetyllä koordinaatiokomponentilla. Toisaalta toisinnot voivat kommunikoida keskenään suoraan, mikä on hyödyllistä varsinkin jatkuvan yhteydenpidon tapauksessa, esimerkiksi reaaliaikaista ääntä siirrettäessä. Kohdassa 4.2 käsittelemäni yhteiskirjoitusväline Calliopen arkkitehtuuriratkaisu muistuttaa läheisesti H3:sta.

**Asymmetriset arkkitehtuurit.** Aiemmin tässä kohdassa esittelemäni arkkitehtuuriratkaisut koostuivat samanlaisista hajautetuista komponenteista ja mahdollisesti niistä poikkeavasta palvelimesta. Asymmetristen arkkitehtuurien tapauksessa komponentit on hajautettu eri tietokoneisiin epätasaisesti (Roth ja Unger, 2000). Tyypillisesti asymmetristen arkkitehtuurien tapauksessa yhdelle tietokoneelle on sijoitettu enemmän komponentteja kuin muille. Toinen vaihtoehto on, että kaikille tietokoneille on sijoitettu kaikki komponentit, eli järjestelmä on toistettu, mutta vain yksi toisinnosta käyttää tiettyjä komponentteja. Toisintoa, joka tarjoaa palveluja muille toisinnolle, kutsutaan omistautuneeksi palvelimeksi. Ratkaisu on tuttu nykyisistä Internetin välityksellä toimivista peleistä, joissa yksi pelaaja käynnistää peliohjelman omistautuneena palvelimena, johon muut pelaajat ottavat yhteyttä verkon kautta.

**Asymmetrisen arkkitehtuuriratkaisun etuja ja mahdollisuuksia.** Asymmetrisen ratkaisu pyrkii hyötymään keskitettyjen ja toistettujen arkkitehtuurien hyvistä puolista ja välttämään huonoja puolia hybridiarkkitehtuurien tapaan (Roth ja Unger, 2000). Eräs merkittävä mahdollisuus asymmetrisessä arkkitehtuuriratkaisussa on järjestelmän toteuttaminen toistettuna siten, että toisinnot valitsevat joukostaan omistautuneen palvelimen ajoaikaisesti. Omistautunut palvelin voi toimia samassa roolissa kuin keskitetyt palvelimet. Jos omistautunut palvelin saa jostain syystä toimintahäiriön, voivat muut toisinnot valita joukostaan uuden omistautuneen palvelimen, joka ottaa kaatuneen toisinnon paikan. Näin järjestelmän vikasietoisuus voi olla yhtä hyvä kuin toistetussa arkkitehtuuriratkaisussa, mutta omistautuneelle palvelimelle keskitetyn yhtenäisyyden hallinta on helpompaa. Joustava omistautuneen palvelimen valinta ratkaisee keskitetyn arkkitehtuuriratkaisun muodostaman pullonkaulan.

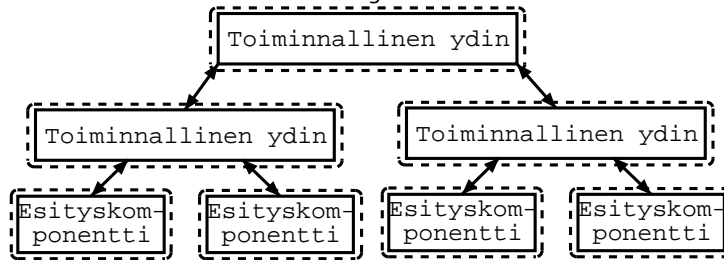
**Asymmetrisen arkkitehtuuriratkaisun huonoja puolia.** Asymmetrinen järjestelmä, jossa on joustava omistautuneen palvelimen valintamekanismi, on vielä monimutkaisempi tekninen ratkaisu kuin hybridijärjestelmä (Roth ja Unger, 2000). Asymmetrisen järjestelmän suunnittelijoiden on ratkaistava monta teknistä haastetta, esimerkiksi kuinka valita omistautunut palvelin ja mistä tietty toisinto tietää, että palvelin on kaatunut eikä toisinto itse ole verkkokatkoksen uhri.

**Kaksi asymmetristä arkkitehtuuriratkaisua.** Kuvassa 12 on kaaviot kahdesta Rothin ja Ungerin (2000) esittelemästä asymmetrisestä arkkitehtuuriratkaisusta: A2 ja A3. A2:n tapauksessa toiminnallinen ydin on sijoitettu tietylle tietokoneelle esityskomponentin lisäksi. Muilla tietokoneilla on vain esityskomponentti, joka on yhteydessä toiminnalliseen ytimeen. A2 on sovellutus hybridiarkkitehtuuriratkaisusta H2. A3 on asymmetrinen versio hybridiarkkitehtuurista H3, joka painottaa järjestelmän toistettua luonnetta. A3:ssa koordinaatiokomponentti sijaitsee omistautuneella palvelimella, josta käsin se koordinoi ryhmätyöjärjestelmän toimintaa. A3 on esimerkki ratkaisusta, jossa on mahdollista toteuttaa dynaamisesti ajonaikainen omistautuneen palvelimen valinta. Sekä A2:ssa että A3:ssa hajautetut komponentit kommunikoivat myös suoraan keskenään samaan tapaan kuin toistetuissa ja hybridiarkkitehtuuriratkaisuissa.

**Usean palvelimen arkkitehtuuriratkaisu.** Viimeinen Rothin ja Ungerin (2000) esittelemistä arkkitehtuuriratkaisuista, jota tarkastelemme, on kuvassa 13 oleva usean palvelimen käyttöön perustuva arkkitehtuuriratkaisu M2. Perusajatuksena M2:ssa on suorituksen ja laskennan hajauttaminen käyttämällä useita palvelimia, joissa on järjestelmän toiminnallinen ydin. M2:n tapauksessa toiminnallinen ydin on hajautettu kahteen tasoon, joista ylempi hoitaa koko järjestelmän toimintoja korkealla tasolla ja alempi puolestaan huolehtii esityskomponenttien tarpeista. Hajauttamalla toiminnallisen ytimen toiminta pyritään parantamaan tehokkuutta. Kääntöpuolena tehokkuuden lisäykseen on monimutkaisten algoritmien tarve laskennan hajauttamisessa ja mahdollisessa kuormituksen tasapainottamisessa.



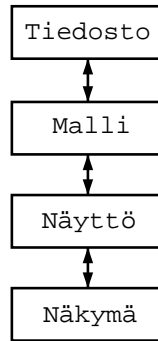
M2: Hierarkkisesti hajautettu tila



Kuva 13: Useaan palvelimeen perustuva arkkitehtuuriratkaisu (Roth ja Unger, 2000).

### 3.1.4 Vetoketjumalli

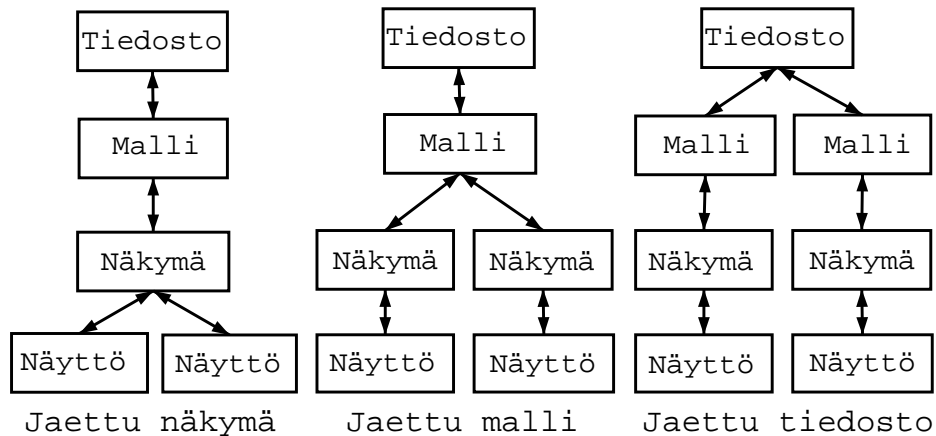
Pattersonin (1995) ryhmätyöohjelmien arkkitehtuurien kuvaustapa, jota myöhemmin alettiin kutsua vetoketjumalliksi (Hofte, 1998; Roth ja Unger, 2000), perustuu kahteen lähtökohtaan. Ryhmätyöohjelman perushaaste on yhtenäisyyden säilyttäminen sen kaikissa ilmentymissä ja ryhmätyöohjelmat voidaan jakaa neljään tasoon, jotka kuvaavat järjestelmän tilaa. Näyttö (display) kuvaa sen tiedon tilaa, joka on näkyvissä käyttäjälle tietokoneen monitorin välityksellä. Näkymä (view) on tila, joka yhdistää näytön ja järjestelmän käsittelemän tiedon, jota kuvaa malliksi (model) nimetty tila. Tiedosto (file) kuvaa järjestelmän käsittelemän tiedon pysyvää tilaa, pitkäaikaistalletusta. Nimitys vetoketjumalli juontaa juurensa mielikuvaan tiloista vetoketjuna, joka on mahdollista avata, jolloin syntyy rinnakkaisia tiloja. Kuvassa 14 näkyy vetoketjumallin tasot yhden käyttäjän järjestelmässä, jota ei ole hajautettu. Kuvassa laatikot kuvaavat tasoja ja näiden väliset nuolet tasojen välistä kommunikaatiota ja riippuvuutta.



Kuva 14: Sovelluksen tilan tasot (Patterson, 1995).

Edellä kuvatuista lähtökohdasta arkkitehtuuriratkaisut voidaan Pattersonin (1995) mukaan jakaa kolmeen kategoriaan: jaetun tilan arkkitehtuureihin, synkronisoidun tilan arkkitehtuureihin sekä näiden sekoitukseen, hybridiarkkitehtuuriin. Jaetun tilan arkkitehtuurit vastaavat alakohdassa 3.1.1 kuvailtuja keskitettyjä arkkitehtuureita ja synkronisoidun tilan arkkitehtuurit alakohdassa 3.1.2 käsiteltyjä toistettuja arkkitehtuuriratkaisuja. Vetoketjumallin hybridiarkkitehtuurit vastaavat puolestaan alakohdassa 3.1.3 esiteltyjä arkkitehtuuriratkaisuja.

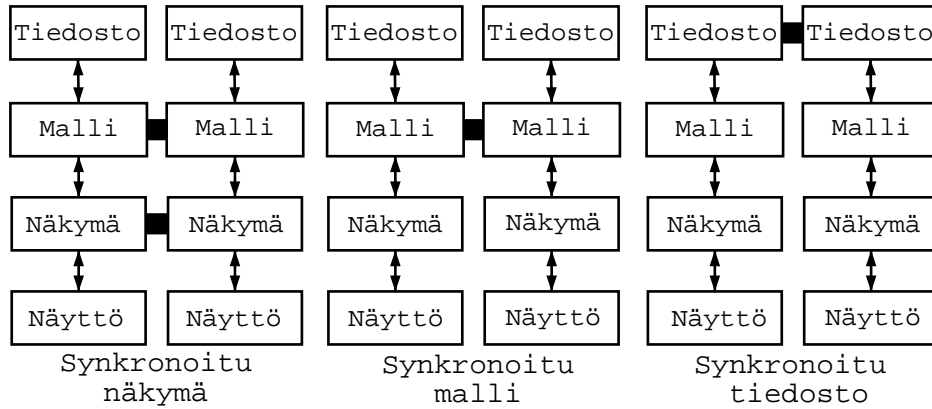
Kuvassa 15 on kolme erilaista arkkitehtuuriratkaisua, jotka näyttävät, mitä vaihtoehtoja tilan jakamiseksi kaikkien järjestelmän käyttäjien kesken tiedon yhdenmukaisuuden takaamiseksi on. Kuvan laatikot kuvaavat tasoja ja näiden väliset nuolet tasojen välistä kommunikaatiota ja riippuvuutta. Vetoketjuarkkitehtuuri käyttää abstraktiona kaikille käyttäjille keskitetyn komponentin sijasta kaikkien käyttäjien kesken jaettua tiedon tilaa tietyssä järjestelmän tasossa.



Kuva 15: Yhtenäisyyden takaaminen tilan jakamisella (Patterson, 1995).

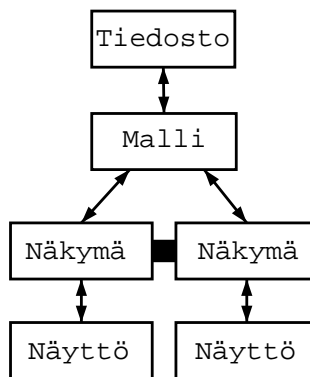
Kuvassa 15 kuvatut ratkaisut edustavat kaikki keskitettyä arkkitehtuuria edellisen alakohdan 3.1.1 terminologiaa käyttääksemme. Kuvassa rinnakkaiset tilaa kuvaavat laatikot merkitsevät eri käyttäjien eri koneilla sijaitsevia järjestelmän osia, jotka ovat yhteydessä keskitetyllä palvelimella sijaitseviin keskitettyihin komponentteihin.

Tilan jakamisen sijasta arkkitehtuuriratkaisu voi perustua kahden tai useamman järjestelmän toisinnon tilojen synkronoimiseen (Patterson, 1995). Tällöin jokaisella käyttäjällä on oma, paikallinen kopio järjestelmän tilasta. Järjestelmän tilojen yhdenmukaisuus edellyttää tässä tapauksessa jonkinlaista mekanismia tai protokollaa, joka huolehtii tilojen yhdenmukaisuudesta. Vetoketjumalli ei ota kantaa varsinaiseen tilojen yhtenäisyyden takaavan mekanismin toiminnasta mutta kuvaa kolme erilaista tapaa synkronoida tilat. Kuvassa 16 synkronointiside on kuvattu paksuna horisontaalisena järjestelmän toisintojen välisenä viivana. Synkronoidut näkymät edellyttävät myös mallin synkronoimista, koska näkymä heijastaa mallin tilaa.



Kuva 16: Yhtenäisyyden takaaminen tilan synkronoisella (Patterson, 1995).

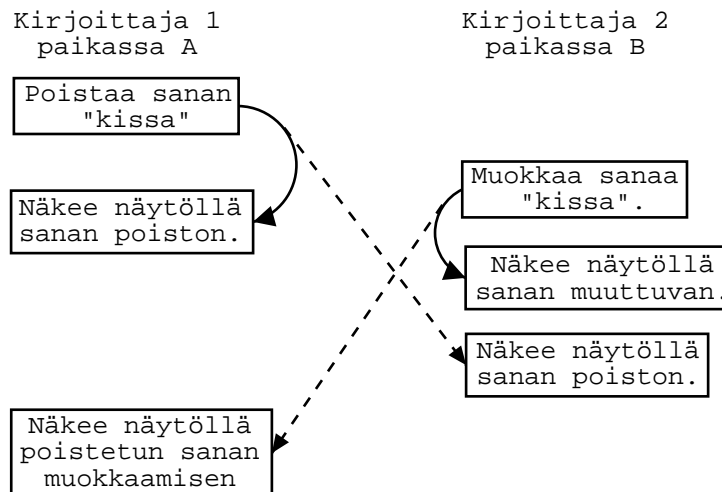
Alakohdan 3.1.2 terminologian mukaisesti tilan synkronoimisen kautta syntyviä ratkaisuja nimitetään toistetuiksi järjestelmiksi. Toistettu järjestelmä toimii ilman keskitettyä palvelinohjelmaa ja siinä jokainen järjestelmän toisinto hoitaa myös palvelimen tehtäviä. Hybridiarkkitehtuuri, jossa sovelletaan sekä tilan jakamista että yhtenäisyyden ylläpitämistä tilojen synkronoinnin kautta, näyttää vetoketju-mallilla kuvattuna kuvan 17 kaltaiselta. Kuvan hybridiarkkitehtuuri vastaa alakohdassa käsiteltyä 3.1.3 arkkitehtuuriratkaisua H3.



Kuva 17: Jaettuun malliin ja synkronoituihin näkymiin perustuva hybridi (Patterson, 1995).

## 3.2 Yhtenäisyyden hallinta

Yhteiskirjoitusprosessissa maantieteellisesti hajallaan olevien kirjoittajien tekemät muutokset dokumenttiin eivät välttämättä saavu oikeassa järjestyksessä verkoviiven vuoksi. Grudin (1994) käyttää esimerkkinä tilannetta, jossa kaksi kirjoittajaa muokkaavat samaa lausetta samanaikaisesti. Kuvitellaan tilanne, jossa kirjoittajat 1 ja 2 muokkaavat samaa lausetta kahdessa maantieteellisesti erillään sijaitsevassa paikassa. Järjestelmässä paikallisten muokkaustoimien palaute on vakioaikaista, mutta muiden käyttäjien tekemien muutoksien näkyvyys riippuu verkon toiminnasta. Ongelmatilanne syntyy, kun kirjoittaja 1 päättää poistaa sanan ja kirjoittaja 2 ei saa poistosta ajoissa tietoa vaan muokkaa sanaa, joka on jo oikeastaan poistettu. Jos verkkoliikenne takeltelee, niin kumpikaan kirjoittajista 1 ja 2 eivät pahimmassa tapauksessa saa palautetta toistensa toimista ennen kuin ne on tehty. Kuva 18 havainnollistaa edellä mainittua tapausta. Kuvassa yhtenäiset viivat merkitsevät suoraa vuorovaikutusta ja katkoviivat epäsuoraa vuorovaikutusta (vertaa kuvaan 1 luvussa 2.1). Kuvassa aika etenee ylhäältä alas.



Kuva 18: Esimerkkiongelman yhtenäisyyden ylläpitämisessä, mukaelma lähteestä Grudin (1994).

Yllä kuvatun tapauksen kaltainen tilanne on käyttäjän kannalta hyvin hämmentävä, koska tietokoneohjelmien käyttäjät ovat tottuneet siihen, että ohjelma suorittaa sen toiminnon, jonka käyttäjä sen kääsee tekemään. Kun dokumenttiin vaikuttaa ryhmätyöohjelmassa muutkin tahot kuin yksi tietty käyttäjä, saattaa näytöllä inhimillisestä näkökulmasta katsottuna tapahtua kaikenlaista käsittämätöntä. Jos yhteiskirjoitusjärjestelmä saa dokumentin muokkauskomennot väärässä järjestyksessä, saattaa dokumentti korruptoitua. Tietoisuus (katso kohtaa 2.4) on eräs keino hälventää käyttäjien epätietoisuutta ja vähentää sekaannusta, mutta ryhmätyöohjelman järkevän toiminnan kannalta käyttäjien ja tuotoksen välisen vuorovaikutuksen perustana on jonkinlainen yhtenäisyyden hallintamekanismi, jonka tehtävänä on taata kaikille yhtenäinen esitys tiedosta.

Dourish (1994) luettelee kolme asiaa, joihin yhtenäisyyden hallintamekanismin on otettava kantaa. Aluksi on päätettävä, minkä kokoisten objektien suhteen yhtenäisyyttä pyritään hallitsemaan. Hofte (1996) luettelee yhteiskirjoittamisvälineille relevantteja objekteja: dokumentti, luku, kappale, kirjain, tarkemmin määrittelmätön mielivaltainen yksikkö tai tapahtuma. Seuraavaksi on päätettävä, missä tilanteessa objektit ovat epäyhtenäisessä tilassa. Jotta yhtenäisyyttä voidaan ylläpitää, on ensin määriteltävä, milloin yhteinen tuotos on yhtenäinen ja missä tilanteessa se katsotaan epäyhtenäiseksi. Ryhmäeditorin tapauksessa voidaan esimerkiksi määritellä, että yhteinen tuotos on yhtenäinen, jos tietyt käyttäjät näkevät saman tekstin sanan tarkkuudella, eli tietyn käyttäjän näkymä dokumentista voi poiketa muista hetkellisesti siten, ettei siinä näy jonkun muun käyttäjän syöttämää sanaa. Lopuksi yhtenäisyyden hallintamekanismin on kyettävä synkronoimaan epäyhtenäinen tuotos eli saattamaan epäyhtenäinen tieto yhtenäiseksi tavalla tai toisella.

Yhtenäisyyden määrittelemine ei ole välttämättä helppoa kaikissa tilanteissa (Dourish, 1994; Hofte, 1996): Onko esimerkiksi ryhmän työstämä dokumentti epäyhtenäinen, kun eri käyttäjät näkevät erilaisen kuvauksen yhteisestä dokumentista? Jos ryhmä työskentelee WYSIWIS-käyttöliittymän (2.1.2) kautta, niin silloin vastaus kysymykseen on kyllä. Toisaalta, jos joku ryhmän jäsenistä rentouttaa

käyttöliittymänsä kytkentää vaikkapa ajan suhteen muihin käyttäjiin nähden, niin silloin kysymyksessä ei välttämättä ole epäyhtenäisyys. Epäyhtenäisyyden kriteereitä määriteltessä on huomioitava sekä jaetun tiedon tila että esitys (Hofte, 1996). Käyttöliittymän kytkennän eri asteissa yhtenäisyyskriteerit saattavat vaihdella.

