VAPAAN OHJELMISTOPROJEKTIN KAMPPAILUT REKURSIIVISEN YHTEISÖLLISYYDEN PUOLESTA

Etnografinen tapaustutkimus Hyperstruktuuri-tutkimusryhmän tarinasta 1999–2004

Juha Leminen

Pro gradu –tutkielma

Sosiologia
Yhteiskuntatieteiden ja filosofian laitos
Jyväskylän yliopisto

Heinäkuu 2006

TIIVISTELMÄ

VAPAAN OHJELMISTOPROJEKTIN KAMPPAILUT REKURSIIVISEN YHTEISÖLLISYYDEN PUOLESTA
Etnografinen tapaustutkimus Hyperstruktuuri-tutkimusryhmän tarinasta 1999–2004
Juha Leminen
Jyväskylän yliopisto
Yhteiskuntatieteiden ja filosofian laitos
Sosiologia
Ohjaaja: Jukka Laari
Heinäkuu 2006
98 sivua + 4 liitettä

Tämä tutkimus käsittelee Jyväskylän yliopistolla toimineen Hyperstruktuuri-tutkimusryhmän vapaiden ohjelmistoprojektien vaiheita vuosina 1999–2004. Tarkastelen tutkimusryhmän projektia vapaana ohjelmistoprojektina ja esitän projektin tapahtumat etnografisena kuvauksena. Keskeinen lähtökohta on hahmottaa ryhmän ohjelmistoprojektien kehitystä, eli mitä projekti saavutti ja miten se muuttui vuosien aikana. Empiirisesti johdettu tutkimusongelma on seurata, sitä miten projekti puolustaa vapaata toimintamalliaan käyttäen erilaisia argumentteja projektissaan vapaudestaan ja toisaalta ohjelmistonsa markkinointia koskien.

Näkokulmani tutkimuksessa on analysoida millaisia erilaisia diskursiivisia argumentteja käytetään ja miten toimintatavat muuttuvat vuosien kuluessa ryhmän toiminnassa. Aineistossa projekti näyttäytyy tieteellisenä tutkimusprojektina, vapaana ohjelmistoprojektina Internetissä sekä tutkimus- ja kehitystoimintana teknologiayritysten kanssa.

Tutkimuksen teoreettisia käsitteitä ovat rekursiivinen yhteisöllisyys, lahjatalous ja luottamus. Tutkimuksen tavoite on kehittää Christopher Keltyn teoriaa rekursiivisesta yhteisöstä. Tutkimuksen näkökulma on hahmottaa ohjelmoijien Internet-välitteistä toimintaa diskursiivisuuden materiaalisen luonteen ja vaihtotalouden kautta.

Ohjelmistoprojektin käytäntöjä havainnoitiin ohjelmistoprojektin työtiloissa, jossa seurattiin ohjelmoijien arkea ja haastateltiin ohjelmoijia. Havainnoin ohjelmistoprojektia myös seuraamalla sen toimintaa ja viestiliikennettä Internetin kehitysfoorumeilla vuosien 2001–2004 aikana. Tutkimuksessani on analysoitu vapaan ohjelmistoprojektin erilaisia kamppailuja Internet-ympäristössä sekä ohjelmiston esittelytilaisuuksia. Tutkimusryhmän vapaat ohjelmistokäytännöt tuotetaan uudestaan mm. patenttiongelmien yhteydessä. Tähän liittyen tutkimusryhmän projektin nimi vaihtui vuosien aikana. Projektin nimiä olivat Gzigzag, Gzz ja Fenfire.

Tutkimuksen tuloksena esitän hypoteettisen mallin siitä, miten vapaan ohjelmistoprojektin toimintaa voidaan kuvata yhteiskunnan sosiaalista rakennetta vasten.

AVAINSANAT: rekursiivinen yhteisöllisyys, vapaa ohjelmistoprojekti, Hyperstruktuuri-tutkimusryhmä, etnografinen tapaustutkimus, Internet-välitteinen toiminta

SISÄLTÖ

1. JOHDANTO 1

2. TUTKIMUKSEN TEOREETTINEN VIITEKEHYS 9
2.1 Innovaation käsite, ohjelmistokeksinnön oikeudellinen säätely ja innovaation toimintakulttuuri 9
2.2 Linux kiertävänä hyödykkeenä aikalaisanalyyttisestä näkökulmasta 11
2.3 Ohjelmoijayhteisö lahjataloutena 13
2.4 Luottamuspohjaiset verkostot vapaassa ohjelmistotoiminnassa 18

3. VAPAA OHJELMISTOPROJEKTI REKURSIIVISEN LIITTOUTUMISEN MUOTONA 20
3.1 Rekursiivisen yhteisön käsite 21
3.2 Diskursiivisuuden materiaalinen luonne käytännöissä 24
3.3 Ohjelmoijayhteisö informaatioaikakauden etnografisena tutkimuskohtena 25