Yksinkertaisin kuviteltavissa oleva mekanismi, joka sallii usean käyttäjän muokata samaa tuotosta yhtenäisyyttä ylläpitäen, on vuoron perään toimiminen. Silloin yksi käyttäjä muokkaa dokumenttia kerrallaan luovuttaen vuoronsa toiselle käyttäjälle tehtyään haluamansa muutokset. Vuoron perään toimimiseen perustuva yhteiskirjoitusjärjestelmä ei kuitenkaan ole juurikaan tavallista sähköpostin vaihtoa parempi vaihtoehto. *Lukitsemiseksi* kutsuttu tekniikka on sovellus vuoron perään toimimisesta (Greenberg ja Marwood, 1994; Hofte, 1996). Siinä tietyn kokoinen osa dokumenttia lukitaan muilta käyttäjiltä sillä välin, kun tietty käyttäjä muokkaa kyseistä osaa. Lukitus voidaan vapauttaa, kun käyttäjä ei enää muokkaa häntä varten lukittua kohtaa. Käsittelen alakohdassa 3.2.1 erilaisia tapoja toteuttaa lukitusmekanismi yhteiskirjoitusvälineessä.

Lukitsemisesta poikkeava lähestymistapa yhtenäisyyden hallintaan on *Sarjoittaminen* (Greenberg ja Marwood, 1994; Hofte, 1996; Begole, 1998). Sarjoittamisen perusajatuksena on, että käyttäjien aiheuttamat tapahtumat suoritetaan peräjälkeen, sarjassa, jolloin yhteisen dokumentin tila pysyy kaikille käyttäjille samana. Käyttäjän näkökulmasta sarjoittaminen mahdollistaa samanaikaisen työskentelyn missä tahansa dokumentin osassa toisin kuin aiemmin mainittu lukitseminen. Sarjoittamismekanismiin haasteena on ratkaista tilanteet, joissa kaksi käyttäjää kohdistavat molemmat toisistaan poikkeavan toimenpiteen dokumentin samaan kohtaan lähes samanaikaisesti. Mekanismin on päätettävä kumman käyttäjän toimenpide toteutetaan ja kuinka toteuttamatta jääneeseen toimenpiteeseen suhtaudutaan. Joissakin tilanteissa sarjoittamismekanismiin on jopa muutettava jo tehtyjä toimenpiteitä korjatakseen eri syistä tapahtuneet virhetilanteet. Tutustumme sarjoittamiseen tarkemmin alakohdassa 3.2.2.

### 3.2.1 Lukitseminen

Lukitseminen on yksinkertaisin tässä kohdassa esiteltävistä yhtenäisyyden hallintamenetelmistä. Lukitsemisessa muokkauksen kohteena oleva dokumentin osa lukitaan siksi aikaa, kun sitä muokataan (Greenberg ja Marwood, 1994; Hofte, 1996) Lukitseminen tapahtuu siten, että käyttäjän halutessa muokata tiettyä dokumentin kohtaa järjestelmä antaa luvan muokata kohtaa vasta tarkistettuaan, ettei se ole lukittu. Jos kohtaa ei ole lukittu, lukitsee järjestelmä sen käyttäjää varten. Niin kauan kuin kohta on lukittuna, muut käyttäjät eivät voi lukita sitä itselleen. Kohta pysyy lukittuna, kunnes käyttäjä ei enää muokkaa sitä, jolloin lukitus poistetaan. Näin yhteiskirjoitusjärjestelmä voi varmistua siitä, että ainoastaan yksi käyttäjä muokkaa tiettyä dokumentin osaa eikä yhtenäisyyden rikkovia konfliktitilanteita pääse syntymään.

Hofte (1996) erittelee kolme pääpiirrettä, joiden mukaan lukitseminen voidaan toteuttaa. Ryhmätyöohjelma voi joko hoitaa lukitsemisen automaattisesti, tai antaa käyttäjän lukita ja vapauttaa lukitus. Lukittavan objektin koko on toinen pääpiirre, yhteiskirjoittamisen tapauksessa lukittava objekti on osa yhteistä dokumenttia. Kolmas pääpiirre on lukitusmenettelytapa, jonka käsittelyyn omistan tämän alakohdan loppuosan. Ryhmätyöohjelma voi myös antaa käyttäjilleen vallan muuttella näitä piirteitä, jolloin niistä voidaan puhua parametreinä. Esimerkiksi kohdassa 4.2 käsittelemäni yhteiskirjoitusväline Calliope (Mitchell, 1996) antaa käyttäjän valita neljän lukitsemistason välillä, jotka ovat: sana, rivi, kappale ja vapaasti valittu alue yhteisestä dokumentista.

Greenberg ja Marwood (1994) jakaa lukituksen kolmeen luokkaan sen mukaan, jääkö järjestelmä odottamaan lukituspäätöstä ja saako varmistamattoman lukituksen vapauttaa ennen järjestelmän vastausta. *Ei-optimistinen lukitus* toimii siten, että järjestelmä ei anna käyttäjän muokata lukittavaa kohtaa ennen kuin kohta on lukittu. Tämä saattaa joissakin tapauksissa merkitä sitä, että käyttäjä joutuu odottamaan tapauksen käsittelyä.



*Optimistinen lukitus* toimii pääpiirteittäin kuten optimistinen sarjoittaminen, eli tapahtumien oletetaan tapahtuvan oikeassa järjestyksessä ja väärässä järjestyksessä saapuvat tapahtumat korjataan oikeille kohdilleen (Greenberg ja Marwood, 1994). Optimistinen lukitus toimii siten, että käyttäjälle annetaan alustava lukitus välittömästi tämän pyydettyä lukitusta johonkin dokumentin osaan. Jos osa on jo lukittu, kumoutuu kaikki käyttäjän alustavan lukituksen aikana tekemät muutokset. Muussa tapauksessa järjestelmä lukitsee kohdan.

*Täysin optimistinen lukitus* antaa käyttäjän muokata alustavasti lukittua kohtaa ja vapauttaa muokatun kohdan ennen kuin järjestelmä vahvistaa lukituksen (Greenberg ja Marwood, 1994). Täysin optimistinen lukitus antaa käyttäjälle huomattavasti vapautta mutta saattaa samalla aiheuttaa tarvetta hyvin monimutkaisille korjaustoimenpiteille. *Puolioptimistinen lukitus* on kompromissiratkaisu optimistisen ja ei-optimistisen lukituksen välillä. Siinä käytetään myös alustavaa lukitusta, mutta alustavasti lukittua dokumentin osaa ei voi vapauttaa ennen kuin järjestelmä vahvistaa lukituksen. Tällä tavoin vältetään täysin optimistisen lukituksen mahdollisesti aiheuttamat monimutkaiset korjaustoimenpiteet. Taulukko 13 koostaa lukituksessa käytettävät optimismitasot. Mitä optimistisempi järjestelmä on, sitä enemmän sen tulee muistaa tapahtumista pystyäkseen havaitsemaan ja sijoittamaan oikein väärässä järjestyksessä saapuvat tapahtumat.

Taulukko 13: Optimismitasot lukituksessa (Greenberg ja Marwood, 1994).

<b>Optimismitaso</b>	<b>Onko kohde muokattavissa lukitusta odotettaessa?</b>	<b>Voiko muutetun kohteen vapauttaa lukitusta odotettaessa?</b>
Ei-optimistinen	ei	ei
Puolioptimistinen	kyllä	ei
Täysin optimistinen	kyllä	kyllä

### 3.2.2 Sarjoittaminen

*Sarjoittaminen* on yhtenäisyyden hallintamekanismi, jossa tapahtumat joko pakotetaan suoritettaviksi peräkkäin tai jossa väärässä järjestyksessä saapuvat tapahtumat sijoitetaan oikeille paikoilleen korjaustoimenpiteillä (Greenberg ja Marwood, 1994). Sarjoittamisessa saatetaan myös viivyttää tapahtumia niiden oikean suoritusjärjestyksen takaamiseksi (Hofte, 1996). Sarjoittaminen antaa kaikille käyttäjille sellaisen vaikutelman, että tapahtumat tapahtuvat peräjälkeen, sarjassa. Sarjoitusalgoritmit perustuvat useimmiten tapahtumille annettaviin aikaleimoihin, joiden perusteella voidaan päätellä tapahtumien keskinäinen järjestys.

Jos ryhmätyöohjelmisto varmistaa, että tapahtumat suoritetaan oikeassa järjestyksessä, on kyseessä *Ei-optimistinen sarjoittaminen* (ei-optimistisen synonyyminä käytetään myös pessimististä ja konservatiivista sarjoittamista) (Greenberg ja Marwood, 1994). Ei-optimistinen sarjoittaminen perustuu siihen, että tapahtumia ei suoriteta ennen kuin aiemmat tapahtumat on suoritettu ja palaute suorittamisesta on saapunut kaikille käyttäjille. Menetelmän ongelmana on muokkausoperaatioiden lisääntynyt aikavaativuus, koska vaikka kaksi käyttäjää muokkaisivatkin eri sanoja, käsittelee yhteiskirjoitusväline vain yhtä tapahtumaa kerrallaan ja toisen käyttäjän tapahtuma joutuu odottamaan.

*Optimistisessa sarjoittamisessa* tehdään optimistinen olettaus, että tapahtumat saapuvat yleensä oikeassa järjestyksessä Greenberg ja Marwood (1994). Jos tapahtumat kuitenkin sattuvat tulemaan väärässä järjestyksessä, on ryhmätyöohjelman havaittava epäkohta ja puututtava tilanteeseen korjaamalla jo tapahtunut dokumentin yhtenäisyyden rikkoutuminen. Optimistinen sarjoittaminen on nopeampi käyttäjän kannalta kuin ei-optimistinen sarjoittaminen, mutta sen toteuttaminen voi olla hyvin vaikeaa. Optimistisen sarjoittamisen synonyyminä käytetään myös aggressiivista sarjoittamista.

### 3.2.3 Operationaalinen transformaatio

*Operationaalinen transformaatio* on optimistiseen sarjoittamiseen perustuva yhtenäisyyden hallintamekanismi, joka sopii erityisesti toistettuihin järjestelmiin (katso alakohta 3.1.2) reaaliaikaisen työskentelyn yhteyteen (Sun ja Ellis, 1998; Begole, 1998). Siinä kaikki tapahtumat, jotka johtavat yhteisen tuotoksen tilan muuttumiseen, kuvataan atomisina operaatioina, joihin liitetään koko järjestelmälle yhtenäinen aikaleima. Atomiset operaatiot suoritetaan ensin paikalliseen yhteisen tuotoksen kopioon ja lähetetään sitten tiedoksi muille hajautetuille toisinoille. Jokainen toisinto sijoittaa muilta toisinoilta saamansa atomiset operaatiot paikallisten atomisten operaatioiden lomaan aikaleiman perusteella. Tietty toisinto saattaa saada operaatioita muilta toisinoilta epäjärjestyksessä, jolloin sen on muutettava omaa esitystään yhteisestä tuotoksesta perumalla liian aikaisin suorittamansa atomiset operaatiot ja suorittamalla sen jälkeen operaatiot oikeassa järjestyksessä.

Sun ja Ellis (1998) määrittelevät yhtenäisyyden uudelleen operationaalisen transformaation yhteyteen paremmin sopivaksi. Heidän mukaansa yhteiskirjoitusväline on yhtenäinen, jos se aina noudattaa kolmea sääntöä:

1. **Yhtenäisyys:** kun kaikki atomiset operaatiot on suoritettu kaikille toisinoille, silloin kaikki kopiot dokumentista ovat identtiset.
2. **Kausaliteetin säilyttäminen:** kaikille atomisille operaatioille O1 ja O2 pätee, että jos O1 on tapahtunut ennen O2:sta, niin se suoritetaan kaikissa toisinoissa ennen O2:sta.
3. **Aikomuksen säilyttäminen:** kaikille atomisille operaatioille O pätee, että niiden suorittamisen seuraukset ovat kaikissa toisinoissa samat ja tietyn operaation suorittaminen ei vaikuta muiden atomisten operaatioiden suorittamisen seurauksia.

Edellä olevan yhtenäisyyden määritelmän pohjalta on mahdollista toteuttaa opera-

tionaalisen transformaation ajatusta toteuttava yhtenäisyysmekanismi. Operationaalisen transformaation merkittävä etu on viiveetön paikallinen toiminta (Begole, 1998; Sun ja Ellis, 1998). Mekanismi sopii myös hyvin toistettuun järjestelmään, koska se ei ole mitenkään riippuvainen muista toisoinnoista. Jos esimerkiksi jokin toisoinnoista kaatuu, ei siitä seuraa muille minkäänlaista haittaa. Lisäksi merkittävä operationaalisen transformaation etu on sen riippumattomuus verkkoviiveestä. Tapahtumat, jotka tulevat verkkoviiveen vuoksi myöhässä on mahdollista suorittaa perumalla aiemmin suoritettuja operaatioita ja tekemällä kaikki operaatiot sen jälkeen oikeassa järjestyksessä.

### 3.3 Ryhmätyöohjelmien kehitystyökalut

Ryhmätyöohjelmat ovat vaativampia kehittää kuin tavalliset yhden käyttäjän ohjelmat. Ryhmätyöohjelmien kehittäminen on toteutettava piirteitä, joita yhden käyttäjän järjestelmissä ei välttämättä tarvita, kuten yhtenäisyyden hallinta (katso kohta 3.2), järjestelmän käsittelemän tiedon ja käyttöliittymän erottaminen (katso kohta 3.1) ja käyttäjien roolien tukeminen (katso alakohta 2.2.1) (Patterson *et al.*, 1990). Kehitystyökalut ovat komponenttikokoelmia ja sovelluskehysjä, joiden tarkoituksena on helpottaa ryhmätyöohjelmien kehittämistä. Kehitystyökalujen perusajatuksena on tehdä ryhmätyöohjelman kehittämisestä yhtä helppoa kuin yhdelle käyttäjälle suunnatun järjestelmän kehittämisestä. Greenberg ja Roseman (1999) ja Dewan ja Choudhary (1992) nimeävät kehitystyökalujen pääkomponenteiksi hajautusarkkitehtuurin, kommunikaatioabstraktiot, session hallinnan ja usean käyttäjän käyttöliittymäkomponentit.

**Hajautusarkkitehtuuri.** Eräs merkittävimmistä eroista usean käyttäjän ja yhden käyttäjän ohjelmistoissa on hajautettu järjestelmäarkkitehtuuri (Dewan, 1996; Greenberg ja Roseman, 1999). Ryhmätyöohjelmien kehittäminen helpottuu oleellisesti, jos järjestelmän kehittäjä ei tarvitse suunnitella ja toteuttaa hajautusarkkitehtuuria, vaan hän voi käyttää kehitystyökalujen valmiita ratkaisuja. Hajautusarkkitehtuurin tehtävänä on automaattisesti hallita käyttäjien prosesseja ja näiden vä-

listä kommunikaatiota (Greenberg ja Roseman, 1999). Arkkitehtuurin tärkeänä tehtävänä on muun muassa yhtenäisyyden hallinta. Arkkitehtuuri ei kuitenkaan voi ennakoida kaikkea sitä, mitä sovelluskehittäjä sillä haluaa tehdä. Siksi on tärkeää, että perusarkkitehtuuri on laajennettavissa erilaisin mekanismien ja rajapintojen kautta.

**Ohjelmointiabstraktiot ja rajapinnat.** Ohjelmointiabstraktiot ja erilaiset rajapinnat ovat väline ryhmätyöjärjestelmän kehittäjille välittää sovelluskohtaista tietoa järjestelmässä (Greenberg ja Roseman, 1999). Tämä tieto voi liittyä esimerkiksi käyttäjien käyttöliittymien ylläpitoon, käyttäjien synnyttämiin tapahtumiin tai järjestelmän käsittelemän tiedon päivittämiseen. Eräs ohjelmointiabstraktio, jonka kehitystyökalu voi tarjota ryhmätyöohjelman toteuttajalle, on tavallisten aliohjelmakutsujen lisäksi kaikille hajautetuille komponenteille välitettävä versio aliohjelmakutsusta. Toinen kehitystyökalujen mahdollisesti tarjoama ohjelmointirajapinta on tapahtumien hallintaan liittyvät tapahtumien välitys- ja seurantapalvelut. Ryhmätyöohjelman kehitystyökalu voi esimerkiksi lähettää automaattisesti tietoa sessiohallintatapahtumista kaikille hajautetuille komponenteille, jolloin nämä voivat saattaa tilansa ajan tasalle. Kaikille hajautetuille komponenteille välitettävät aliohjelmakutsut sekä tapahtumien seuranta ja välitys liittyvät ryhmätyöohjelman koordinoimiseen. Näiden lisäksi kehitystyökalu voi tarjota jonkinlaisen abstraktiorajapinnan järjestelmän käsittelemän tiedon ja käyttöliittymän erottamiseksi ja ylläpitämiseksi. Käyttöliittymän ja järjestelmän käsittelemän tiedon erottaminen on tärkeää siksi, että samasta tiedosta voi olla useita erilaisia esitysmuotoja eri käyttäjillä tiukan WYSIWIS-mallin sijasta (katso alakohta 2.1.2).

**Usean käyttäjän käyttöliittymäkomponentit.** Usean käyttäjän käyttöliittymäkomponentit ovat käyttöliittymän osia, joissa on huomioitu monta käyttäjää yhden sijasta (Greenberg ja Roseman, 1999). Esimerkiksi alakohdassa 2.4.2 käsittelemäni tietoisuusinformaatiota välittävät komponentit ovat tällaisia. Koska myös sovelluskohtaisia tietoisuuskomponentteja saatetaan tarvita, on kehitystyökalujen hyvä tarjota myös sellaisten kehittämiseen tarkoitettuja perustietoisuuskomponentteja.

**Sessionhallintakomponentit.** Jotta ryhmä yhteistyöhaluisia ihmisiä voi käyttää hajautettua järjestelmää, tarvitaan sessionhallintaa (katso alakohta 2.2.3). Sessionhallintakomponentit hoitavat sessionhallinnan perustehtäviä, esimerkiksi session käynnistämistä, siihen liittymistä sekä siitä poistumista (Greenberg ja Roseman, 1998, 1999). Myös käyttäjien lisäys ja heidän oikeuksiensa hallinta kuuluu sessionhallintakomponenttien velvollisuuksiin. Jotkut kehitystyökalut jopa tarjoavat käyttäjilleen valmiin käyttöliittymän sisältämäänsä sessionhallintamekanismin (Greenberg ja Marwood, 1994; Begole, 1998).

Seuraavaksi tutustumme lähemmin kolmeen sovelluskehikseen: Suite:een (Dewan ja Choudhary, 1992), Rendezvous:iin (Patterson *et al.*, 1990) ja Group-Kit:iin (Roseman ja Greenberg, 1996).

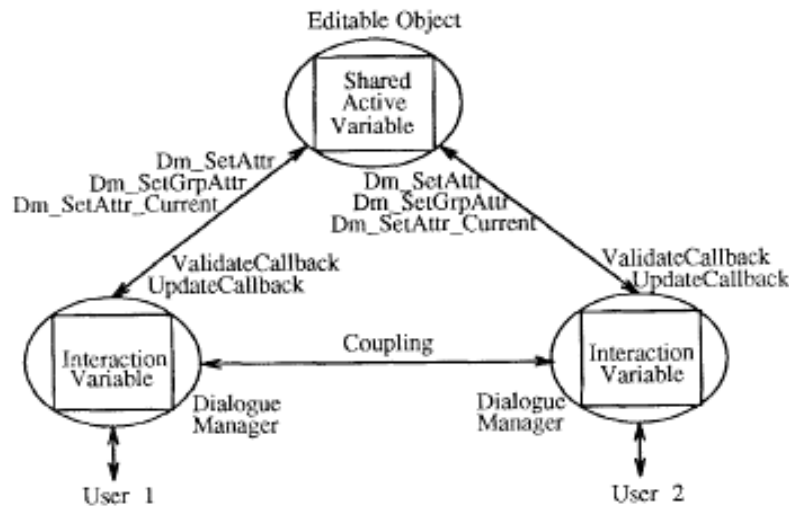
### 3.3.1 Suite

Suite koostuu hybridiarkkitehtuuriratkaisun varaan rakennetusta sovelluskehiksestä, jonka toiminnallisuutta ryhmätyöohjelman toteuttaja voi muokata haluamukseen käyttämällä erilaisia ohjelmointiabstraktioita, joiden toimintaa säädetään C-kielisillä rajapintafunktioilla (Dewan ja Choudhary, 1992). Suite toimii Unix-ympäristöissä.

**Hajautusarkkitehtuuri.** Suite edustaa alakohdassa 3.1.3 esiteltyä hybridiarkkitehtuuria H2 (Roth ja Unger, 2000). Suiten arkkitehtuuriratkaisussa keskitetty kaikkien käyttäjien kesken jaettu komponentti on nimetty muokattavaksi objektiksi (editable object kuvassa 19). Hajautetut käyttäjäkohtaiset komponentit ovat yhteydessä sekä muokattavaan objektiin että toisiinsa (dialogue manager kuvassa 19).

**Ohjelmointiabstraktiot ja rajapinnat.** Suite käyttää ohjelmointiabstraktioina jaettua aktiivista muuttujaa ja vuorovaikutusmuuttujaa (Dewan ja Choudhary, 1992). Aktiivinen muuttuja (active variable kuvassa 19) on kaikkien käyttäjien kesken jaettu esitys Suitella tehdyn ryhmätyöohjelman käsittelemästä tiedosta.

Käyttäjät muokkaavat ja ovat muuten vuorovaikutuksessa aktiivisen muuttujan kanssa vuorovaikutusmuuttujan (interaction variable kuvassa 19) kautta.



Kuva 19: Suiten arkkitehtuuriratkaisu, ohjelmointiabstraktiot ja rajapinnat (Dewan ja Choudhary, 1992).

Suiten vuorovaikutusmuuttujat ovat yhteydessä jaettuun aktiiviseen muuttujaan rajapintakutsujen kautta, joilla ne muuttavat aktiivisen muuttujan tilaa (Dewan ja Choudhary, 1992). Aktiivinen tila voi säilyttää käyttäjäkohtaista ja ryhmäkohtaista tietoa. Vuorovaikutusparametrit kytkeytyvät aktiiviseen muuttujaan erityisten rajapintafunktioiden avulla (ValidateCallback ja UpdateCallback kuvassa 19). Varsinainen käyttöliittymä ja tiedon esitys rakennetaan Suiten avulla muokkaamalla vuorovaikutusmuuttujan parametreja, joilla säädetään esimerkiksi käyttäjälle näkyvän ikkunan kokoa ja siinä näkyvän tiedon muotoseikkoja.

**Usean käyttäjän käyttöliittymäkomponentit.** Suitessa ei ole varsinaisia valmiita usean käyttäjän käyttöliittymäkomponentteja. Sen sijaan sillä voi rakentaa eri tavoin toisiinsa kytkettyjä käyttöliittymiä edellä mainitun vuorovaikutusmuuttuja-abstraktion parametreja muuttamalla (Dewan ja Choudhary, 1992). Suiten avulla rakennetut käyttöliittymät on mahdollista kytkeä tiukan WYSIWIS-periaatteen

mukaisesti (katso alakohta 2.1.2). Suite tukee myös eriasteista WYSIWIS:n rentouttamista ajan suhteen antamalla ryhmätyöohjelman toteuttajan määritellään usein hajautetut komponentit päivittävät näkymänsä jaetusta tiedosta tai kuinka suuret muutokset välitetään jaettujen ja keskitettyjen komponenttien välillä (Dewan ja Choudhary, 1992; Greenberg ja Roseman, 1999). Suitessa käyttöliittymien kytkentää hallitaan asettamalla erityisiä kytkentäparametreja.

**Sessionhallintakomponentit.** Suitessa ei ole varsinaista valmista sessionhallintaa vaan ainoastaan erilaisia sellaisen toteuttamista tukevia valmiita rajapintafunktioita (Dewan ja Choudhary, 1992). Suite tarjoaa toteuttajille esimerkiksi funktion, jolla hajautettu komponentti aloittaa session, ja toisen, jolla kaikille sessiossa mukana oleville käyttäjille voi tiedottaa uuden henkilön liittyneen mukaan siihen.

### 3.3.2 Rendezvous

Rendezvous on 1990-luvun alussa kehitetty ryhmätyöohjelmien kehitystyökalu (Patterson *et al.*, 1990). Se toimii Unixin X-ikkunointiympäristössä ja se on toteutettu Common Lispillä.

**Hajautusarkkitehtuuri.** Rendezvousin arkkitehtuuriratkaisun ytimessä on näkemys järjestelmän käsittelemän tiedon keskittämisestä (Patterson *et al.*, 1990; Greenberg ja Roseman, 1999). Keskitettyä tietoa ylläpidetään kaikkien käyttäjien kesken jaetun tietoabstraktion avulla. Käyttäjakohtaiset näkymät tietoabstraktiosta sallivat poikkeavuuksia tiedon esittämiseen eri käyttäjien välillä, eli Rendezvous tukee WYSIWIS-periaatteen rentouttamista (2.1.2). Käyttäjakohtaisten näkymien tiedon välittämisen käyttäjälle hoitaa X-ikkunointijärjestelmä.

Roth ja Unger (2000) luokittelevat Rendezvous-sovelluskehityksen arkkitehtuuriratkaisun C1b-tyyppiseksi keskitetyksi ratkaisuksi (katso alakohta 3.1.1). Rendezvousia voi myös pitää eräällä tapaa toistettuna, koska se luo itsenäiset toistetut prosessit jokaista käyttäjää varten, mutta toistetut prosessit sijaitsevat kuitenkin keskitetysti samalla tietokoneella (Begole, 1998). Rendezvous on siten käsitteel-



lisesti hajautettu mutta fyysisesti keskitetty järjestelmä.

Greenberg ja Roseman (1999) kritisoivat Rendezvousin tapaa sijoittaa sekä keskitetty tietoabstraktio että käyttäjäkohtaiset tiedon näkymät samalle tietokoneelle. Ratkaisu on hidas ja huonosti skaalautuva, koska laskentaa ei hajauteta lainkaan. Lisäksi Rendezvous toimii ainoastaan sellaisissa ympäristöissä, joissa on käytössä X-ikkunointi.

**Ohjelmointiabstraktiot ja rajapinnat.** Rendezvous perustuu yksinkertaiseen perusparadigmaan nimeltä Abstraction-Link-View (ALV) (Patterson *et al.*, 1990; Greenberg ja Roseman, 1999), josta käytän tässä alakohdassa suomennosta abstraktio-linkki-näkymä. ALV on tapa erottaa järjestelmän käsittelemä tieto käyttöliittymästä. Abstraktio tarkoittaa keskitetyn tiedon esitystä, joka on keskitetty ja jaettu kaikkien käyttäjien kesken. Tiedon yhtenäisyyden takaaminen kaikkien käyttäjien kesken on keskeinen syy keskitetyn ratkaisun käyttöön. Käyttäjakohtaiset näkymät näyttävät käyttäjille erilaisiakin esityksiä abstraktion sisältämästä jaetusta tiedosta. Abstraktio ja näkymät on yhdistetty erityisellä linkillä, joka huolehtii näkymän päivittämisestä abstraktion tietoa vastaavaksi. Jos abstraktio muuttuu, linkki päivittää näkymän ja vastaavasti käyttäjän muuttaessa näkymää linkki päivittää abstraktiota.

**Usean käyttäjän käyttöliittymäkomponentit.** Rendezvousissa ei ole valmiita käyttöliittymäkomponentteja, ainoastaan sellaisten kehittämistä tukevia abstraktioita (Patterson *et al.*, 1990).

**Sessionhallintakomponentit.** Rendezvousin sessionhallinta tarjoaa ainoastaan yksinkertaisen mekanismin, jonka avulla käyttäjät voivat luoda session, liittyä siihen ja lähteä siitä pois (Greenberg ja Roseman, 1999).

### 3.3.3 GroupKit

GroupKit on tcl/tk:lla toteutettu ryhmätyöohjelmien kehitystyökalu (Roseman ja Greenberg, 1996). GroupKit toimii usealla laitteistoalustalla, koska tcl/tk on

käännetty useimmille nykyään käytössä oleville käyttöjärjestelmille. Toteutuksesta johtuva laitteistoriippumattomuus mahdollistaa GroupKit-sovellusten ajamisen usealta eri laitteistoalustalta, jopa siten, että samaan sessioon osallistuvat käyttäjät käyttävät eri käyttöjärjestelmiä. Tämä piirre on merkittävä etu alakohdissa 3.3.1 ja 3.3.2 esiteltyihin laitteistoriippuvaisiin Suiteen ja Rendezvousiin verrattuna.

**Hajautusarkkitehtuuri.** GroupKit:in arkkitehtuuriratkaisu vastaa pääpiirteittäin alakohdassa 3.1.2 käsittelemääni toistettua D2b-arkkitehtuuriratkaisua (Roth ja Unger, 2000). GroupKit:iä ei kuitenkaan voi sanoa täysin toistetuksi järjestelmäksi, koska se käyttää keskitettyä komponenttia välittääkseen toisinoille tietoja muista toisinoista näiden käynnistyessä. Kun toisinto on saanut keskitetyltä komponentilta tiedon muiden toisintojen sijainnista ja aktiivisista sessioista, siirtyy se täysin hajautettuun tilaan eikä enää ole riippuvainen keskitetystä komponentista (Begole, 1998; Greenberg ja Roseman, 1999).

**Ohjelmointiabstraktiot ja rajapinnat.** GroupKit tarjoaa ryhmätyöohjelman toteuttajalle kolme ohjelmointiabstraktiota (Roseman ja Greenberg, 1996). Ensimmäinen näistä on mahdollisuus kutsua kaikille toisinoille yhteistä aliohjelmamistä tahansa toisinnosta käsin. Toinen abstraktio on yhteisen tiedon esityksen tukeminen, jonka kaikki käyttäjät näkevät samanlaisena. Kaikki toisinnot voivat muokata yhteistä tiedon esitystä, josta itse asiassa on kopio jokaisella toisinnolla. Yhden toisinnon muuttaessa jaettua tietoa saavat kaikki toisinnot tiedon muutoksesta. Kolmas ohjelmointiabstraktio on viestien välittäminen yhdeltä toisinnolta kaikille muille toisinoille. Viestinvälitysmekanismiin kuuluu myös viestien vastaanotto ja käsittely.

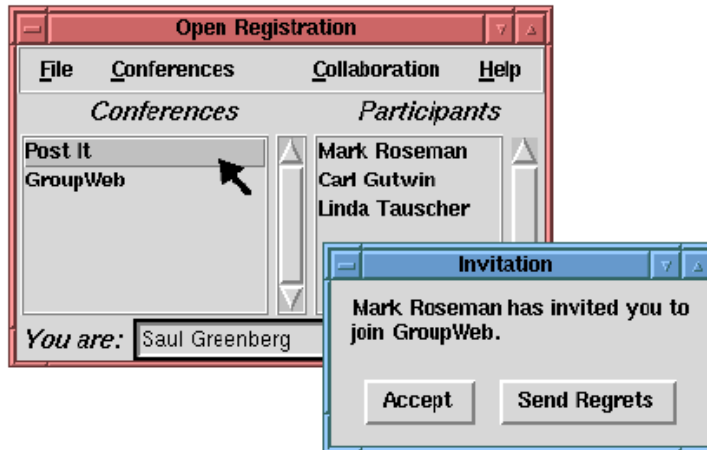
**Usean käyttäjän käyttöliittymäkomponentit.** GroupKit on oikea usean käyttäjän käyttöliittymien pioneeri. GroupKit:in käyttöliittymäkomponentit jakautuvat kolmeen luokkaan: jaettuihin osoitinvälineisiin, tietoisuuskomponentteihin ja sessioinformaatiota välittäviin komponentteihin (Greenberg ja Roseman, 1999). Osoitinvälineisiin tutustuimme kohdassa 2.3.3. GroupKit:in tietoisuuskomponentteihin kuuluu esimerkiksi jaettuja osoitinvälineitä, jaettuja vierityspalkkeja ja erilaisia usean käyttäjän hahmonäkymiä, joita esittelen alakohdassa 2.4.2.

Sessioninformaatiota välittävien komponenttien tarkoitus on parantaa ryhmän tietoisuutta siitä, ketä meneillään olevassa sessiossa on mukana. Kuvassa 20 on esimerkki komponentista, joka listaa sessiossa mukana olevat henkilöt ja antaa heistä tarkempiakin tietoja tarvittaessa.



Kuva 20: GroupKit'in sessioninformaatiota välittävä käyttöliittymäkomponentti (Greenberg ja Roseman, 1999).

**Sessionhallintakomponentit.** GroupKit:ssä on kehittynyt sessionhallinta, joka tukee kolmea erilaista tapaa liittyä sessioon (Greenberg ja Roseman, 1999). Käyttäjää voi luoda uuden session, liittyä olemassa olevaan sessioon tai hänet voidaan kutsua mukaan käynnissä olevaan sessioon. GroupKit tarjoaa ryhmätyöohjelman toteuttajalle jopa valmiin käyttöliittymäkomponentin, josta on esimerkki kuvassa 21.



Kuva 21: GroupKit:in sessionhallintakomponentin käyttöliittymä (Greenberg ja Roseman, 1999).

Kuvan 21 ruudunkaappauksessa on näkyvissä kaksi eri sessiota, joista GroupKit käyttää nimitystä konferenssi: Post It ja GroupWeb. Käyttäjä on valinnut Post It -konferenssin ja komponentti näyttää listan siihen osallistuvista henkilöistä Participants-otsikon alla. Kuvassa Mark Roseman -niminen käyttäjä on juuri kutsunut komponentin käyttäjän mukaan GroupWeb-konferenssiin. Käyttäjä voi joko hylätä tai hyväksyä kutsun.

### 3.4 Yhteenveto

Tässä luvussa olemme tutustuneet yhteiskirjoitusvälineiden ja yleisemmin ryhmätyöohjelmien toteuttamiseen liittyviin keskeisiin kysymyksiin. Arkkitehtuuriratkaisu on keskeinen osa ryhmätyöohjelman toteutusta. Yhteiskirjoitusvälineen toteuttajan on hyvä tuntea erilaisten arkkitehtuuriratkaisuiden hyvät ja huonot puolet, jotta hän voi valita parhaiten tarkoituspärisä sopivan vaihtoehdon. Yhtenäisyyden hallinta on ryhmätyöohjelmien välttämätön piirre, jota ei yhden käyttäjän järjestelmissä ole. Ryhmätyöohjelman toteuttajan on ymmärrettävä yhtenäisyyden hallinnalle asetettavat perusvaatimukset, koska ilman yhtenäisyyden hallintaa

ei järkevää ryhmätyöohjelmaa voi toteuttaa. Erilaisten kehitystyökalujen käyttö ryhmätyöohjelmia toteutettaessa on perusteltua, koska ne tarjoavat valmiita pakettiratkaisuja muun muassa arkkitehtuuriratkaisun ja yhtenäisyyden hallinnan suhteen, jotka ovat eräitä ryhmätyöohjelmien työläimmin toteutettavista piirteistä.

**Yhteiskirjoitusvälineen arkkitehtuuriratkaisut.** Yhteiskirjoitusvälineen arkkitehtuuriratkaisulle on olemassa useita erilaisia malleja. Rothin ja Ungerin (2000) luokittelumalli tunnistaa viisi erilaista hajautusarkkitehtuuria ryhmätyöohjelmille. Hajautusvaihtoehdoilla on kirjaintunnus, joka kirjoitetaan isolla kirjaimella. Kirjaintunnukseen liittyy numero, joka kuvaa hajautusvaihtoehdon alatyyppejä. Taulukossa 14 on koottu hajautusvaihtoehdot ja niihin liittyvä kirjaintunnus. Taulukossa on arkkitehtuurivaihtoehdon yhteydessä suluissa viite alakohtaan, jossa käsitellään kyseisten ratkaisujen alatyyppejä tarkemmin.

Taulukko 14: Rothin ja Ungerin (2000) luokittelumallin hajautusvaihtoehdot.

Hajautusvaihtoehto	Kuvaus	Tunnus
Arkkitehtuuri keskite- tyillä komponenteilla	Arkkitehtuuriratkaisussa on ainakin yksi keskitetty komponentti. Asiakaskomponentit eivät ole yhteydessä toisiinsa, ainoastaan keskitettyyn komponenttiin (3.1.1).	C
Suoran kommunikaa- tion arkkitehtuuri	Ei keskitettyjä komponentteja lainkaan, kaikki järjestelmän ilmentymät kommunikoivat toistensa kanssa (3.1.2).	D
Hybridiarkkitehtuuri	Ratkaisussa on ainakin yksi keskitetty komponentti sekä jonkinlaista kommunikaatiota asiakaskomponenttien välillä (3.1.3).	H
Asymmetrinen arkki- tehtuuri	Ei keskitettyä komponenttia, mutta järjestelmän komponenttien jako ei ole yhdenmukainen kaikkien toisintojen kesken, osalla toisinoista ei ole kaikkia järjestelmän komponentteja (3.1.3).	A
Useita palvelimia	Keskitetyt komponentit on hajautettu useaan eri tietokoneeseen (3.1.3).	E

Alakohdassa 3.1.4 esittelemäni Pattersonin (1995) vetoketjumalli on vaihtoehtoinen tapa kuvata ryhmätyöohjelman arkkitehtuuriratkaisua. Vetoketjumalli korostaa järjestelmän tilojen välistä kytkentää. Vetoketjumallin tuntemus on tärkeää ryhmätyöohjelmien kehittäjille, koska se on laajalle levinnyt tapa esittää käytettyjä perusratkaisuja.

**Yhtenäisyyden hallinta.** Ryhmätyöohjelman yhtenäisyydellä tarkoitetaan sitä, että kaikille järjestelmän käyttäjille taataan sama esitys käsitellystä tiedosta. Yhtenäisyyden hallinta on nimitys keinoille, joilla yhtenäisyyttä ylläpidetään. Kohdas-

sa 3.2 käsittelen kolmea tapaa hallita yhtenäisyyttä: lukitseminen, sarjoittaminen ja operationaalinen transformaatio. Lukitsemisesta ja sarjoittamisesta on erilaisia vaihtoehtoja, jotka perustuvat erilaisiin perusoletuksiin siitä, millaisia konfliktitilanteita yhtenäisyyttä hallittaessa voi syntyä. Operationaalinen transformaatio on eräs sarjoittamisen kehittynyt versio.

**Ryhmätyöohjelmien kehitystyökalut.** Kohdassa 3.3 käsittelemäni ryhmätyöohjelmien kehitystyökalut on tarkoitettu helpottamaan ryhmätyöohjelmien kehittämistä. Kehitystyökalut voivat auttaa ryhmätyöohjelmien kehittäjiä tarjoamalla valmiin sovelluskehityksen hajautusarkkitehtuureineen, ohjelmointiabstraktioita, sessionhallintakomponentteja ja usean käyttäjän käyttöliittymäkomponentteja (Greenberg ja Roseman, 1999). Erilaiset kehitystyökalut tukevat edellä mainittuja asioita hyvin vaihtelevasti. Osa niistä tarjoaa hyvinkin matalan tason työkaluja, kuten alakohdan 3.3.1 Suite ja alakohdan 3.3.2 Rendezvous. Alakohdassa 3.3.3 käsittelemäni GroupKit puolestaan paljon valmiita komponentteja, joita yhdistelemällä ryhmätyöohjelma on mahdollista toteuttaa helpommin.

## 4 Esimerkkijärjestelmiä

Yhteiskirjoitusvälineitä on toteutettu viimeisen kahdenkymmenen vuoden aikana melkoinen määrä. Münchenissä toimivan yliopiston tutkimusryhmän ylläpitämään listaan on kirjattu tasan 50 yhteiskirjoitusjärjestelmää (CSCW Pages @ Technische Universität München, 2001). Quilt (Fish *et al.*, 1988; Leland *et al.*, 1988) oli ensimmäisiä järjestelmiä, joka lähestyi vakavasti tietokoneavusteista yhteiskirjoittamista. Quiltin erityispiirteenä oli konfiguroitavien hierarkkisten roolien tukeminen. PREP (Neuwirth *et al.*, 1990; Cavalier *et al.*, 1991) edustaa Quiltin ohella varhaisia asynkronisia järjestelmiä. PREP:in erityispiirteenä on innovatiivinen käyttöliittymä, jonka toimivuuden puolesta puhuu järjestelmän kehityskaari, joka jatkuu tänäkin päivänä. GROVE (GRoup Outline Viewing Editor) (Ellis *et al.*, 1991) oli ensimmäisiä täysin synkronisia yhteiskirjoitusvälineitä, jotka tuki usean käyttäjän yhtäaikaista työskentelyä dokumentin parissa. GROVE on myös teknisesti edistynyt järjestelmä; sen arkkitehtuuriratkaisu on täysin hajautettu (katso kohta 3.1).

SEPIA (Structured Elicitation and Processing of Ideas for Authoring) (Haake ja Wilson, 1992; Streitz *et al.*, 1992) ja SASSE((Synchronous Asynchronous Structured Shared Editor) (Baecker *et al.*, 1993) edustavat seuraavaa kehityskelta yhteiskirjoitusvälineiden parissa, koska ne tukevat sekä asynkronista että synkronista työskentelyä. SEPIA esitteli maailmalle uuden työskentelymallin, jossa kirjoittajat voivat välillä työskennellä itsenäisesti ja toisaalta tarvittaessa tiiviissä yhteistyössä muiden kanssa (katso alakohta 2.1.2). SASSE on puolestaan tietoisuusinformaation pioneeri. Sen kehittäjiä kiinnosti erityisesti erilaiset reaaliaikaiseen työskentelyyn liittyvän tietoisuuden välittämiseen liittyvät käyttöliittymäkomponentit. SASSE:n käyttöliittymästä on ruudunkaappaus alakohdassa 2.4.2, jossa esitellään tietoisuusmekanismeja.