4. HYPERSTRUKTUURI-TUTKIMUSRYHMÄN VAPAA OHJELMISTOPROJEKTI SOSIOLOGIAN TUTKIMUS-KOHTEENA 28
4.1 Tutkimuskysymykset 35
4.2 Aineisto 36
4.3 Aineiston tulkinnasta ja tutkimusprosessista 37
4.4 Tapaustutkimuksen validiteetti ja yleistäminen 39

5. NELSONIN JA LUKAN YHTEISHANKKEEN ALKU 41

6. VAPAAN OHJELMISTOLIIKKEEN PERUSTAJA KOMMEN-TOIMASSA GZIGZAGIN OHJELMISTOVERSIOTA 48
6.1 Hyödyllinen ohjelmistotuote vai tieteellinen tutkimustyö? 50
6.2 Laboratorioesineitä, ideoita norsunluutornissa vai toimiva vapaa ohjelmisto 51

7. GZZ-RYHMÄN OHJELMISTOTYÖN ARKEA HAVAINNOI-MASSA TOUKOKUUSSA 2002 54
7.1 Gzz-ryhmän työvälineet ohjelmistotyössä 56
7.2 Kuvaus Gzigzagin työstä ja ohjelmistovirheen korjaamisesta maanantain 20. toukokuuta 2002 työpäivän tapahtumista 63

8. KUINKA VAPAA OHJELMISTOKEHITYS MUUTTUU? : Gzigzagista Fenfireen – tutkimusryhmän projektit 2002–2004 69
8.1 Nelson: ”En koskaan olettanut olevani Open Source projektissa.” 73
8.2 ”Ei enää yhteistyötä Nelsonin kanssa” 75
8.3 Rahoitushakemukseen ja tieteellisen julkaisutoimintaan panostamista 79
8.4 Pääkehittäjä Lukka etsii yhteistyökumppaneita Free Software / Open Source yhteistyöhön 80
8.5 Uusi patenttiongelma-projekti vaarassa? 83

9. POHDINTAA 87
9.1 Kohti rekursiivisesti yhteisöllistä toimintatapaa? 88

LÄHTEET 92

LIITTEET

1. JOHDANTO

Oli aurinkoinen kesäpäivä heinäkuussa 2001. Astuimme tutkijakollegan[1] kanssa sisään Jyväskylän yliopiston Agoraan, jossa tapaisimme Gzigzag-projektin jäsenet ohjelmistotekniikan laitoksen tiloissa. Gzigzag oli nimi Jyväskylän yliopistolla toimivan Hyperstruktuuri-tutkimusryhmän ensimmäiselle vapaalle ohjelmistoprojektille. Olimme tulossa seuraamaan ryhmän ja Richard Stallmanin välistä tapaamista. Saavuimme Gzigzag-projektin tiloihin, jossa ryhmä opiskelijoiden näköisiä henkilöitä oli kokoontunut pelaamaan kiinalaista Go-lautapeliä. Meille tarjottiin kahvia ja ohjelmoijat opastivat meidät lyhyesti lautapelin salaisuuksiin. Siirryimme ryhmän jäsenten kanssa ensimmäisen kerroksen kokoustilaan odottamaan Stallmanin saapumista. Stallman saapuikin pian avustajansa kanssa paikalle. Ilmapiiristä aisti, että paikalla oli vain ohjelmoijia, joille hakkerointi oli tärkeä elämänmuoto. Gzigzag-projektin pääkehittäjä Tuomas Lukka esitteli Gzigzagin ohjelmistoversiota, josta kehittyi Lukan ja Stallmanin välille mielenkiintoinen keskustelu. Stallman esitti kriittisiä kommentteja ohjelmistoversiosta, mutta totesi sen olevan kuitenkin kehityskelpoinen. Keskustelun jälkeen siirryimme Stallmanin ja Lukan ryhmän jäsenten kanssa lounaalle, jolloin Stallman kaivoi esiin kannettavansa ja alkoi selata sähköpostejaan. Stallman oli sanottavansa sanonut ja valmistautui Jyväskylän yliopiston päärakennuksessa puhumaan ohjelmistojen vapaudesta ja tekijänoikeuksista. Stallman kiersi ympäri maailmaa erilaisissa tilaisuuksissa puhumassa vapaiden ohjelmistojen puolesta ja viemässä julistustaan käyttäjänoikeudesta (copyleft) eteenpäin.