1990-luvun loppupuolella yhteiskirjoitusvälineitä ei enää ilmestynyt samaa tahtia kuin vuosikymmenen alussa. Kehitys keskittyi sen sijaan analysoimaan olemassa olevia välineitä ja niiden käyttöä eri tilanteissa (Mitchell *et al.*, 1995). Tutki-



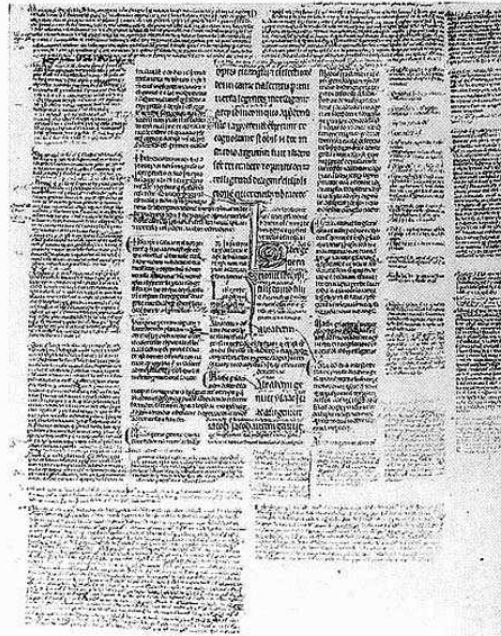
mus keskittyi prototyypin tuottamiseen sijaan yksityiskohtiin, kuten tietoisuuteen (Gutwin *et al.*, 1995; Gutwin ja Greenberg, 1996), kommunikaatiomodaliteetteihin (Neuwirth *et al.*, 1994a) ja dokumentin hallintaan (Neuwirth *et al.*, 1992; Munson ja Dewan, 1994; Lee *et al.*, 1998; Sun, 2000). Internetin kehitys 1990-luvun loppupuolella mahdollisti uusia teknisiä ratkaisuja, joista hyvä esimerkki on eräs uusimmista tulokkaista yhteiskirjoittamisen saralla, REDUCE (Real-time, Distributed, Unconstrained Collaborative Editing) (Yang *et al.*, 2000; Sun ja Sosic, 1999). REDUCE on Internet-pohjainen, täysin hajautettu järjestelmä, jossa sovelletaan uusinta verkkotekniikkaa. REDUCE-järjestelmää käytetään Internet-selaimen kautta, jolloin järjestelmä on käyttäjien ulottuvissa aina, kun he ovat Internet-yhteyden päässä.

Tarkastelen tämän luvun kohdissa tarkemmin kahta mielenkiintoista yhteiskirjoitusvälinettä soveltaen luvussa 2 käsittelemieni asioiden perusteella tekemääni piirteiden jaottelua (katso kohta 2.5), johon lisään kaksi uutta piirrekategoriaa luvusta 3: järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta (katso kohta 3.4).

Kohdassa 4.1 tutustumme PREP-järjestelmään, jonka juuret ulottuvat aina 1100-luvulle saakka. PREP edustaa vertailussa ensisijaisesti asynkroniseen työkentelyyn tarkoitettua yhteiskirjoitusvälinettä, joka painottaa kommenttien ja huomautusten käyttöä yhteiskirjoituksessa. Tutustumme SASSE:sta paljon vaikutteita saaneeseen Calliope-yhteiskirjoitusjärjestelmään kohdassa 4.2. PREP-järjestelmästä poiketen Calliope on ensisijaisesti synkroninen järjestelmä. Lisäksi Calliope edustaa tämän luvun alakohdissa käsitellyistä järjestelmistä sovelluskehityksen, GroupKit:in (katso alakohta 3.3.3), varaan rakennettua ryhmätyöohjelmaa. Kohdassa 4.3 tarkastelen kahden muun yhteiskirjoitusvälineen mielenkiintoisia piirteitä analysoimatta kyseisiä järjestelmiä tarkemmin.

## 4.1 PREP

PREP-järjestelmä on Carnegie Mellonin yliopistossa 1990-luvun alusta alkaen yhteiskirjoittamisen sosiaalisia ja kognitiivisia ulottuvuuksien tutkimista varten kehitetty yhteiskirjoitusväline (Neuwirth *et al.*, 1990; Cavalier *et al.*, 1991; Neuwirth *et al.*, 1992, 1994a,b). Nimi PREP tulee englanninkielisestä sanasta preparation, joka merkitsee valmistelua tai esikäsittelyä. Järjestelmän kehittäjät tahotoivat näin ilmaista sen olevan tarkoitettu ensisijaisesti ryhmässä tuotettavan dokumentin esikäsittelyyn, kuten ideointiin, kirjoittamiseen ja kommentoimiseen, ei suinkaan lopulliseen muotoiluun (Neuwirth *et al.*, 1990). PREP-järjestelmällä on mielenkiintoinen sidos varhaiselle keskiajalle; järjestelmän käyttöliittymä on kehitetty munkkien 1100-luvulta lähtien käyttämää käsikirjoitusten kommentointimekanismia mukaillen (Cavalier *et al.*, 1991). Kuvassa 22 on sivu 1200-luvulta peräisin olevasta käsikirjoituksesta, joka jakautuu selvästi eri sarakkeisiin. Sivun teksti on järjestetty sarakkeittain siten, että alkuperäinen raamatullinen ote on keskellä, ja sitä kommentoivat ja tutkivat muistiinpanot sijaitsevat sivun laidoilla. Muistiinpanot on liitetty keskellä sijaitsevaan päätekstiin erilaisilla kalligrafisilla merkinnöillä (joita ei näy kuvassa). Lisäksi kirjasinkoko ja -tyyli erottavat päätekstin ja eri kirjoittajien lisäykset toisistaan. Tämä varhainen esimerkki osoittaa, kuinka entisaikojen tiedeyhteisö hyödynsi yhteiskirjoittamista tavalla, joka on säilynyt selkeänä ja tulkittavana näihin päiviin saakka.

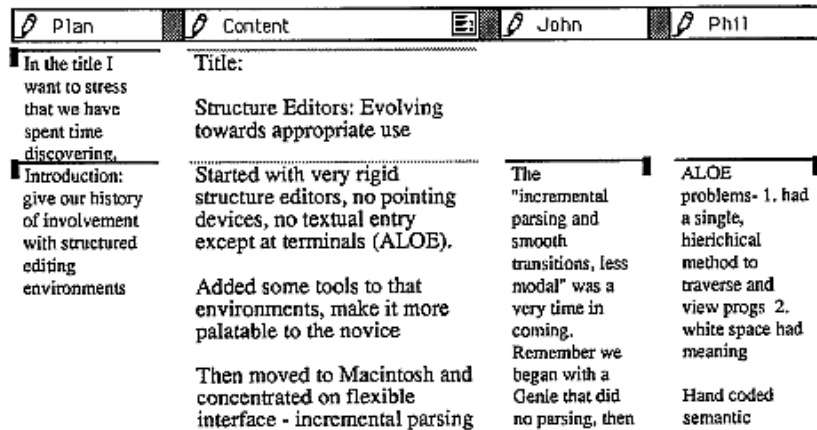


Kuva 22: Sivu 1200-luvulta peräisin olevasta käsikirjoituksesta (Medieval Manuscript Manual, 2001).

Toimivaksi havaittu ratkaisu innosti PREP-järjestelmän kehittäjiä tutkimaan tarkemmin munkkien käyttämää yhteiskirjoitusjärjestelmää. Tutkimusten perusteella he kehittivät keskeisiä suuntaviivoja tietokoneavusteisen yhteiskirjoittamisen yhteydessä tärkeille kommentointitoiminnoille, joita käsitellen alakohdassa 2.3.2. Myös PREP-järjestelmän käyttöliittymä on syntynyt vanhojen käsikirjoitusten inspiroimana. PREP-järjestelmän erikoispiirteenä onkin tämä nimenomainen käyttöliittymä, jota on kutsuttu taulukkolaskennan ja tekstinkäsittelyohjelman ristisiitokseksi.

Kuvassa 23 on ruudunkaappaus PREP-järjestelmästä. Ruudunkaappauksessa näkyy, kuinka käyttöliittymä on jaettu sarakkeisiin. Vasemmanpuoleisimmassa sarakkeessa on suunnitelma dokumentille, toisessa vasemmalta on työstettävä dokumentti. Kaksi oikeanpuoleisinta saraketta kuuluvat ryhmän jäsenille ja toimivat kommentointisarakkeina. PREP:in tekijät toteavat järjestelmänsä olevan dynaa-

minen versio varhaiskeskiajan munkkien järjestelmästä liittää selityksiä käsikirjoituksiin (Neuwirth *et al.*, 1990).



Kuva 23: Ruudunkaappaus PREP-järjestelmästä (Cavalier *et al.*, 1991).

#### 4.1.1 Yhteinen dokumentti

**Perustekstinkäsittely.** PREP-järjestelmä toimii Macintosh System -käyttöjärjestelmän versiossa 6.0 sekä myöhemmissä versioissa, joskaan toimivuutta ei ole täysin kattavilla testeillä varmistettu. PREP-järjestelmä tukee perustekstinkäsittelypiirteitä, kuten kirjasinkokoja, -tyylejä ja -lajeja (Coutouriaux *et al.*, 1993). Lisäksi PREP osaa tallettaa ja avata dokumentteja, jotka on kirjoitettu ASCII-, RTF- ja MacWrite-tiedostomuodoissa. PREP-järjestelmän ensimmäinen versio, joka oli puhtaasti asynkroninen editori, toimii hyvin myös itsenäisesti tavallisena tekstieditorina erikoisella sarakkekäyttöliittymällä höystettynä ilman verkkoyhteyttä. Uudemman PREP-järjestelmän version toimivuudesta itsenäisesti ei löydy kirjallisuudesta tietoa.

**Käyttöliittymä.** PREP-järjestelmässä käyttäjä voi säätää käyttöliittymänsä kytkeä suhteessa muihin käyttäjiin vuorovaikutusparametrien avulla (Neuwirth *et al.*, 1994b). Vuorovaikutusparametrejä voidaan tarkastella sekä käyttöliittymä-

män kytkennän (katso alakohta 2.1.2) että tietoisuuden (katso kohta 2.4) kannalta. PREP:in käyttäjät voivat määrittellä, kuinka nopeasti muutokset välittyvät, milloin muutokset välittyvät ja minkä kokoisia muutoksia vuorovaikutusparametrit käsittelevät. Kuvassa 24 on ote PREP:in käyttöliittymästä, jolla käyttäjä voi säätää sitä, kuinka hänen tekemät muutokset näkyvät muille käyttäjille.

The image shows a settings dialog box with three sections, each with a title and a list of options:

- Send changes to collaborators:**
  - Only when you push the "Send Changes" button
  - Automatically when any change is complete
- Changes are considered complete when:**
  - You push the "Mark As Changed" button
  - You make changes and move to another paragraph or column
  - You finish a particular editing operation (type, cut, paste, etc.)
  - You make any change, including every keystroke
- Send changes over the network:**
  - Via "Express Mail"
  - Via "First Class"
  - Via "Parcel Post"

Kuva 24: Työskentelyn intensiteetin säätäminen PREP-editorissa (Neuwirth *et al.*, 1994b).

Ruudunkaappauksessa näkyvä ylin kehys hallitsee aikaa. Vaihtoehtoina on automaattinen muutosten ilmoittaminen niiden tapahtuessa ja manuaalinen muutosten ilmoittaminen käyttäjän painaessa tarkoitusta varten olevaa nappia. Keskimmissä kehyksessä käyttäjä määrittelee, milloin muutos on tapahtunut. Vaihtoehdot kattavat manuaalisen muutoksesta ilmoittamisen lisäksi kappaleen muokkaamisen ilmoittamisen, muokkausoperaation ilmoittamisen ja kaikkien muutosten ilmoittamisen yksittäistä kirjainta myöten. Alimman kehyksen avulla käyttäjä voi säätää sitä, kuinka nopeasti järjestelmä yrittää välittää tiedon muutoksesta. "Express Mail" -vaihtoehto merkitsee nopeinta, eniten verkkoresursseja tarvitsevaa välitystapaa ja vastaavasti "Parcel Post" hitainta, resursseja säästävää välitystapaa.

**Versionhallinta.** PREP-järjestelmä ei sisällä varsinaista versionhallintajärjestel-

mää, joka sisältäisi alakohdassa 2.1.3 mainitut versionhallinnan peruspiirteet. PREP:issä on kuitenkin toiminto, jolla dokumentin eri versioita on mahdollista vertailla. Neuwirth *et al.* (1992) kutsuvat kehittämäänsä toimintoa joustavaksi vertailuksi (flexible diff). Joustavan vertailun perusajatuksena on vertailun säätämisen mahdollisuus kutakin tilannetta vastaavaksi erityisten parametrien avulla. Käsittelin joustavaa vertailua parametreineen tarkemmin alakohdassa 2.1.3. PREP esittää joustavan versioiden vertailun hyödyntäen sarakepohjaista dokumentin esitystään. Kuvassa 25 on PREP-järjestelmän ruudunkaappaus, jossa vasemmanpuoleisimmassa sarakkeessa on alkuperäinen versio, toisessa sarakkeessa vasemmalta lukien uusi versio ja kolmannessa sarakkeessa vasemmalta vertailuraportti. Vertailuraportissa lisäykset ovat kursivissa ja poistot alleviivattuja.

Original	Revision	Comparison	Explanation
Way to decrease coordination difficulty is to communicate less distracting information and more relevant information.	One way to decrease coordination difficulty is to communicate less distracting information and more relevant information - in effect, lowering the "signal-to-noise" ratio.	<i>Way</i> <del>One way</del> ... - in effect, lowering the "signal-to-noise" ratio	"signal-to-noise" - is this an understood phrase?
The approach is incorporated in the "work in preparation" (PREP) Editor, a word processor being developed to study and to support collaborative writing processes.	The approach to comparison is incorporated in the "work in preparation" (PREP) Editor, a writing environment being developed to study and to support collaborative writing processes.	<del>to comparison</del> ... a writing environment <u>a word processor</u> ...	I don't want to position PREP as a word processor.

Kuva 25: Joustavaa versioiden vertailua PREP-editorissa (Neuwirth *et al.*, 1992).

#### 4.1.2 Koordinointi

**Roolit ja sosiaaliset protokollat.** PREP-järjestelmä ei tue lainkaan ennalta määrättyjä, eksplisiittisiä rooleja tai sosiaalisia protokollia. Se pyrkii paremminkin tarjoamaan käyttäjilleen valikoimaa erilaisia asetuksia ja työkaluja, joiden avulla ryhmän jäsenet voivat omaksua tiettyjä rooleja järjestelmän tukemana. Ala-

kohdassa 4.1.1 käsittelemäni joustavan versioiden vertailun eräs motivoiva tekijä on erilaisten roolien edellyttämät erilaiset vaatimukset versioiden vertailussa (Neuwirth *et al.*, 1992). Samoin PREP-järjestelmän vuorovaikutusparametrit (josta lisää tämän alakohdan lopussa) on tarkoitettu myös tukemaan erilaisten roolien vaatimaa erilaista vuorovaikutusastetta (Neuwirth *et al.*, 1994b). Tiettyyn kohtaan keskittynyt kirjoittaja saattaa haluta olla rauhassa eikä ole kiinnostunut muiden tekemistä muutoksista muualla dokumentissa, kun taas toimittaja odottaa kärsimättömänä viimeisiä muutoksia ennen viimeistä aikarajaa.

**Kirjoitusprosessi.** PREP tuki alunperin pelkästään asynkronista, vähemmän intensiivistä työskentelyä (Neuwirth *et al.*, 1990; Cavalier *et al.*, 1991). Tuki koostui lähinnä huomautusten ja kommenttien lisäämisestä dokumenttiin ja dokumentin eri versioiden hallintaan. Myöhempi versio PREP:istä lisäsi järjestelmään synkronisia piirteitä, kuten usean yhtäaikaisen kirjoittajan tuen ja parametroidun dokumentin muutosten välittämisen erityisiä vuorovaikutusparametrejä säätämällä (Neuwirth *et al.*, 1994b). PREP:in ainoa projektinhallintaan viittaava piirre on aikarajojen liittäminen kommentointisarakkeisiin. Aikarajat voivat olla pakollisia tai suositeltavia (Neuwirth *et al.*, 1994b). Pakollisen aikarajan kuluttua saraketta ei voi enää muuttaa. Suositeltava aikaraja sallii sarakkeen muuttamisen myös aikarajan jälkeen mutta huomauttaa myöhäistä käyttäjää aikarajan ylittämisestä.

**Kirjoitussessio.** PREP:issä kirjoitussession luonti tapahtuu siten, että yksi käyttäjä käynnistää PREP-editorin, avaa siihen dokumentin ja antaa muille oikeuden olla yhteydessä juuri aloitettuun kirjoitussessioon (Hofte, 1996). Aloittava käyttäjä voi määritellä, mitä oikeuksia muilla käyttäjillä on mihinkin sarakkeisiin nähden. PREP rinnastaa session ja dokumentin tukien vain yhtä samanaikaista kirjoitussessiota, jota edustaa muokattavaksi avattu dokumentti.

### 4.1.3 Kommunikointi

**Vapaamuotoinen keskustelu.** PREP ei tue vapaamuotoista keskustelua missään modaliteetissa, ellei seuraavassa tekstikappaleessa käsiteltäviä huomautuksia ja

kommentteja käsitetä sellaiseksi.

**Huomautukset ja kommentit.** PREP-järjestelmä tukee erityisen hyvin huomautuksia ja kommentteja. Sen toteuttajien eräitä motivoiva tekijä olikin muinainen kommentointimenetelmä sekä yhteiskirjoittamisen ja kommentoinnin välisen suhteen tutkiminen (Cavalier *et al.*, 1991; Neuwirth *et al.*, 1990). PREP:in toteuttajien keskeisenä pyrkimyksenä oli tuottaa itsensä selittävä käyttöliittymä, jossa käyttäjät näkevät kerralla kaiken relevantin informaation: varsinaisen dokumentin ja siihen liittyvät huomautukset ja kommentit. Kuvan 23 ruudunkaappauksesta ilmenee hyvin PREP-järjestelmän toteuttajien pyrkimys vähentää visuaalisia esteitä kirjoittajien väliltä. PREP toteuttaa alakohdassa 2.3.2 mainitut neljä huomautuksille ja kommentteille asetettavaa perusvaatimusta.

**Sanaton viestintä.** PREP ei tue sanatonta viestintää.

#### 4.1.4 Tietoisuus

**Työtilatietoisuus.** PREP-järjestelmässä painottuu kommentteihin ja huomautuksiin liittyvä tietoisuusinformaatio. Alakohdassa 2.3.2 mainitut huomautuksille ja kommentteille asetettavat vaatimukset voidaan tulkita tietoisuuden kannalta rinnastamalla vaatimukseen kysymyksiä alakohdassa 2.4.1 esitellyn tietoisuusviitekehysten lailla (katso mainitun alakohdan taulukko 8). Taulukossa 15 on huomautuksille ja kommentteille asetettavien vaatimusten kanssa rinnastettu kysymyksiä, joita käyttäjä voi esittää miettiessään, mikä dokumentin ja siihen liitettyjen huomautusten tila oikein on.



Taulukko 15: Kommentoinnille asetettavien vaatimusten suhde tietoisuusinformaatioon.

<b>Vaatus kommentteille</b>	<b>Tietoisuuskysymys</b>
Pääteksti erottuu kommenteista.	Mikä osa näkyvästä tuotoksesta on kommenttia?
Kommenttien tulee olla näkyvillä päätekstiä luettaessa.	Onko tätä kohtaa päätekstissä kommentoitu? Missä mahdolliset kommentit ovat?
Kommenttien ja päätekstin suhde on selkeästi havaittavissa.	Mihin tämä kommentti liittyy?
Kommenttien kirjoittaja selviää helposti.	Kuka tämän kommentin on kirjoittanut?

PREP-järjestelmä vastaa kaikkiin taulukossa 15 mainittuihin kysymyksiin yksinkertaisesti hyvin suunnitellun käyttöliittymänsä kautta. Kommentit erottuvat PREP:in käyttöliittymässä päätekstissä, koska näille on omat selkeästi merkityt sarakkeensa. Järjestelmän käyttäjä näkee sekä päätekstin että kommenttisarakkeet yhtä aikaa näytössä. Kommenttien ja päätekstin suhde on PREP:issä selkeä, koska pääteksti ja kaikki siihen liitetyt kommentit sijaitsevat horisontaalisesti samalla tasolla. Kommenttien kirjoittajat puolestaan selviävät helposti yhdellä silmäyksellä, koska kommentoijan henkilöllisyys käy ilmi kunkin kommenttisarakkeen otsikosta ja tietyn sarakkeen kaikki kommentit ovat tämän henkilön hengentuotteita.

PREP-järjestelmässä on mahdollista muokata erityisiä vuorovaikutusparametrejä, jotka on kuvailtu alakohdassa 4.1.1. Vuorovaikutusparametrien muuttaminen vaikuttaa siihen, kuinka käyttäjä näkee muiden käyttäjien tekemät muutokset ja kuinka muut käyttäjät näkevät hänen tekemänsä muutokset. Vaikka parametrien säätäminen ei vaikuta suoranaisesti tietoisuusinformaation välittämiseen, koska PREP ei sellaista suoranaisesti välitä, vaikuttaa se epäsuorasti käyttäjien tietoisuuteen ympäristöstään. PREP:in vuorovaikutusparametrien avulla tietty käyttäjä voi päättää jossain määrin, mitä muut käyttäjät näkevät hänen tekevän, eli vaikut-

taa epäsuorasti siihen, mistä muut ovat tietoisia. Tämä piirre voi heikentää alakohdassa 2.4.1 taulukossa 8 esitettyjä työtilatietoisuuteen liittyviä tietoisuusnäkökulmia tai antaa jopa harhaanjohtavaa tietoa. Käyttäjän valta itseensä liittyvään tietoisuusinformaatioon välittämiseen on mielenkiintoinen kysymys, johon ei ole yksiselitteistä vastausta. Tulisiko käyttäjälle sallia valta päättää, miten ja missä määrin muut hänet näkevät yhteiskirjoitusympäristössä?

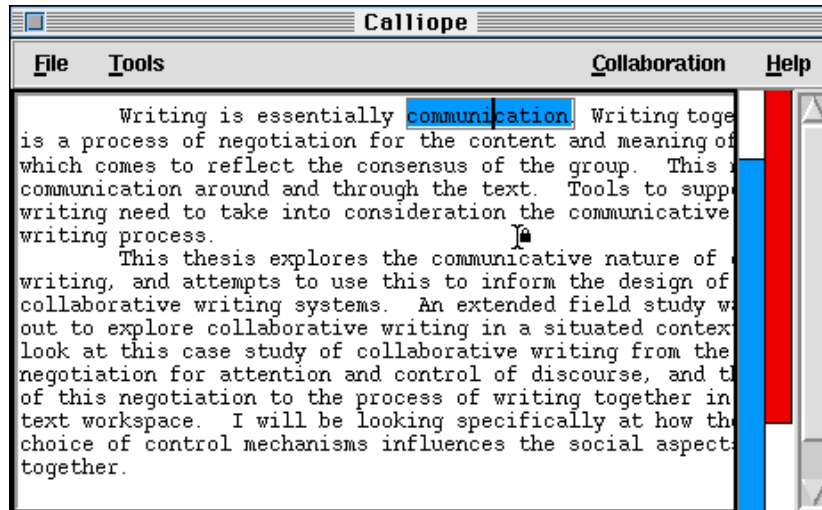
#### 4.1.5 Järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta

**Järjestelmäarkkitehtuuri.** PREP:in järjestelmäarkkitehtuuri on alakohdassa 3.1.1 esittelemäni keskitetty C2. PREP tallettaa dokumentin kaikkien käyttäjien kesken jaettuun keskitettyyn tietokantaan.

**Yhtenäisyyden hallinta.** PREP-editorin yhtenäisyyden hallinta on ilmeisesti aika primitiivisellä tasolla, ainakaan järjestelmää kuvaavissa artikkeleissa ei asiasta erityisemmin puhuta. PREP-järjestelmässä dokumentti on jaettu sarakkeittain ja sarakkeiden sisällä vielä käyttäjän määrittelemiin osiin. PREP tallettaa dokumentin osat tietokantaan, yhtenäisyyden hallinta tapahtuu ilmeisesti jonkinlaisen lukitusmekanismin avulla (Hofte, 1996) (katso alakohta 3.2). PREP-järjestelmän kehittäjät esittivät visioita ja vaatimuksia yhtenäisyyden hallinnalle (Neuwirth *et al.*, 1994b), mutta niiden toteutumisesta ei ole kirjallisuudessa tietoa. On todennäköistä, että mainitut uudet piirteet sisällytettiin PREP-järjestelmän kaupalliseen jälkeläiseen CommonSpace:een.

## 4.2 Calliope

Alex Mitchellin (1996) kehittämä Calliope-järjestelmä on nimetty kreikkalaisesta mytologiasta löytyvän kirjoittajien ja taiteilijoiden muusan, Zeus-jumalan tyttären Calliopen mukaan. Calliope on ensisijaisesti synkronista työskentelyä varten suunniteltu järjestelmä, joka painottaa reaaliaikaisen tietoisuuden välittämistä (katso kohta 2.4). Kuvassa 26 on Calliopen ruudunkaappaus.



Kuva 26: Ruudunkaappaus Calliope-järjestelmästä (Mitchell, 1996).

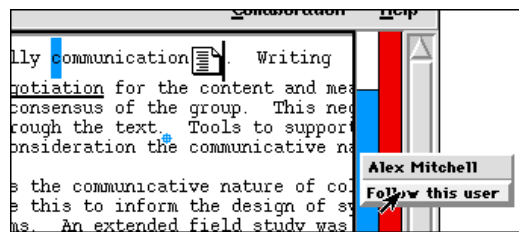
Calliopen juuret ulottuvat Posnerin ja Baeckerin (1992) tekemään perustutkimukseen yhteiskirjoittamisen saralla ja tutkimuksien perusteella kehitettyyn yhteiskirjoitusvälineen vaatimusmäärittelyyn (Baecker *et al.*, 1993). Calliopella on kaksi edeltäjää: yhteiskirjoitusvälineet SASE ja SASSE (Baecker *et al.*, 1993, 1994). Calliopen on vaikuttanut SASE:n ja SASSE:n avulla hankittujen kokemusten lisäksi myös kenttätutkimukset yhteiskirjoittamisen tukemisessa tietokoneitse (Mitchell *et al.*, 1995; Mitchell, 1996).

Calliope on rakennettu GroupKit-komponenttikokoelman varaan, jota käsitellen alakohdassa 3.3.3 (Mitchell, 1996). GroupKit perustuu tcl/tk:hon, joten Calliope on toteutettu myös tcl/tk:lla. Calliopen vahvuus on sen laajennettavuus, joka on GroupKit-pohjan seuraus. Calliopen yhteyteen on yksinkertaista lisätä erilaisia komponentteja, jotka mahdollistavat esimerkiksi projektinhallintaa tai kommunikaatiota. Käsitellen tässä kohdassa Calliopeta itsenäisenä sovelluksena huomioimatta siihen mahdollisesti lisättävissä olevia laajennuksia.

### 4.2.1 Yhteinen dokumentti

**Perustekstinkäsittely.** Calliope-järjestelmä on toteutettu GroupKit-komponenttikokoelman (katso alakohta 3.3.3) avulla, joten se toimii kaikissa ympäristöissä, jotka tukevat tcl/tk:ta. Nykyisin tuki tcl/tk:lle löytyy useimmille eri laitteistoalustoille. Calliopen tekstinkäsittelyvalmiudet ovat melko suppeat, se tukee ilmeisesti ainoastaan perustekstin kirjoittamista.

**Käyttöliittymä.** Calliope-järjestelmä tukee tiukkaa WYSIWIS-käyttöliittymää ja tilan suhteen rentoutettua käyttöliittymää. Calliopen käyttäjä voi kytkeytyä tietyn toisen käyttäjän näkymään, jolloin hän näkee saman näkymän kuin toinen käyttäjä. Kuvassa 27 näkyy, kuinka käyttäjä on kytkeytymässä Alex Mitchell -nimisen käyttäjän käyttöliittymään.



Kuva 27: Tiukan WYSIWIS-käyttöliittymän kytkeminen päälle Calliopessa (Mitchell, 1996).

**Versionhallinta.** Calliope ei tue versionhallintamekanismeja, ainakaan Mitchell (1996) ei mainitse versionhallinnasta mitään.

### 4.2.2 Koordinointi

**Roolit ja sosiaaliset protokollat.** Calliopessa ei ole erityisiä rooleja, kaikilla käyttäjillä on myös yhteneväiset oikeudet järjestelmässä. Järjestelmän toteuttaja Mitchell (1996) perustelee tätä sillä, että hän halusi järjestelmän käyttäjien roolien syntyvän luonnollisten, järjestelmän ulkoisten, sosiaalisten protokollien kautta.

**Kirjoitusprosessi.** Calliope ei sisällä erityisiä prosessinhallintamekanismeja.

**Kirjoitussessio.** Calliope käyttää GroupKit-komponenttikokoelman avointa sessionhallintamekanismia, jota käsitellen alakohdassa 3.3.3 (Mitchell, 1996). Calliope-järjestelmä ylläpitää keskitettyä sessionvälityspalvelua, johon Calliopen toisinnot ottavat yhteyttä käynnistyessään. Sessionvälityspalvelu listaa käyttäjälle kaikki valittavissa olevat sessiot ja välittää toisinnolle tiedon muiden toisintojen sijainnista suoraa kommunikaatiota varten. Kuvassa 28 on käyttäjälle aukeava sessionvalitsemisnäkyvä.



Kuva 28: Kirjoitussession aloittaminen Calliope-järjestelmässä (Mitchell, 1996).

Calliope käyttää sessiosta nimitystä konferenssi. Calliopessa kirjoitussessioon osallistutaan rekisteröitymällä mukaan jo olemassa olevaan sessioon (kuvassa 28 on yksi kirjoitussessio, jonka nimi on Calliope). Rekisteröitymisen yhteydessä näkyy, ketkä käyttäjistä ovat mukana eri kirjoitussessioissa. Kuka tahansa käyttäjä voi aloittaa kirjoitussession ja liittyä mihin tahansa kirjoitussessioon.

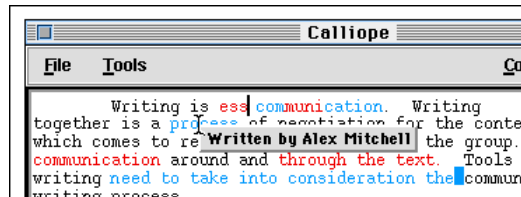
#### 4.2.3 Kommunikointi

**Vapaamuotoinen keskustelu.** Calliopessa ei ole tukea puheen välittämiseksi eikä tekstipohjaiselle keskustelulle (Mitchell, 1996). Jonkinlainen tekstipohjainen kes-

kustelu on mahdollista seuraavassa kappaleessa käsittelemieni huomautusten ja kommenttien kautta, jotka ovat eräänlaisia kaikkien muokattavissa olevia teksti-ikkunoita.

**Huomautukset ja kommentit.** Calliope tukee kolmenlaisia huomautuksia ja kommentteja: sisäisiä, ulkoisia sekä henkilökohtaisia (Mitchell, 1996). Sisäiset kommentit liitetään tekstiin pienen teksti-ikkunan muodossa. Kuka tahansa käyttäjä voi liittää dokumenttiin kommenttitekstin sisältävän teksti-ikkunan. Kuka tahansa käyttäjä saa myös muokata teksti-ikkunan sisältämää tekstiä, kuitenkin vain yksi käyttäjä kerrallaan. Ulkoiset kommentit ovat viitteitä järjestelmän ulkoisiin WWW-sivuihin URL-osoitteen muodossa. Kirjoittajat voivat liittää linkin ulkoiseen kommenttiin tai lähteeseen URL:n määrittelemällä. Henkilökohtaiset kommentit toimivat samalla tavalla kuin sisäiset kommentit, mutta ne ovat vain yhden käyttäjän näkyvissä ja muokattavissa.

**Sanaton viestintä.** Calliopessa on erityinen jaettu piirustusalue, jossa käyttäjät voivat tekstin syöttämisen lisäksi piirtää yksinkertaisella piirtotyökalulla (Mitchell, 1996). Vaikka jaettu piirustusalue ei edusta pelkästään sanatonta viestintää, niin se voi tukea myös sitä. Hieman mielikuvitusta käyttäen voisi kuvitella jonkun käyttäjän suttaavan kiivaan näköisesti editorin kommentin päälle piirustustyökalullaan. Tämä todellakin olisi sanatonta viestintää.

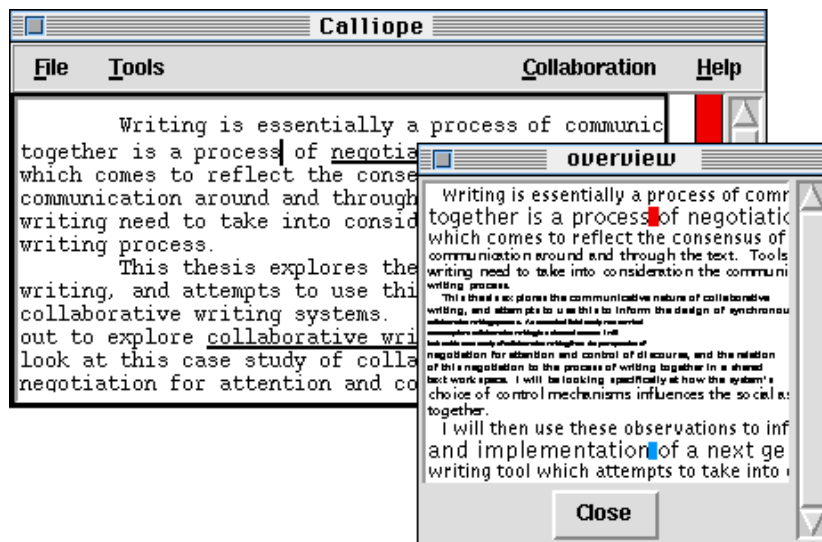


Kuva 29: Calliopen jaettu piirustusalue (Mitchell, 1996).

Kuvassa 29 näkyy Calliopen jaettu piirustusalue, jossa kaksi eri värein eroteltua käyttäjää kommunikoi suunnitelmistaan.

## 4.2.4 Tietoisuus

**Työtilatietoisuus.** Eräs Calliopen suunnittelun peruslähtökohdista on tietoisuusinformaation välittäminen (Mitchell, 1996). Calliopen käyttöliittymässä on kaksi tietoisuusmekanismia: jaetut vierityspalkit ja eräs alakohdassa 2.4.2 esittelemäni hahmonäkymän sovellus. Kuvassa 30 näkyy Calliopen yleisnäkymä dokumentista, joka on niin kutsuttu kalansilmänäkymä. Kalansilmänäkymässä osa näkymästä esitetään eri tavalla kuin muut osat. Tässä tapauksessa dokumentista näytetään yleisnäkymä, jossa eri käyttäjien näkemät osat dokumentista näkyvät isommalla tekstillä ja muut osat pienemmällä tekstillä.



Kuva 30: Calliopen kalansilmä-yleisnäkymä (Mitchell, 1996).

Yleisnäkymässä näkyy jokaisen käyttäjän kursorin kohta tekstissä värillisenä neulönä. Tavalliseen hahmonäkymään verrattuna kalansilmä-yleisnäkymä näyttää enemmän informaatiota: se näyttää kirjoittajan sijainnin lisäksi myös tekstin, jota kirjoittaja mahdollisesti kirjoittaa tai lukee.

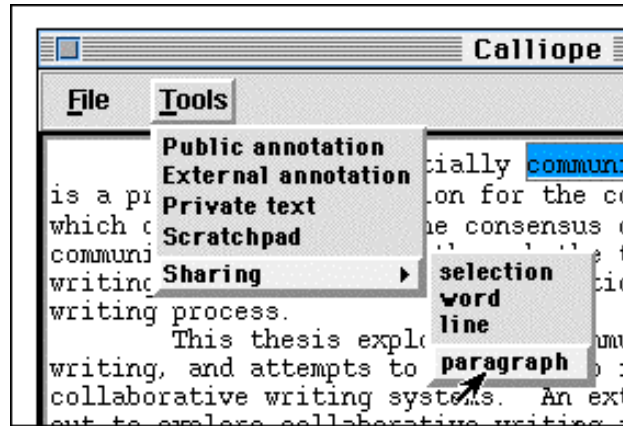
#### 4.2.5 Järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta

**Järjestelmäarkkitehtuuri.** Koska Calliope on toteutettu GroupKit-kehitystyökalun sovelluskehityksen varaan, noudattelee sen arkkitehtuuri GroupKit:in arkkitehtuuria, jota käsitellen tarkemmin alakohdassa 3.3.3. Calliopessa on kuitenkin GroupKit:in arkkitehtuurista poikkeavia piirteitä (Mitchell, 1996). Calliopen järjestelmäarkkitehtuuri on hajautettu jaetun tiedon kannalta, jokaisella järjestelmän toisinnolla on oma kopionsa dokumentista. Sessionhallinta ja osa yhtenäisyyden hallintaa hoidetaan Calliopessa keskitettyjen komponenttien kautta. Calliopen arkkitehtuuriratkaisu on sekoitus toistettua ja keskitettyä, joten se on hybridijärjestelmä, joka muistuttaa suuresti alakohdassa 3.1.3 esiteltyä hybridiarkkitehtuuriratkaisua H3.

Calliopen keskitetty sessionhallintakomponentti (jota käsitellen käyttäjän näkökulmasta alakohdassa 4.2.2) säilyttää tietoa kaikista aktiivisista kirjoitussessioista ja niihin osallistuvien henkilöiden tiedot sekä järjestelmässä aktiivisina olevien hajautettujen toisintojen sijainnit (Mitchell, 1996). Sessionhallintakomponentti välittää tiedot sessioon osallistuvista toisunnoista ja niiden käyttäjistä sessioon liittyville toisinnolle mahdollistaen näin suoran kommunikaation toisintojen välillä. Calliopen osaksi keskitetyn yhtenäisyyden hallinnan toiminnasta kerron lisää seuraavassa kappaleessa yhtenäisyyden hallinnan yhteydessä.

**Yhtenäisyyden hallinta.** Calliopen yhtenäisyyden hallinnassa yhdistyy optimistinen sarjoittaminen (katso alakohtaa 3.2.1) ja optimistinen lukitseminen (katso alakohtaa 3.2.2) (Mitchell, 1996). Käyttäjän kannalta katsottuna Calliope hallitsee yhtenäisyyttä käyttäjälähtöisen optimistisen lukitusmekanismin kautta. Halutessaan muokata dokumenttia käyttäjä lukitsee asianmukaisen kohdan valitsemallaan lukitustarkkuudella. Kuvassa 31 on ruudunkaappaus Calliopesta, jossa näkyy käyttäjälle tarjottavat lukitsemistasot: sana, rivi, kappale ja vapaasti valittu alue yhteisestä dokumentista.





Kuva 31: Lukittavan objektin koon valitseminen Calliope-järjestelmässä (Mitchell, 1996).

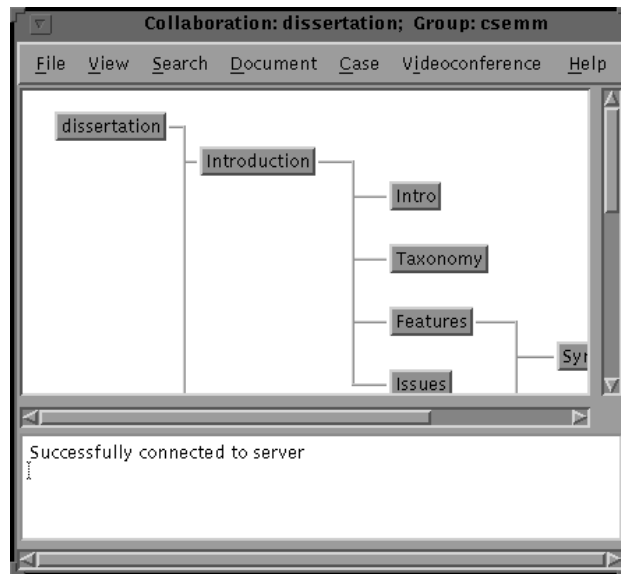
Käyttäjän tehdessä lukituspyynnön Calliope tarkastaa aluksi paikallisesti, onko kyseessä oleva kohta jo lukittu. Jos paikallinen Calliopen toisinto ei ole tietoinen muiden käyttäjien tekemistä ristiriitaisista lukituksista, myöntää se käyttäjälle lukituksen kohtaan. Sitten Calliopen toisinto lähettää lukituspyynnön keskitetyn kommunikaatiokomponentin kautta muille toisinoille tiedoksi. Keskitetty komponentti kuitenkin käsittelee lukituspyyntöjä sarjoittamisen kautta (Mitchell, 1996). Kaikki lukituspyynnöt käsitellään aikajärjestyksessä, ja konfliktitilanteet ratkaistaan ensiksi ehtineen hyväksi. Konfliktitilanteessa hävinneen lukituspyynnön keskitetylle komponentille lähettäneen toisinnon on peruttava optimistisessä hengessä jo myöntämänsä lukituspäätös ja kaikki muutokset, jotka käyttäjä on mahdollisesti ehtinyt tehdä. Sarjoitusratkaisu on optimistinen, koska se olettaa, ettei konflikteja juurikaan synny. Calliopen keskitetyn sarjoitusmekanismin on lähtökohtana tarkoitus hoitaa harvinaiset konfliktitilanteet.

## 4.3 Muita järjestelmiä

Tässä kohdassa esittelen joidenkin yhteiskirjoitusvälineiden mielenkiintoisia piirteitä. Alakohdassa 4.3.1 tutustumme tapaan esittää dokumentti käyttöliittymässä sen loogisen rakenteen mukaan. Eräs yhteiskirjoitusvälineen keino hallita rooleja on käyttäjien oikeuksien määrittäminen rooliin mukaisesti, mitä tarkastelemme alakohdassa 4.3.2.

### 4.3.1 DCWA-yhteiskirjoitusvälineen dokumentin esitys

DCWA (Distributed Collaborative Writing Aid) on yhteiskirjoitusväline, joka esittää ryhmän kesken jaetun dokumentin hierarkkista loogista jakoa käyttäen (Chang *et al.*, 1995; Fouss, 1996). Kuvassa 32 on ruudunkaappaus DCWA-järjestelmästä, jossa näkyy, kuinka työstettävä dokumentti on jaettu loogisiin osiin. Osajako voi noudattaa vaikkapa dokumentin jakoa lukuihin kohtiin ja alakohtiin, kuten kuvassa 32.



Kuva 32: Ruudunkaappaus DCWA-järjestelmän dokumentin loogisesta näkymästä (Fouss, 1996)

Looginen näkymä toimii siten, että jokainen neliönä kuvattu dokumentin osa sisältää joko tekstiä tai uuden loogisen kuvauksen. Dokumentista muodostuu puurakenne, jonka lehtinä on tekstiä. Yhteiskirjoitusväline SEPIA (Haake ja Wilson, 1992; Streitz *et al.*, 1992) käyttää tässä alakohdassa kuvaillun kaltaista dokumentin kuvausta. Eräs esimerkki merkittävästä yhteiskirjoittamisen alueesta, jolla loogista näkymää voi hyödyntää, on ohjelmointi (Fouss, 1996). Looginen jako voi yhteiskirjoitetun C-kielisen ohjelman tapauksessa olla esimerkiksi puurakenne, jossa juurena on koko ohjelma, juuren lapsisolmuina moduulit ja lehtinä aliohjelmat.