Noin kaksi ja puoli vuotta myöhemmin saavuin entisen Gzigzag-projektin tiloihin keväisenä maaliskuun päivänä seuraamaan Fenfire-projektin kokousta. Tuomas Lukan aloittamasta Gzigzag-projektista oli jäljellä Fenfire-projektin pieni ohjelmoija-joukko. Benja, Janne ja Matti olivat nykyisen Fenfiren jäseniä. Lyhyen kokouksen aihe oli miten ryhmä työskentelisi muutaman kuukauden Soneran rahoituksella. Toinen tärkeä asia oli tulisiko ryhmän rekisteröityä yhdistykseksi jatkaakseen ohjelmiston kehittämistä yhdistysmuotoisena organisaationa. Ryhmän kehittämästä ohjelmistosta ei oltu saatu käytettävää versiota vuosien kuluessa kuin prototyypin tasolle, mutta vapaa ohjelmistoprojekti jatkui. Keskustelin englanniksi Benjan kanssa ryhmän tilanteesta sekä

[1] Tutkija Jussi Silvonen.

vaihdoin Jannen ja Matin kanssa kuulumisia kokouksen jälkeen. Projekti jatkui pienimuotoisena, vaikka Fenfiren[2] ”tuli” ei ollut niin vahva kuin aikaisempina vuosina. Fenfire oli kamppailunsa käynyt ja joutui vuonna 2004 mukautumaan pienen ohjelmistoprojektin arkeen, kun ryhmän taloudelliset resurssit olivat pienentyneet ja osittain sen vuoksi henkilöt projektissa vaihtuivat. Fenfire eli pienen projektin realistista arkea Jyväskylän yliopistolla Agoran uumenissa.

Mutta mitä kaikkea projektille tapahtui vuosien 1999–2004 kuluessa? Tämä tutkimus on tarina ryhmän synnystä, noususta ja tuhosta. Tutkimukseni tavoite on sosiologisesta näkökulmasta ymmärtää ohjelmistoprojektin toimintaa ja pohtia sitä, että voidaanko sen avulla hahmottaa vapaan ohjelmistotoiminnan suhdetta yhteiskunnan institutionaalisiin rakenteisiin silloin kuin puhutaan innovaatioista, joiden kehitys on luonteeltaan avointa. Näitä ovat esimerkiksi hyödykkeet, jotka toimivat avoimen lähdekoodin ohjelmistoilla, samalla tavalla kuin ohjelmistotuotteet, joita kehitetään avoimesti. Miten nämä yhteisölliset tavat toimia syntyvät ja kehittyvät? Tutkimukseni kuluessa haastattelemieni ohjelmoijien oli vaikeaa nimetä toimintaansa sitouttamalla se edustamaan ohjelmistoliikkeitä. He käyttivät kuitenkin vapaisiin ohjelmistoihiin liittyvää terminologiaa Internetin kotisivustoillaan ja keskustelivat kanssani terminologiasta haastattelutapaamisten yhteydessä. Minulle tämä herätti sosiologisen ambition, kuvata vapaan ohjelmistoprojektin toimintaa hahmottamalla sitä sosiologian käsitteiden avulla. Antropologi Christopher Keltyn artikkelista (2005), minulle syntyi idea rekursiivisista yhteisöistä (Recursive Publics). En ollut kuitenkaan tyytyväinen Keltyn teoriaan vaan asetin haasteekseni kehittää hänen teoriaansa. Tähän palaan tutkimukseni lopussa. Johdannossa esittelen tutkimukseni lähtökohdat ja tutkimusongelman vapaiden ohjelmistoliikkeiden historian valossa.

Tutkimukselleni keskeistä on etnografinen tutkimustapa. Kelty toteaa, että Emic lähtöinen tutkimustapa on ollut hänen tapansa hahmottaa ohjelmistoyhteisöjen toimintaa.

---

[2] Fenfire tarkoittaa suomeksi virvatulta. Fenfire oli ryhmän viimeisen projektin nimi. Virvatulta pidetään luonnossa esiintyvänä yliluonnollisena valoilmiönä, joka on pallomaisen abstrakti muodoltaan. Sen tieteelliset selitykset ovat vaihtelevia. Katso Wikipedian määritelmä http://fi.wikipedia.org/wiki/Virvatuli, tarkistettu 8.5.2006. Wikipedia toteaa myös virvatulen vertauskuvallisuudesta: ” *Runoissa ja lauluissa virvatulet ovat yleensä maailman houkutusten vertauskuva, ne ovat samalla tavalla pettäviä ja katoavia. Eino Leinon runossa Nocturne sanoo elämänsä taistelut jo läpikäynyt:* "En mä enää aja virvatulta". *Tällä hän tarkoittaa häilyväisten haaveiden ja onnen lupausten jahtaamisen lopettamista, ja tyytymistä siihen, mitä on käsillä.”*