### 4.3.2 Quiltin roolihierarkkia

Eräs tapa käsitellä alakohdassa 2.2.1 käsittelemiäni rooleja ja sosiaalisia protokollia yhteiskirjoitusvälineessä on niiden määrittäminen käyttäjän oikeuksien kautta. Käyttäjien oikeudet määritellään tyypillisesti sessiokohtaisesti (katso alakohdassa 2.2.3) siinä vaiheessa, kun sessiota käynnistetään. Quiltin (Fish *et al.*, 1988; Leland *et al.*, 1988) tapauksessa tehdään juuri näin. Sivuutan tässä alakohdassa kuitenkin sessionhallinnan ja keskityn roolien ja oikeuksien suhteeseen Quiltissa.

Quiltissa roolit ovat hierarkkisia, eli ylempänä hierarkiassa olevilla rooleilla on kaikki alempana olevien oikeudet muiden oikeuksien lisäksi (Fish *et al.*, 1988). Roolihierarkkia toimii siten, että lukija on alimpana, kommentoija keskellä ja kirjoittaja ylimpänä. Kommentoijalla on kaikki lukijan oikeudet, kirjoittajalla puolestaan kaikki lukijan ja kommentoijan oikeudet. Taulukossa 16 on esimerkki roolien ja oikeuksien suhteesta Quiltissa erilaisten kommenttien tapauksessa. Quiltissa on mahdollista perusdokumentin kirjoittamisen lisäksi käyttää erilaisia kommentteja, taulukkoon 16 olen valinnut kolme erilaista: kommentiksi tulkittava muokausehdotus, kaikille näkyvä julkinen kommentti ja tietyille käyttäjälle suunnattu viesti. Mahdolliset toimenpiteet perusdokumentin ja kommenttien suhteen ovat luominen, muokkaaminen, tuhoaminen sekä lukeminen. Taulukossa esiintyy edellä mainittujen roolien lisäksi kaksi yleisnimikettä, joilla voidaan tapauskohtaisesti tarkoittaa mielivaltaista roolia. Luoja merkitsee roolista riippumatta kysei-

sen kommentin luonutta käyttäjää ja vastaanottaja käyttäjää, jolle suunnattu viesti on osoitettu.

Taulukko 16: Quilt-yhteiskirjoitusvälineen roolihierarkkia, (Leland *et al.*, 1988).

	<b>Perusdoku- mentti</b>	<b>Muokkaus- ehdotus</b>	<b>Julkinen kom- mentti</b>	<b>Suunnattu viesti</b>
<b>Luo</b>	kirjoittaja	kirjoittaja	kommentoija	kommentoija
<b>Muokkaa</b>	kirjoittaja	luoja, kirjoitta- ja	luoja	luoja
<b>Tuhoa</b>	kirjoittaja	luoja, kirjoitta- ja	luoja	vastaanottaja, luoja
<b>Lue</b>	lukija	kirjoittaja	kommentoija	luoja, vastaa- nottaja

Taulukosta 16 käy esimerkiksi ilmi, että Quiltissa lukija-roolilla on hyvin vähän oikeuksia. Lukija saa ainoastaan lukea perusdokumenttia. Kommentoijalla on laajemmat oikeudet: hän saa luoda julkisia kommentteja ja osoittaa henkilökohtaisia suunnattuja viestejä kelle tahansa käyttäjälle. Kirjoittajat saavat tehdä kaikkea; he ovat muun muassa ainoat, joilla on oikeus käsitellä perusdokumenttia.

#### 4.4 Yhteenveto

Tässä luvussa olemme tutustuneet erilaisiin yhteiskirjoitusvälineisiin. PREP:in (katso kohta 4.1) ja Calliopen (katso kohta 4.2) tarkempi piirteiden erottelu näyttää, kuinka eri tavoilla yhteiskirjoittamisen tukeminen tietokoneitse on mahdollista. Asynkroniseen työskentelyyn ensisijaisesti suunniteltu PREP painottaa kommenttien näkyvyyttä ja helppoa luettavuutta (katso alakohta 2.3.2), kun taas Calliope painottaa reaaliaikaisessa työskentelyssä tarpeellista tietoisuusinformaatiota (katso kohta 2.4) ja monipuolista yhtenäisyyden hallintaa (katso kohta 3.2). Taulukossa 17 on PREP:in ja Calliopen piirteiden vertailu.

Taulukko 17: PREP:in ja Calliopen vertailu.

<b>Piirre</b>	<b>PREP</b>	<b>Calliope</b>
<b>Yhteinen dokumentti:</b>		
perustekstinkäsittely	Tukee alakohdassa 2.1.1 mainittuja piirteitä.	Tukee ilmeisesti pelkästään tekstin syöttämistä ja tallettamista.
käyttöliittymän kytkentä	Tiukka WYSIWIS, ajan ja populaation suhteen rentoutettu WYSIWIS.	Tiukka WYSIWIS ja populaation suhteen rentoutettu WYSIWIS.
versionhallinta	Versioiden joustava vertailu.	Ei versionhallinnan mekanismeja.
<b>Koordinointi:</b>		
kirjoitusprosessi	Tukee pehmeitä ja kovia aikarajoja.	Ei kirjoitusprosessin tukea.
kirjoitussessio	Avoin sessionhallinta.	Dokumenttiin perustuva sessiohallinta.
<b>Kommunikointi:</b>		
huomautukset ja kommentit	Sarakepohjainen kommentointi.	Kommentit omissa teksti-ikkunoissaan.
sanaton viestintä	Ei sanatonta viestintää.	Jaettu piirustusalue.
<b>Tietoisuus</b>	Sarakepohjainen käyttöliittymä.	Jaetut vierityspalkit, hahmonäkymä.
<b>Arkkitehtuuri</b>	Keskitetty C2.	Hybridi H3.
<b>Yhteinäisyyden hallinta</b>	Tarkemmin määrittämätön lukitsemismekanismit.	Optimistisen lukitsemisen ja optimistisen sarjoittamisen yhdistelmä.

Taulukon 17 jaottelu perustuu luvussa 2 käyttämäni jaotteluun, johon olen lisännyt luvusta 3 piirrekategoriat arkkitehtuuri ja yhtenäisyyden hallinta. Olen jättänyt taulukosta pois kommunikointiin liittyvän tekstipohjaisen keskustelun, koska

kumpikaan vertailemistani järjestelmistä ei tue sitä. Samoin olen sivuuttanut roolit, koska kumpikaan järjestelmä ei ota kantaa käyttäjien rooleihin suoraan. Järjestelmät olettavat kirjoitusryhmän hoitavan roolien hallinnan järjestelmän.

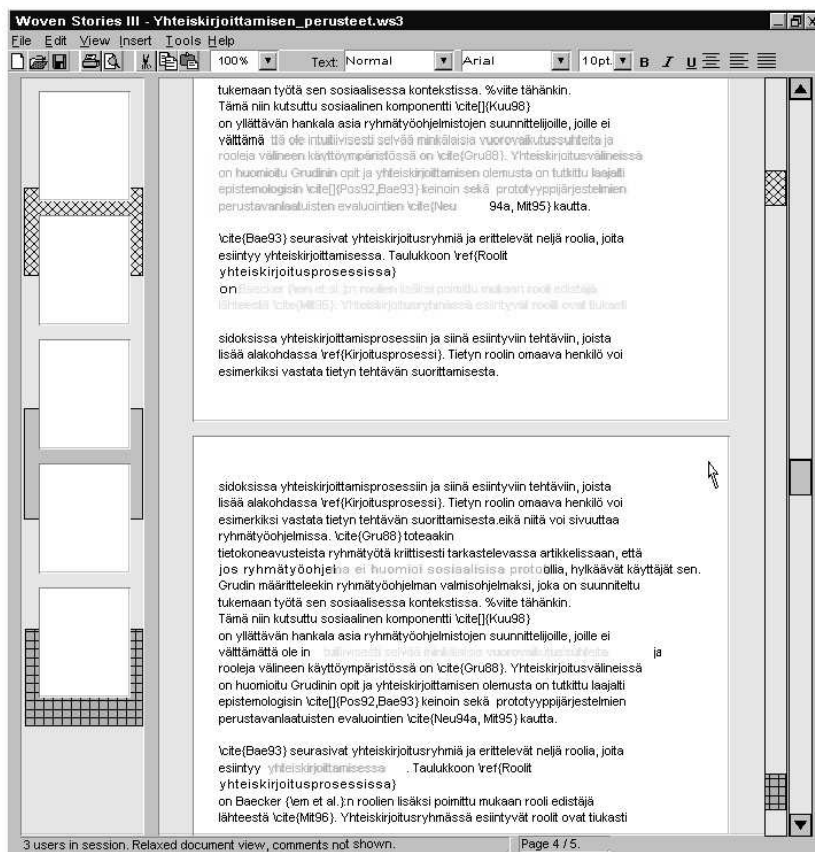
Kohdassa 4.3 käsittelen kahta mielenkiintoista piirrettä muista järjestelmistä. Dokumentin looginen esitys sen osien mukaisesti saattaa olla hyvin käyttökelpoinen piirre yhteiskirjoitusvälineessä (katso alakohta 4.3.1). Kirjoitusryhmän roolien määrittäminen oikeuksien kautta on eräs mahdollisuus huomioida alakohdassa 2.2.1 käsittelemäni roolit ja sosiaaliset protokollat yhteiskirjoitusvälineessä.

## **5 Tietokoneavusteisen yhteiskirjoittamisen tulevaisuus**

Tässä luvussa hahmottelen näkemystäni siitä, millainen seuraavan sukupolven yhteiskirjoitusväline voisi olla. Visioni perustuu tätä tutkielmaa kirjoittaessani tutki maani kirjallisuuteen sekä kokemuksiini toteuttaessani Woven Stories II yhteiskirjoitusvälinettä (Gerdt *et al.*, 2001). Kohdassa 5.1 esittelen ajatuksiani tulevaisuuden yhteiskirjoitusvälineen piirteistä luvusta 4 tutun jaottelun mukaisesti. Luvun loppuun, kohtaan 5.2 olen kerännyt sekä itse keksimiäni että kirjallisuudesta löytämiäni mielenkiintoisia aiheita jatkotutkimukselle yhteiskirjoittamisen parissa.

### **5.1 Tulevaisuuden yhteiskirjoitusväline**

Minkälainen on tulevaisuuden yhteiskirjoitusväline? On luultavasti turvallista sanoa, että se muistuttaa nykyisiä prototyyppisiä, joista kaikkein käyttökelpoisimmat piirteet elävät edelleen ja kehittyvät. Kuvassa 33 on näkemykseni siitä, miltä nykyteknologiaa hyödyntävä yhteiskirjoitusväline voisi näyttää.



Kuva 33: Hahmotelma yhteiskirjoitusvälineen käyttöliittymälle.

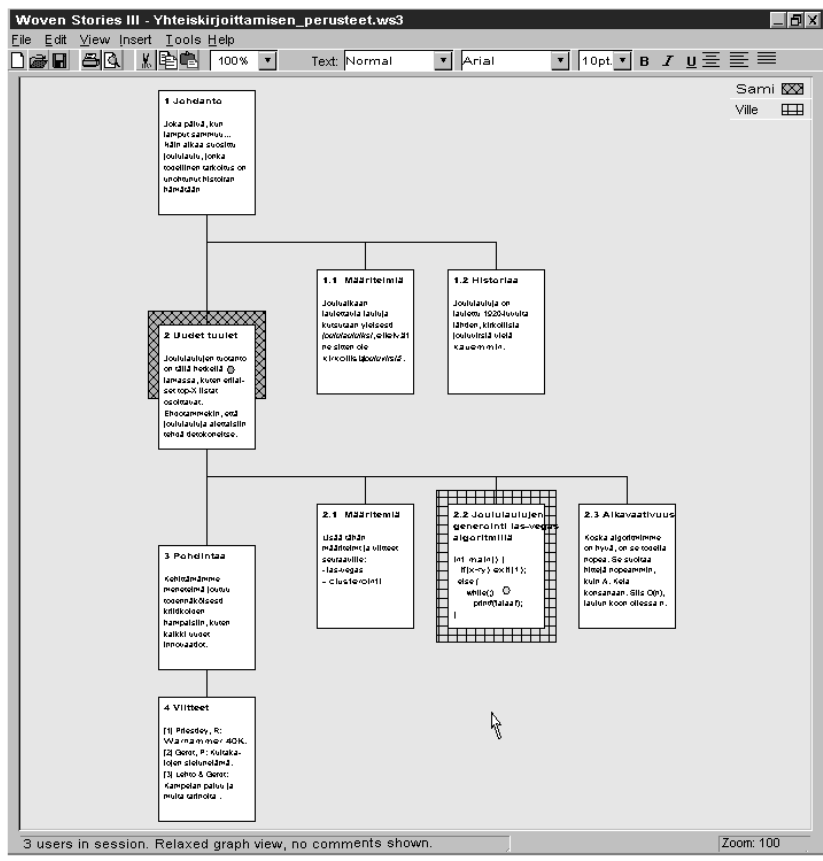
Kuvan 33 piirretyissä ruudunkaappauksessa on yhdistelty piirteitä useasta nykyisestä järjestelmästä. Esimerkiksi oikeassa laidassa näkyvät usean käyttäjän vierityspalkit (katso alakohtakohta 2.4.2) ovat samanlaiset kuin SASSE:ssa ja Callipessä. Vasemmassa laidassa näkyvä hahmonäkymä on versio esimerkiksi SASSE:ssa käytetystä vastaavasta. Tekstin väritys kirjoittajakohtaisten värien mukaan on myös käytössä Callipessä ja SASSE:ssa. Muut piirteet, kuten valikot ja WYSIWYG-dokumentti (katso alakohdat 2.1.1 ja 2.1.2), ovat tuttuja tavallisista yhden käyttäjän tekstinkäsittelyohjelmista.

Seuraavissa alakohdissa esittelen ajatuksiani tulevaisuuden yhteiskirjoitusvälineestä luvussa 4 käyttämäni piirteiden jaottelun mukaisesti järjestettynä.



### 5.1.1 Yhteinen dokumentti

**Perustekstinkäsittely.** Yhteiskirjoitusvälineen tulisi korvata tavallinen tekstieditori. Tästä seuraa, että sen on toimittava myös itsenäisenä tekstinkäsittelyohjelmalla ja näkymän dokumentista on oltava samanlainen kuin tavallisissa tekstieditoreissa. Yhteiskirjoitusväline ei saa säilyttää käyttäjiään ihmeellisillä piirteillä, vaan sen on johdateltava heidät yhteiskirjoittamisen piiriin tutulta pohjalta. Luonnollisesti yhteiskirjoitusväline voi tarjota useita dokumentin esitystapoja, mikä on jopa suotavaa, koska tavallisen tekstieditorin näkymä dokumentista ei sovellu kovin hyvin kaikkiin yhteiskirjoitustilanteisiin. Esimerkiksi tehtävien jako kirjoittajien kesken saattaa olla helpompaa, jos jaettavat yksiköt ovat dokumentin loogisia osia, kuten lukuja ja kappaleita. Kuvassa 34 on piirretty ruudunkaappaus yhteiskirjoitusvälineen dokumentin loogisesta näkymästä. Siinä näkyy koko dokumentti lukujen ja kohtien mukaan jaettuna. Näkymän kautta on mahdollista toteuttaa esimerkiksi toiminto, jolla tietyt osat dokumenttia annetaan tietyille tai tietyille käyttäjien muokattavaksi rajoittamalla muiden oikeuksia niiden suhteen. Oikeuksien jako käyttäjien kesken voidaan kuvata loogisessa näkymässä esimerkiksi värein tai erilaisin symbolein.



Kuva 34: Dokumentin looginen kuvaus yhteiskirjoitusvälineessä.

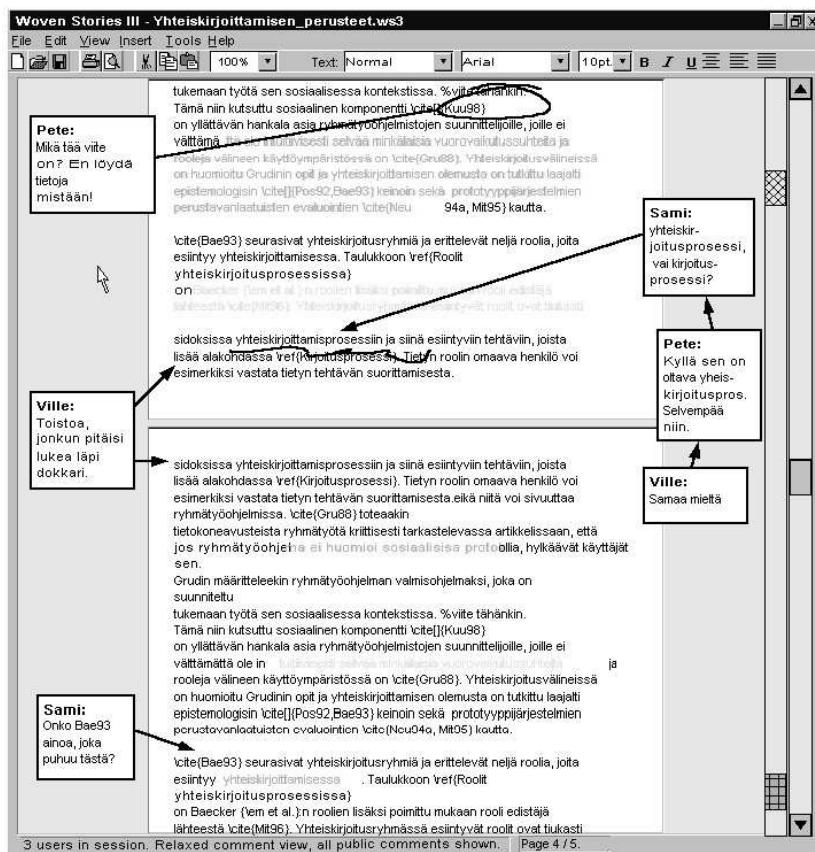
Kuvan 34 poikkeaa alakohdassa 4.3.1 esittelemästani DCWA-yhteiskirjoitusvälineen loogisesta näkymästä siten, että siinä kuvataan jokainen dokumentin osa atomisena, ei kokoelmana, joka voisi sisältää lisää osia. Tämä johtuu siitä, että uskon tulevaisuuden yhteiskirjoitusvälineen hyötyvän tarkennettavasta näkymästä, jossa käyttäjä voi siirtyä kauemmas dokumentista nähdäkseen koko dokumentin loogisen rakenteen. Löydettyään etsimänsä osan dokumentista käyttäjä voi tarkentaa näkymäänsä siirtymällä niin lähelle kyseistä osaa, että teksti näkyy hänen näytöllään sopivan kokoisena. Kuvatun kaltaista tarkennettavaa näkymää on käytetty KidPad-nimisessä lapsille tarkoitettussa tarinankerrontavälineessä, jossa se on ollut hyvin suosittu (Benford *et al.*, 2000).

**Versionhallinta.** Versionhallinta (katso alakohta 2.1.3) on eräs piirre, joka on hyvin käyttökelpoinen yhteiskirjoitusvälineessä. Pidän yhteiskirjoitusvälineen minimivaatimuksena versionhallinnan suhteen versioiden jäädyttämistä, versioiden vertailua ja eri versioiden selailua. Esimerkiksi asynkroninen työskentely vaikeutuu, jos käyttäjä ei voi vertailla, mitä on tapahtunut sen jälkeen, kun hän on viimeksi työstänyt dokumenttia.

### 5.1.2 Kommunikaatio

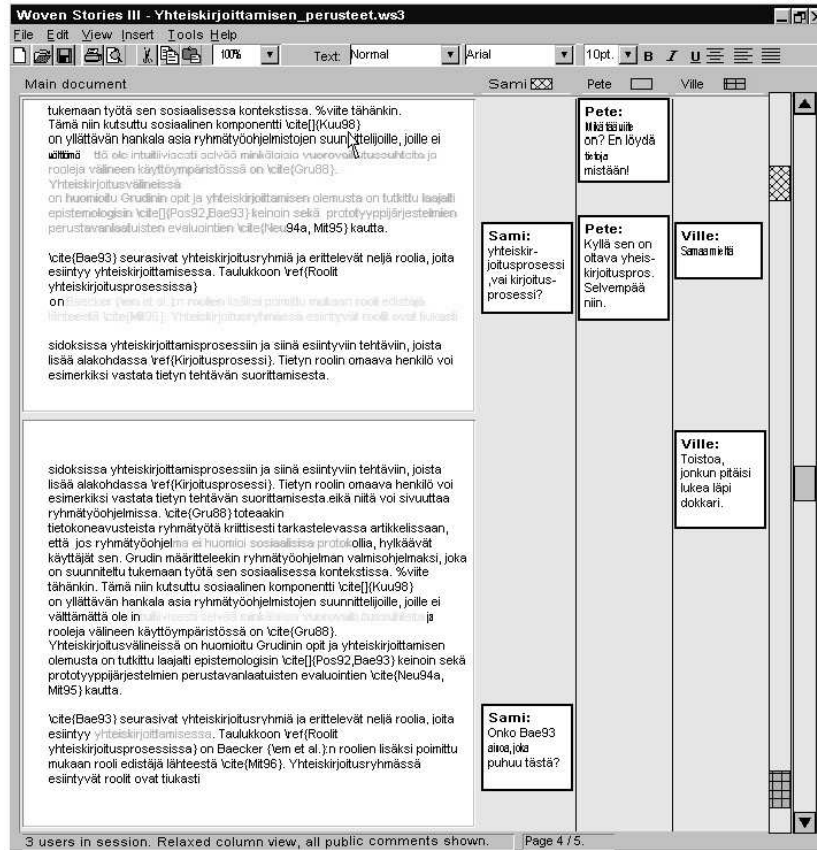
**Tekstipohjainen keskustelu osoitinvälineillä terästettynä.** Kaikista alakohdassa 2.3.1 käsittelemistäni tekstipohjaisen keskustelun epäkohdista huolimatta uskon sen muodostavan hyvän perustan kirjoittajien väliselle reaaliaikaiselle kommunikoinnille yhteiskirjoittamisvälineessä. Näkemystäni vahvistaa se tosiasia, että tekstipohjainen keskustelu on hyvin suosittua ja sitä käytetään lähes kaikissa viimeisen viiden vuoden aikana ilmestyneissä Internet-pohjaisissa moninpeleissä. Lisäksi keskusteluohjelma on hyvin helppo toteuttaa ja se vaatii hyvin vähän resursseja verrattuna muihin reaaliaikaisiin kommunikaatiokeinoihin, kuten äänen välittämiseen. Uskon, että monet edellä mainituista ongelmista voidaan kiertää käyttämällä osoitinvälineitä (katso alakohta 2.3.3) ja kommentointikeinoja tekstipohjaisen keskustelun yhteydessä.

**Huomautukset ja kommentit.** Mielestäni alakohdassa 2.3.2 käsittelemäni vaatimukset huomautuksille ja kommenteille ovat järkeviä. Kuvassa 35 on näkemystäni siitä, miltä vapaamuotoinen kommentointi voisi näyttää yhteiskirjoitusvälineessä. Siinä jokainen kommentti sisältää kommentoijan nimen ja ne on mahdollista sitoa tiettyyn kohtaan dokumentissa nuolilla ja vapaasti piirretyillä viivoilla. Vapaamuotoisen kommentoinnin salliminen saattaa johtaa melkoiseen sekasotkuun, joten on tärkeää, että käyttäjä voi halutessaan kätkeä kommentit.



Kuva 35: Vapaamuotoinen kommentointi yhteiskirjoitusvälineessä.

PREP-järjestelmän (katso kohta 4.1) ensisijaisesti huomautusten ja kommenttien käytön kannalta suunnitellun käyttöliittymän innoittamana laajennan visiotani kommentoinnin suhteen. Kuvassa 36 on piirretty ruudunkaappaus, jossa kuvassa 35 esitetty vapaamuotoinen kommentointi on järjestetty sarakkeittain. Jokaisella käyttäjällä on oma sarakkeensa, johon hän voi lisätä kommentteja. Sarakkeiden ulkopuolelle ei voi lisätä kommentteja, ainoastaan työstää dokumenttia. Ruudunkaappauksessa kommentteja ei ole kytketty niiden viittaamiin kohtiin nuolin ja viivoin selvyden vuoksi.



Kuva 36: Sarakkeittain jaoteltu kommentointi yhteiskirjoitusvälineessä.

Sarakeittain järjestetyn käyttöliittymän ongelmaksi voi muodostua tila. Sarakkeita on yksi jokaista käyttäjää varten, joten usean käyttäjän tapauksessa kaikki sarakkeet eivät saata mahtua ruudulle. Silloin yhteiskirjoitusvälineen on osattava näyttää käyttäjän valitsemat sarakkeet ja ilmaista samalla selvästi, että muitakin sarakkeita kommentteineen on saatavilla.

Calliopen ulkoisten kommenttien liittämismahdollisuus URL (Universal Resource Locator) -osoite määrittelemällä on mielestäni hyvä (katso alakohta 4.2.3). URL:ien käyttöä voisi laajentaa kattamaan myös dokumenttien osia, jolloin kuvassa 34 esitetyssä dokumentin loogisessa näkymässä osa loogisista osista voisi olla ulkoisia resursseja, kuten WWW-sivuja.

### 5.1.3 Koordinaatio

**Roolit, sosiaaliset protokollat ja sessionhallinta.** Quiltin (katso alakohta 4.3.2) suhtautuminen rooleihin on mielestäni hyvä. Roolit ovat yhteiskirjoitusvälineen kannalta kokoelmia oikeuksia yhteisen tuotoksen ja ryhmän hallinnan suhteen. Uskon varsinkin mekanismin, jolla voi määritellä uusia rooleja oikeuskokoelmia määrittelymekanismilla olevan käyttökelpoinen yhteiskirjoitusvälineessä. Roolien määrittelymekanismilla yhteiskirjoitusväline on mahdollista sopeuttaa eri tilanteisiin, esimerkiksi koulumaailman voimakkaan autoritäärisistä roolijaoista yliopistomaailman tasavertaisempiin kirjoitusrooleihin.

**Kirjoitusprosessi.** Kirjoitusprosessin ohjaileminen ilman aikarajoja on luullakseni mahdotonta. Tästä syystä pidän pehmeitä ja kovia aikarajoja minimivaatimuksena yhteiskirjoitusvälineen työkaluvalikoimassa kirjoitusprosessin suhteen. Laajentaisin kuitenkin PREP:issä ja Quiltissa käytettyjä aikarajoja (katso alakohta 2.2.2) laajemmiksi kokonaisuuksiksi, jotka nimeän aktiviteeteiksi. Aktiviteetti on kokoelma aikarajoja ja aikaan sidottuja oikeuksia. Yhteiskirjoitusvälineen käyttäjä voi aktiviteetin avulla määritellä kirjoitusprosessin osan tietyn tuotoksen suhteen. Esimerkiksi kirjoittajien luku- ja kirjoitusoikeudet voivat aueta tietyn dokumentin suhteen tiettyinä aikana ja päättyä tiettyyn aikarajaan. Kirjoittajien oikeuksiin liitetään kommentoijien oikeudet, heidän oikeutensa lisätä kommentteja aukeaa vaikkapa kolme viikkoa ennen kirjoittajien aikarajaa ja päättyy viikkoa ennen sitä. Näin syntyy runko kirjoitusprosessille, jossa kirjoittajilla on aluksi aikaa työstää tuotostaan, joka aukeaa kommentoijille nähtäväksi ja kommentoitavaksi tietyksi aikajaksoksi, jonka jälkeen kirjoittajilla on vielä aikaa muokata dokumenttia ennen lopullista prosessin päättymisaikaa.

### 5.1.4 Tietoisuus

Reaaliaikaisen informaation välittämiseen on olemassa useita käyttöliittymämekanismia, kuten monen käyttäjän vierityspakit ja hahmonäkymät (katso alakohta

ta 2.4.2), näitä olenkin sisällyttänyt visiooni. Kuvassa 34 näkyy hahmonäkymän versio, jolle sopii paremmin nimitys tutkanäkymä. Siinä kunkin käyttäjän näkymä näkyy hahmonäkymän kaltaisesti väritettynä neliönä. Väritetyn neliön koko saattaa kattaa jopa koko ruudun, jos käyttäjä tarkentaa näkymänsä siten, että se kattaa koko alueen.

Käyttäjän on pystyttävä säätämään tietoisuusinformaation määrää, koska työskentelyn intensiteetin vaihtelu on kirjoitusprosessille tyypillistä. Helpoin tapa antaa käyttäjälle mahdollisuus säätää tietoisuusinformaatiota on antaa käyttäjän määrittellä mitä tietoisuusinformaatiokomponentteja hän näkee. Esimerkiksi kuvassa 33 näkyvän hahmonäkymän voi piilottaa, kuten kuvissa 35 ja 36 on tehty, jolloin käyttöliittymä on rauhallisempi. Erilaisten tietoisuuskomponenttien näkyvyyden säätelyn mahdollisuus on tärkeää siksi, että yhteiskirjoitusvälinettä on voitava käyttää myös tavallisena tekstieditorina. Tietoisuuskomponentit ovat täysin turhia, ellei järjestelmää käytä kuin yksi henkilö.

Alakohdassa 4.1.2 käsittelemäni PREP-järjestelmän vuorovaikutusparametrit antavat käyttäjän määrittellä mitä muut heistä näkevät ja milloin. Suhtaudun epäilevästi siihen, että käyttäjälle annetaan valta päättää mitä informaatiota järjestelmä hänestä välittää muille välittää. Toisaalta joissakin tilanteissa emme kerro kaikkea ryhmätyöskentelyn aikana, joten yhteiskirjoitusvälineeseen saattaa olla hyvä sisällyttää mahdollisuus myöntää käyttäjille oikeus säädellä mitä tietoa muut heistä saavat. Mielestäni on tärkeää, että kaikille järjestelmän käyttäjille on selvää, missä määrin he saavat tietoa muista, jos yksittäiset käyttäjät kerran voivat sitä säädellä. Näin käyttäjien on helpompi tulkita sitä informaatiota mitä he saavat järjestelmän kautta.

### **5.1.5 Järjestelmäarkkitehtuuri ja yhtenäisyyden hallinta**

**Arkkitehtuuriratkaisu.** Nykyisissä Internet-pohjaisissa peleissä käytetään usein täysin hajautettuja ja toistettuja arkkitehtuuriratkaisuja (katso alakohta 3.1.2). Toistettuja järjestelmiä käytetään usein asymmetrisesti, jolloin yksi toisoinnoista

toimii omistautuneena palvelimena, jonne esimerkiksi toisintojen kesken jaetun pelitilanteen koordinointi keskitetään (katso alakohta 3.1.3). Uskon tämän kaltaisen arkkitehtuuriratkaisun toimivan hyvin myös yhteiskirjoitusjärjestelmässä. Se on useissa peliohjelmissa hyväksi havaittu ratkaisu, joka ei myöskään ole liian vaikeakäyttöinen peruskäyttäjän kannalta katsottuna. Ratkaisun etuna on toisinnon itsenäisen toiminnan mahdollisuus ja erittäin dynaaminen verkottuminen muiden toisintojen kanssa. Haittapuolena on ratkaisun monimutkaisuus suunnittelun ja toteutuksen kannalta katsottuna.

Eräs toinen käyttökelpoinen ominaisuus Internet-pohjaisten moninpelien maailmassa on keskitetyt osoitepalvelimet välityspalveluineen. Nämä toimivat siten, että pelikumppaneita etsivä pelaaja ilmoittaa itsensä keskitetylle WWW-palvelimelle. Käyttäjä voi ohjata peliohjelman etsimään pelikumppania WWW-palvelimelta. Kun pelikaveri löytyy, voi käyttäjä kutsua ilmoittautuneen pelaajan pelisessioon mukaan. Pelisession alettua ei keskitettyä WWW-ilmoituspalvelua enää tarvita. Etuna mekanismeissa on peliohjelmien riippumattomuus WWW-palvelimista, ne ovat vain lisäpiirre, jota on mahdollista käyttää. Käyttäjä voi aina ottaa suoraan yhteyttä tuntemaansa pelikumppaniin. Peliohjelma voi myös etsiä mahdollisia pelikumppaneita useilta eri palvelimilta. GroupKit (katso alakohta 3.3.3) käyttää kuvattua kaltaista mekanismia mutta rajoitetummassa mittakaavassa, lisäksi se on riippuvainen keskitetyn välitysmekanismien toiminnasta. Uskon yhteiskirjoitusvälineiden hyötyvän pelikumppanin välityspalvelun kaltaisesta mekanismista, jonka avulla nimenomaan session käynnistäminen ja löytäminen helpottuu.

**Yhtenäisyyden hallinta.** Jos järjestelmä rakentuu täysin toistetun arkkitehtuuriratkaisun varaan, on alakohdassa 3.2.3 operationaalinen transformaatio oikea valinta yhtenäisyyden hallinnalle. Operationaalinen transformaatio toimii hyvin verkkoympäristössä, koska sen vikasietoisuus on hyvä. Sen huonona puolena on hankala suunnittelu ja toteutus. Käyttäjän kannalta saattaa olla hyvä, jos järjestelmä tukee lukitsemista. Lukitseminen voi toimia oikeuksien ja versionhallinnan ohella eräänä keinona hallita dokumenttia, esimerkiksi käyttäjä voi lukita tietyn



kohdan dokumentista viikoksi lähtiessään Kamerunin kiertomatkalle. Käyttäjä voi rauhassa kirjoitella lukitsemaansa osaa lentokoneessa vaikka ei olekaan yhteydessä muihin käyttäjiin. Palatessaan matkalta käyttäjä lisää yhteiskirjoitusvälineeseen lukitsemaansa kohtaa varten kirjoittamansa tekstin ja poistaa lukituksen.

Eräs keskeinen piirre tulevaisuuden yhteiskirjoitusvälineessä on parempi yhteensopivuus muiden järjestelmien kanssa. Käsittelen ajatuksiani yhteensopivuuden kehittämistä XML-pohjaisilla standardeilla seuraavassa kohdassa, koska se vaatii toteutuakseen jatkotutkimuksia.

## **5.2 Tutkimuskohteita tietokoneavusteisen yhteiskirjoittamisen parissa**

Tässä kohdassa tuon esille joitakin kirjallisuudesta löytämiäni ja itse keksimiäni jatkotutkimuskohteita yhteiskirjoittamisen piiristä. Osa aiheista on yleistettävissä koskemaan tietokoneavusteista ryhmätyötä kaikkine sovelluskohteineen. En pyri listaamaan kaikkea, mikä on vielä tutkimatta yhteiskirjoittamisen parissa, vaan tuon esille asioita, jotka erityisesti ovat askarruttaneet mieltäni tätä tutkielmaa kirjoittaessani.

Galegherin ja Krautin (1992) mukaan informaatioteknologiavälineet ovat sosiaalisia objekteja, joilla on määrätty materiaaliset piirteet, mutta joilla on määrittelemättömät tai epämääräiset sosiaaliset vaikutukset ja seuraukset. Yhteiskirjoitusvälineiden käyttökokemuksia ei ole juurikaan tutkittu tieteellisesti. Tästä johdetaan myös tietokoneavusteisen yhteiskirjoittamisen mahdolliset sovelluskohteet ovat lähinnä hyvin perusteltujen arvailujen varassa. Olemassaolevat tilanteet, joissa dokumentteja tuotetaan ryhmässä, ovat luonnollisia käyttökohteita. Yhteiskirjoitusvälineiden suunnittelijoiden on kuitenkin tutkittava tietyssä ryhmätilanteessa vaikuttavia voimia voidakseen tukea sitä tietokoneitse.

Hofte (1996) mainitsee jatkotutkimusta kaivattavan erityisesti sekä asynkronista että synkronista työskentelyä sisältävien prosessien mallintamisessa. Useat ny-

kyisistä yhteiskirjoitusvälineistä tukevat molempia työmuotoja. Synkronista ja asynkronista tietokonepohjaista työskentelyä tukevien kirjoitusprosessien mallien puute vaikeuttaa oikeanlaisten yhteiskirjoitusvälineiden suunnittelua. Nykyiset järjestelmät ovat tässä suhteessa melko alkeellisella tasolla, vaikka useat niistä tukevatkin molempia työmuotoja.

Grudin (1988) painottaa, että ryhmätyötä tukevia järjestelmiä on vaikeaa suunnitella, koska suunnittelijoiden on hyvin vaikeaa, ellei mahdotonta, ymmärtää intuitiivisesti ryhmätyötilannetta. Suunnittelua ei suinkaan helpota se, että ryhmätyöohjelmien evaluointi on hyvin vaikeaa. Suunnittelijoiden on vaikea oppia tekemisistään virheistä, elleivät he löydä niitä (Grudin, 1988). Uskon parhaaseen tulokseen päästävän iteratiivisella kehitysprosessilla, jossa suunnittelijat kehittävät yhteiskirjoitusvälineen, antavat sen loppukäyttäjille kokeiltavaksi ja parantavat välinettä palautteen ja muiden huomioiden mukaisesti.

Mielestäni Gutwinin ja Greenbergin (Gutwin ja Greenberg, 1996) työtilatietoisuutta pitää tarkentaa, koska esimerkiksi opetustilanteessa esiintyy erilaisia tietoisuusasioita kuin siinä on kuvattu. Myös tietoisuuden säätö on mielenkiintoinen tutkimusaihe. Pitäisikö käyttäjän saada säätää sitä informaatiota, mitä hänestä muille välitetään? Synkronisia tietoisuutta välittäviä käyttöliittymäkomponentteja on tutkittu ja toteutettu laajalti mutta asynkronisia vastaavia sen sijaan hyvin vähän. Asynkronisen tietoisuuden tukemisen tutkiminen on kytköksissä aiemmin mainitsemaani kirjoitusprosessin tutkimiseen. Minkälaista tietoa yhteiskirjoitusvälineen käyttäjä, joka palaa matkoiltaan töihin kaipaa ja kuinka tieto on järkevää esittää? Eräs yhteiskirjoitusvälineen keino saattaa käyttäjä ajan tasalle saattaisi olla erilaisten raporttien generointi tietyistä tapahtumista tietyltä ajalta.

Operationaalinen transformaatio on tällä hetkellä uusinta uutta yhtenäisyyden hallinnassa. Sun ja Ellis (1998) sanovat, että sen parissa riittää tutkittavaa vuosikymmeniksi. Tällä hetkellä operationaalista transformaatiota hyödynnetään vain parissa prototyyppijärjestelmässä. Laajempaa käyttöä kokemuksineen kaivataan, jotta mekanismin todellisia ulottuvuuksia voidaan havainnoida.

Mielestäni eräs merkittävimmistä puutteista yhteiskirjoitusvälineissä on yhtenäisen dokumenttistandardin puute. Erilaisia yhteiskirjoitusvälineitä on olemassa yli 50 (CSCW Pages @ Technische Universität München, 2001), niistä suurin osa on keskenään yhteensopimattomia. Suurin osa pystyy tallettamaan dokumentin pelkkänä tekstinä tai esimerkiksi RTF-tiedostomuodon kaltaisessa yhden käyttäjän tekstinkäsittelyohjelmiin tarkoitetuissa yleistalletusmuodossa. Näihin talletusmuotoihin ei sisälly minkäänlaista tietoa esimerkiksi useasta kirjoittajasta, heidän osuuksistaan tuotetusta tekstistä, tekstin osuuksien lukituksesta, käyttäjien oikeuksista dokumentin suhteen, mahdollisista aikarajoista tai tekstin versiohistoriasta. Yhteiskirjoitettu tuotos sisältää kaikkea tätä informaatiota, joka on oleellista tuotoksen käsittelemisessä yhteiskirjoitusvälineessä.

Esitän edellä mainitsemani ongelman ratkaisuksi yleisen yhteiskirjoitustiedostomuodon standardin kehittämistä. Standardi voisi rakentua XML-kielen varaan, jolloin kaikki eri laitteistoalustoilla toimivat järjestelmät pystyisivät käsittelemään sitä. XML-dokumenttia on myös mahdollista lukea useimmilla WWW-selaimilla, jolloin sen sisältämää informaatiota voi tutkia myös ilman varsinaisia yhteiskirjoitusvälineitä. XML-pohjainen standardi rakentuisi yhteiskirjoitusvälineistä riippumattomien asioiden varaan, joita esiintyy yhteiskirjoittamisprosessissa (katso edellistä tekstikappaletta). XML-talletusmuodolle on mahdollista helposti toteuttaa erilaisia filteriohjelmiä, jotka muuttavat talletusmuodon tietyn järjestelmän ymmärtämään muotoon.

XML:n käyttöä yhteiskirjoitusvälineissä voisi myös laajentaa dokumentin talletusmuodon ulkopuolelle. Yhteiskirjoitusvälineissä voisi olla standardoitu kommunikaatorajapinta, jolloin kaksi eri ryhmäeditoria voisi toimia ainakin rajoitetusti yhdessä. Kommunikaatorajapinta kattaisi yhteiskirjoitusvälineiden perustapahtumat, kuten session hallintaan, dokumentin muokkaukseen ja yhtenäisyyden hallintaan sekä tietoisuusinformaation liittyvät tapahtumat. Esimerkiksi kohdassa 3.3 käsittelemäni kehitystyökalut voisivat sisältää standardin kommunikaatorajapinnan, jolloin niillä toteutetut järjestelmät voisivat kommunikoida keskenään.

## 6 Loppupäätelmät

Tässä tutkielmassa olemme tutustuneet tietokoneavusteiseen yhteiskirjoittamiseen kolmesta eri näkökulmasta. Luvussa 2 pyrin tuomaan esille keskeisiä yhteiskirjoittamisen piirteitä ja mahdollisuuksia tukea niitä tietokoneitse. Luku 2 edustaa tutkielmassani suunnittelijan näkökulmaa kartoittamalla yhteiskirjoittamista tietojenkäsittelytieteen sovellusalueena. Luvussa 3 tutustuimme keskeisiin kysymyksiin yhteiskirjoitusvälineen toteuttamisessa. Tarkasteltuamme yhteiskirjoittamista toteutusnäkökulmasta tutustuimme kahteen erilaiseen yhteiskirjoitusvälineeseen luvussa 4, jossa tavoitteenani oli sitoa aiempien lukujen teoreettiset ja toteutukselliset näkökulmat esittelemällä erilaisia tulkintoja niistä.

Tutustuttuani tietokoneavusteiseen yhteiskirjoittamiseen uskon, että sillä tulee olemaan kasvava merkitys yhteiskunnassamme. Yhä useammassa työssä tarvitaan yhteiskirjoitusvälineitä ja nopeasti lisääntyvä etäopetus tarvitsee nimenomaan keinoja dokumenttien tuottamiseen ryhmätyönä. Alalla tarvitaan kuitenkin lisää perustutkimusta ennen kuin tietokoneavusteisesta yhteiskirjoittamisesta kehittyy haastaja sähköpostin ja tavallisen tekstinkäsittelyohjelman muodostamalle kilpailijalle.

Tietokoneavusteinen yhteiskirjoittaminen on suhteellisen uusi tietojenkäsittelytieteen sovellusala. Huomasin tämän yrittäessäni löytää selkeää yleiskuvausta tietokoneavusteisesta yhteiskirjoittamisesta. Useimmat löytämistäni kuvauksista kattoivat ainoastaan kapean alan, kuten yhtenäisyyden hallinnan tai tietyn järjestelmän toteuttamisen, liittämättä niitä tarkemmin laajempaan kokonaisuuteen.

Mielestäni juuri kokonaiskuvan puute hidastaa tietokoneavusteisen yhteiskirjoittamisen kehittymistä yleisesti hyväksytyksi tietojenkäsittelyn sovellusalueeksi. Esimerkiksi kokonaiskuvan tiedostamattomuudesta sopii mielestäni hyvin yli viidenkymmenen toteutetun yhteiskirjoitusvälineen lähes täydellinen yhteensopimattomuus. Ei ole olemassa minkäänlaista yleisesti hyväksyttyä standardia yhteiskirjoitusvälineellä tuotetulle dokumentille. Yhteiskirjoitetun dokumentin tal-

lettaminen pelkkänä tekstinä kadottaa oleellista tietoa esimerkiksi kirjoittajista ja versiohistoriasta. Myös runsaslukuiset kehitystyökalut ovat yhteensopimattomia, mikä johtaa siihen, että tietyllä kehitystyökalulla toteutettu järjestelmä ei voi olla yhteydessä toisella kehitystyökalulla toteutettuun järjestelmään.

Suurin haasteeni tätä tutkielmaa kirjoittaessani oli tietokoneavusteisen yhteiskirjoittamisen määrittäminen pro gradu -tutkielmaa vastaavassa laajuudessa. Katson onnistuneeni melko hyvin muodostaessani yleiskuvaa tietokoneavusteisesta yhteiskirjoittamisesta tämän tutkielman puitteissa. Valitsemani kolme näkökulmaa tukevat toisiaan muodostaen suhteellisen kattavan yleiskuvan. Jos kirjoittaisin tutkielmani nyt uudestaan, saattaisin kuitenkin tinkiä näkökulmien suhteen. Jokaisesta näkökulmasta voisi kirjoittaa erillisen pro gradu -tutkielman, jolloin niistä muodostuisi syvällisempiä katsauksia tietokoneavusteisen yhteiskirjoittamisen saloihin.

Usean eri näkökulman sisällyttäminen tutkielmaani on johtanut joidenkin aiheitten pinnalliseen käsittelyyn, koska laajaa yleiskuvaa muodostaessani ei ollut aikaa tutustua niihin tarpeeksi syvällisesti. Mielestäni esimerkiksi kohta 3.3 kaipaa tarkennusta ja luku 4 hyödyntäisi lukijaa nykyistä enemmän, jos siinä olisi kunnollinen usean yhteiskirjoitusvälineen evaluointi. Tietokoneavusteisen yhteiskirjoittamisen tutkimista vaikeuttaa muun kirjallisuuden kuin konferenssijulkaisujen puute. Lähes kaikki materiaali, jota olen käyttänyt tämän tutkielman lähteenä, on konferenssijulkaisuja tai niistä koottuja kokoelmateoksia. Olen kuitenkin tyytyväinen lopputulokseen, koska sen kirjoittaminen on auttanut minua ymmärtämään tietokoneavusteisen yhteiskirjoittamisen keskeisiä kysymyksiä. Toivon että tutkielmani palvelee samoin myös lukijoitani.

## Viitteet

- R. Baecker, G. Glass, A. Mitchell ja I. Posner. SASSE: The collaborative editor. Teoksessa *Proceedings of the CHI '94 conference companion on Human factors in computing systems*, sivut 459–462. 1994.
- R. M. Baecker, D. Nastos, I. R. Posner ja K. L. Mawby. The user-centered iterative design of collaborative writing software. Teoksessa *Conference proceedings on Human factors in computing systems*, sivut 399–405. 1993.
- L. J. Bannon ja K. Schmidt. CSCW: Four characters in search of a context. Teoksessa J. Bowers ja S. Benford, toimittajat, *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, sivut 3–16. Amsterdam North-Holland, 1991.
- J. M. A. Begole. *Flexible Collaboration Transparency: Supporting Worker Independence in Replicated Application-Sharing Systems*. väitöskirja, Department of Computer Science, Virginia Polytechnic Institute and State University, 1998.
- S. Benford, B. B. Bederson, K. Akesson, V. Bayon, A. Druin, P. Hansson, J. P. Hourcade, R. Ingram, H. Neale, C. O'Malley, K. T. Simsarian, D. Stanton, Y. Sundblad ja G. Taxén. Designing storytelling technologies to encouraging collaboration between young children. Teoksessa *Proceedings of the CHI 2000 conference on Human factors in computing systems*, sivut 556–563, The Hague, The Netherlands. 2000.
- T. Cavalier, R. Chandhok, D. Kaufer, J. Morris ja C. Neuwirth. A visual design for collaborative work: columns for commenting and annotation. Teoksessa *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*, sivut 729–738. 1991.
- K. H. Chang, Y. Gong, T. Dollar, S. Gajiwala, B. Lee ja A. W. Wear. On computer supported collaborative writing tools for distributed environments. Teoksessa *Proceedings of the 1995 ACM 23rd annual conference on Computer science conference*, sivut 222–229. 1995.

- E. F. Churchill, J. Trevor, S. Bly, L. Nelson ja D. Cubranic. Anchored conversations: chatting in the context of a document. Teoksessa *Proceedings of the CHI 2000 conference on Human factors in computing systems*, sivut 454 – 461. 2000.
- S. Coutouriaux, J. Hise, M. Ioannides, B. Mehlenbacher, M. Mitchell, B. Pollak, G. Wilson, K. Lindholm ja M. Shult. *The PREP Editor documentation*. The PREP Editor Project of Carnegie Mellon University, 1993. Sisältyy osoitteessa <http://eserver.org/software/prep/> saatavilla olevaan PREP-järjestelmään.
- CSCW Pages @ Technische Universität München. Internet WWW-sivu, url.: <http://www.telekooperation.de/cscw/>, 2001. (10.9.2001).
- P. Dewan. Multiuser architectures. Teoksessa *Engineering for HCI :Proceedings of IFIP WG2.7 Working Conference on Engineering for Human-Computer Interaction*, sivut 247–270. 1996.
- P. Dewan ja R. Choudhary. A high-level and flexible framework for implementing multiuser user interfaces. *ACM Transactions on Information Systems*, **10**, no. 4, 345–380, 1992.
- P. Dewan ja J. Munson. The role of version control in CSCW applications: A position statement. Teoksessa *Proceedings of the ECSCW*, sivut 33–36. 1995.
- P. Dourish. A divergence-based model of synchrony and distribution in collaborative systems. Tekninen Raportti EPC-1994-102, Rank Xerox Research Centre, Cambridge, Iso-Britannia, 1994.
- P. Dourish ja V. Bellotti. Awareness and coordination in shared workspaces. Teoksessa *Conference proceedings on Computer-supported cooperative work*, sivut 107–114. 1992.
- C. A. Ellis, S. J. Gibbs ja G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, **34**, no. 1, 39–58, 1991.

- R. S. Fish, R. E. Kraut ja M. D. P. Leland. Quilt: a collaborative tool for cooperative writing. Teoksessa *Proceedings of Conference Sponsored by ACM SIGOIS and IEEECS TC-OA on Office information systems (COIS'88)*, sivut 30–37. 1988.
- J. D. Fouss. Computer-aided software engineering in a computer supported cooperative work environment. Tekninen Raportti 96-13, Auburn University, 1996.
- R. S. Fussell, R. E. Kraut, F. J. Lerch, W. L. Scherlis, M. M. McNally ja J. J. Cadiz. Coordination, overload and team performance: effects of team communication strategies. Teoksessa *Proceedings of the ACM 1998 conference on Computer supported cooperative work*, sivut 275–284. 1998.
- J. Galegher ja E. R. Kraut. Computer-mediated communication and collaborative writing: media influence and adaptation to communication constraints. Teoksessa *Conference proceedings on Computer-supported cooperative work*, sivut 155–162. 1992.
- P. Gerdt. Woven Stories 2 documentation index. <http://cs.joensuu.fi/ws2/docs/>, 2001. (13.7.2001).
- P. Gerdt, P. Kommers, C. Looi ja E. Sutinen. Woven stories as a cognitive tool. Teoksessa *Cognitive Technology 2001*, osa 2117 sarjasta *LNAI*, sivut 233–247. Springer-Verlag, 2001.
- S. Greenberg, C. Gutwin ja M. Roseman. Semantic telepointers for groupware. Teoksessa *Proceedings of OzCHI '96 Sixth Australian Conference on Computer-Human Interaction*, Hamilton, New Zealand. 1996.
- S. Greenberg ja D. Marwood. Real time groupware as a distributed system: concurrency control and its effect on the interface. Teoksessa *Proceedings of the conference on Computer supported cooperative work*, sivut 207–217. 1994.
- S. Greenberg ja M. Roseman. Using a room metaphor to ease transitions in groupware. Tutkimusraportti 98/611/02, Department of



- Computer Science, University of Calgary. Internet WWW-sivu, URL: <http://cpsc.ucalgary.ca/grouplab/papers/>, 1998. (5.11.2001).
- S. Greenberg ja M. Roseman. Groupware toolkits for synchronous work. Teoksessa M. Beaudouin-Lafon, toimittaja, *Computer-Supported Cooperative Work*, luku 6, sivut 135–168. John Wiley & Sons Ltd, 1999.
- J. Grudin. Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. Teoksessa *Proceedings of the conference on Computer-supported cooperative work*, sivut 85–93. 1988.
- J. Grudin. Computer-supported cooperative work: history and focus. *Computer*, **27**, no. 5, 19–26, 1994.
- C. Gutwin ja S. Greenberg. Workspace awareness for groupware. Teoksessa *Proceedings of the CHI '96 conference companion on Human factors in computing systems: common ground*, sivut 208–209. 1996.
- C. Gutwin ja S. Greenberg. A framework of awareness for small groups in shared-workspace groupware. Tekninen Raportti 99-1, Department of Computer Science, University of Saskatchewan, Canada, 1999.
- C. Gutwin, G. Stark ja S. Greenberg. Support for workspace awareness in educational groupware. Teoksessa *Proceedings of the Conference on Computer Supported Collaborative Learning*, sivut 147–156. 1995.
- M. Guzdial, J. Rick ja B. Kerimbaev. Recognizing and supporting roles in CSCW. Teoksessa *Proceeding on the ACM 2000 Conference on Computer supported cooperative work*, sivut 261–268. 2001.
- A. Haake ja J. M. Haake. Take coVer: exploiting version support in cooperative systems. Teoksessa *Proceedings of the conference on Human factors in computing systems*, sivut 406–413. 1993.

- J. M. Haake ja B. Wilson. Supporting collaborative writing of hyperdocuments in SEPIA. Teoksessa *Conference proceedings on Computer-supported cooperative work*, sivut 138–146. 1992.
- H. Hofte, ter. Generic service features of CSCW applications. Tekninen Raportti RS/96002, Telematics Research Centre, 1996.
- H. Hofte, ter. *Working Apart Together. Foundations for Component Groupware*. Universal Press, Veenendaal, The Netherlands, 1998.
- ICQ Incorporated. Internet WWW-sivu, URL: <http://www.icq.com>, 2001. (24.7.2001).
- Id Software Corporation. Kotisivu. Internet WWW-sivu, URL:<http://www.idsoftware.com>, 2001. (13.7.2001).
- C. Jensen, S. D. Farnham, S. M. Drucker ja P. Kollock. The effect of communication modality on cooperation in online environments. Teoksessa *Proceedings of the CHI 2000 conference on Human factors in computing systems*, sivut 470–477, The Hague Netherlands. 2000.
- M. Koch. Design issues and model for a distributed multi-user editor. *The International Journal of Computer Supported Cooperative Work*, **3**, no. 4, 359–378, 1995.
- K. Kuutti. CSCW-kurssin luentomoniste. Helsingin yliopiston Tietojenkäsittelytieteen laitos. Internet WWW-sivu, URL: <http://www.cs.hut.fi/tta/courses/CSCW-syksy-98.PDF>, 1998. (24.4.2001).
- A. LaMarca, W. K. Edwards, P. Dourish, J. Lamping, I. Smith ja J. Thornton. Taking the work out of workflow: Mechanisms for document-centered collaboration. Teoksessa *Proceedings of the Sixth European Conference on Computer Supported Cooperative Work*, sivut 1–20, Copenhagen. 1999.
- B. G. Lee, K. H. Chang ja N. H. Narayanan. An integrated approach to version control management in computer supported collaborative writing. Teoksessa

*Proceedings of the 36th annual conference on Southeast regional conference*, sivut 34–43. 1998.

M. D. P. Leland, R. S. Fish ja R. E. Kraut. Collaborative document production using Quilt. Teoksessa *Proceedings of the conference on Computer-supported cooperative work*, sivut 206–215. 1988.

Medieval Manuscript Manual. Typology of medieval books. Internet WWW-sivu, URL: <http://www.ceu.hu/medstud/manual/MMM/home.html>, 2001. (4.10.2001).

A. Mitchell. *Communication and shared understanding in collaborative writing*. pro gradu, Graduate Department of Computer Science, University of Toronto, 1996.

A. Mitchell, I. Posner ja R. Baecker. Learning to write together using groupware. Teoksessa *Conference proceedings on Human factors in computing systems*, sivut 288–295. 1995.

J. Munson ja P. Dewan. A flexible object merging framework. Teoksessa *Proceedings of the conference on Computer supported cooperative work*, sivut 231–242. 1994.

C. M. Neuwirth, R. Chandhok, D. Charney, P. Wojahn ja L. Kim. Distributed collaborative writing: a comparison of spoken and written modalities for reviewing and revising documents. Teoksessa *Conference proceedings on Human factors in computing systems: "celebrating interdependence"*, sivut 51–57. 1994a.

C. M. Neuwirth, R. Chandhok, D. S. Kaufer, P. Erion, J. H. Morris ja D. Miller. Flexible diff-ing in a collaborative writing system. Teoksessa *Conference proceedings on Computer-supported cooperative work*, sivut 147 – 154. 1992.

C. M. Neuwirth, D. S. Kaufer, R. Chandhok ja J. H. Morris. Issues in the design of computer support for co-authoring and commenting. Teoksessa *Proceedings of the conference on Computer-supported cooperative work*, sivut 183–195. 1990.

- C. M. Neuwirth, D. S. Kaufer, R. Chandhok ja J. H. Morris. Computer support for distributed collaborative writing: Defining parameters of interaction. Teoksessa *Proceedings of the conference on Computer supported cooperative work*, sivut 145–152. 1994b.
- C. M. Neuwirth, D. S. Kaufer, R. Chandhok ja J. H. Morris. Computer support for distributed collaborative writing: A coordination science perspective. Teoksessa G. M. Olson, T. W. Malone ja J. B. Smith, toimittajat, *Coordination Theory and Collaboration Technology*, Computers, Cognition, and Work. Lawrence Erlbaum Assoc, 2001.
- J. F. Patterson. A taxonomy of architectures for synchronous groupware applications. *ACM SIGOIS Bulletin, Special issue: workshop write-ups and positions papers from CSCW'94*, **15**, no. 3, 27 – 29, 1995.
- J. F. Patterson, R. D. Hill, S. L. Rohall ja S. W. Meeks. Rendezvous: an architecture for synchronous multi-user applications. Teoksessa *Proceedings of the conference on Computer-supported cooperative work*, sivut 317–328. 1990.
- P. Persson. On the integration of text editing and version control. Teoksessa *Proceedings of NWPER '98, Eighth Nordic Workshop on Programming Environment Research*, sivut 209–219. 1998.
- I. R. Posner ja R. M. Baecker. How people write together. Teoksessa *Proceedings of the Twenty-Fifth International Conference on the System Sciences*, sivut 127–138. 1992.
- V. S. Rao, P. L. McLeod ja K. M. Beard. Adoption patterns of low-structure groupware: Experiences with collaborative writing software. Teoksessa *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*, sivut 41–50. 1996.
- M. Roseman ja S. Greenberg. Registration for real-time groupware. Tutkimusraportti 94-533-02, Department of Computer Science, University of Calga-

- ry. Internet WWW-sivu, URL: <http://cpsc.ucalgary.ca/grouplab/papers/>, 1994. (6.11.2001).
- M. Roseman ja S. Greenberg. Building real-time groupware with GroupKit, a groupware toolkit. *ACM Transactions on Computer-Human Interaction*, **3**, no. 1, 66–106, 1996.
- J. Roth ja C. Unger. An extensible classification model for distribution architectures of synchronous groupware. Teoksessa *Fourth International Conference on the Design of Cooperative Systems*, sivut 113 – 127. 2000.
- J. Schlichter. Lehrveranstaltung "Computergestutzte Gruppenarbeit". Technische Universität München, Fakultät für Informatik. Internet WWW-sivu, URL: [http://www11.in.tum.de/lehre/lectures/ws2001-02/cscw/extension/latex/cscw\\_course-student.pdf](http://www11.in.tum.de/lehre/lectures/ws2001-02/cscw/extension/latex/cscw_course-student.pdf), 2001. (1.11.2001).
- B. Singh. Invited talk on coordination systems. Organizational Computing Conference, 1989.
- M. Smith, J. J. Cadiz ja B. Burkhalter. Conversation trees and threaded chats. Teoksessa *Proceeding on the ACM 2000 Conference on Computer supported cooperative work*, sivut 97–105. 2000.
- R. B. Smith, R. Hixon ja B. Horan. Supporting flexible roles in a shared space. Teoksessa *Proceedings of the ACM 1998 conference on Computer supported cooperative work*, sivut 197–206. 1998.
- M. Sohlenkamp ja G. Chwelos. Integrating communication, cooperation, and awareness: the DIVA virtual office environment. Teoksessa *Proceedings of the conference on Computer supported cooperative work*, sivut 331 – 343. 1994.
- M. Stefik, D. G. Bobrow, G. Foster, S. Lanning ja D. Tatar. WYSIWIS revised: early experiences with multiuser interfaces. *ACM Transactions on Information Systems*, **5**, no. 2, 147–167, 1987.

- N. Streit, J. Haake, J. Hannemann, A. Lemke, W. Schuler, H. Schütt ja M. Thü-ring. SEPIA: a cooperative hypermedia authoring environment. Teoksessa *Proceedings of the ACM conference on Hypertext*, sivut 11–22. 1992.
- C. Sun. Undo any operation at any time in group editor. Teoksessa *Proceeding on the ACM 2000 Conference on Computer supported cooperative work*, sivut 191–200. 2000.
- C. Sun ja C. A. Ellis. Operational transformation in real-time group editors: Issues, algorithms, and achievements. Teoksessa *Proceedings of ACM Conference on Computer Supported Cooperative Work*, sivut 59–68. 1998.
- C. Sun ja R. Sasic. Consistency maintenance in web-based real-time group editors. Teoksessa *Proceedings of 19th IEEE Conference on Distributed Computing Systems*, sivut 15–22. 1999.
- Y. Yang, C. Sun, Y. Zhang ja X. Jia. Real time cooperative editing on the Internet. *IEEE Internet Computing*, **4**, no. 3, 18–25, 2000.

## Liite 1: Sanasto

Taulukko 1: Englannin kielisiä vastineita tutkielman keskeisille termeille.

<b>Suomeksi</b>	<b>Englanniksi</b>
hahmonäkymä	gestalt view
johdoshistoria	derivation history
lukitseminen	locking
kehitystyökalu	toolkit
kytkentä	coupling
omistautunut palvelin	dedicated server
operationaalinen transformaatio	operational transformation
ryhmäeditori	group editor
ryhmätyöohjelma	groupware
sarjoittaminen	serialization
tutkanäkymä	radar view
tietokoneavusteinen yhteiskirjoittaminen	computer supported collaborative writing
tietokoneavusteinen ryhmätyö	computer supported collaborative work (CSCW)
tietoisuus	awareness
työtilatietoisuus	workspace awareness
usean käyttäjän vierityspalkki	multi-user scrollbar
yhtenäisyyden hallinta	consistency control
yhteistyötietoinen	collaboration aware
yhteistyön suhteen läpinäkyvä	collaboration transparent