Emic on selitys, jonka joukko ihmisiä antaa omalle toiminnalleen ja yhteisölleen[3]. Kelty (2003) kuvaa antropologian ja laadullisen tutkimuksen muutosta väittämällä, että aikakautemme totaaliset sosiaaliset faktat[4] ovat erilaisia algoritmeja, joiden kehittämiseen erilaiset yhteisöt uskovat. Pyrin toteuttamaan Keltyn ajatuksia antamalla omassa tutkimuksessani tilaa aineistossa esiintyvien ohjelmoijien omalle äänelle heidän yhteisöstään ja tavoitteistaan. Kuvaan ohjelmistoprojektin yhteisöllisyyttä ja sen yhteistyöverkostojen kehitystä, lähtien liikkeelle mikrotasolta ohjelmoijien välisestä vaihdon kulttuurista. Mielestäni on hedelmällisintä tutkia ohjelmistoryhmän syntyä pääasiassa sosiaalisen näkökulmasta, pyrkien ymmärtämään millaisia sidoksia ja velvoitteita verkostossa muodostuu toimijoiden, yhteisön ja instituutioiden välille. Analyysissäni pyrin tuomaan ilmi kytkökset, miten ohjelmistoryhmän toiminta yliopistollisessa kontekstissa liittyy myös taloudelliseen, teknologiseen ja tieteelliseen toimintaan. Ohjelmistotoiminnalla yliopistollisessa kontekstissa on erilaisia resursseja toteuttaa vapaata projektia kuin puhtaalla vapaalla ohjelmistoprojektilla tai yritysvetoisella vapaalla ohjelmistoprojektilla. Lukan johtamaa projektia voidaan luonnehtia ns. mesenaattimallin[5] toteutumiseksi, jossa projektin rahoitus hankittiin Jyväskylän yliopistossa Agora Centerin vaikutusvaltaisten tahojen toimesta.

Vapaisiin ohjelmistoihin on liitetty ihannoiva käsitys, että ne synnyttävät vapaaehtoista toimintaa, joka hyödyttää toimintaan osallistuvia monin tavoin. Siksi monet yritykset ovat kiinnostuneet Open Sourcesta uutena toimintastrategian mallina. Tällaisissa yhteisöissä yritykset ja ohjelmoijat toimivat kukin omista lähtökohdistaan käsin. Luovassa yhteistoiminnassa ohjelmistotoiminta on parhaimmillaan taloudellisesti yhteisöä hyödyttävää, toisaalta ohjelmoijat voivat saada paremman työvälineen itselleen projektin tuloksena. Vapaassa ohjelmistoympäristössä pyritään muodostamaan kulttuuriltaan toimiva sosiaalinen yhteisö, että pystyttäisiin valmistamaan ohjelmisto-hyödykkeitä, joilla olisi taloudellista vaihtoarvoa. Onko vapaassa ohjelmistotyössä kysymys taloudellisten kiihokkeiden täyttämisestä niille jotka antavat projektille resursseja, vai pyyteettömästä yhteisöllisestä toiminnasta, jossa toimijoilla on erilaisia

[3] Keltyn esitelmä vuodelta 2003, verkossa http://www.rlg.org/en/page.php?Page_ID=2201, tarkistettu 9.5.2006.
[4] Marcel Maussin (1999) lahjateoriassa totaalinen sosiaalinen fakta merkitsi, että lahjanvaihdon tapahtumaan yhdistyivät monet sosiaaliset käytännöt ja yhteiskunnalliset instituutiot.
[5] Kuvaus mesenaattimallista Ingo 2005, 76-79.

motivaatioita toiminnalleen? Tämä on yksi keskeinen kysymys koskien ohjelmistoyhteisöjen toimintaa.

Tutkimukseni on oma versioni ryhmän projekteista, jossa ryhmän toimijat saavat äänensä esille. Projektien tapahtumista on varmasti erilaisia tulkintoja sen toimijoiden keskuudessa sekä projekteja läheltä seuranneiden näkemyksissä. Aineistoluvuissa pyrin kuljettamaan projektin tarinaa mahdollisimman jouhevasti lukijan kannalta, ja tästä syystä tieto ohjelmistoihin liittyvistä teknisistä asioista on pyritty ilmaisemaan ala-viitteissä. Alaviitteistä löytyy usein myös tieto, mistä Internet-lähteestä voi aiheesta kiinnostunut tulkita itse projektien tapahtumia.

Vapaisiin ohjelmistoihin liittyvä toiminta ja kamppailut ilmenevät julkisuudessa ohjelmistoliikkeiden välillä. Liikkeiden välistä kamppailua värittävät erilaiset argumentit, joita ohjelmistoliikkeiden keulahahmot esittävät. Liikkeet käyttävät julkisuudessa erilaisia argumentteja perustellessaan ohjelmistojen vapautta tai avointa lähdekoodia. Vaden ja Vainio (2005, 224) esittävät ohjelmistokehitystoiminnan kolme mallia: Omistettu, Open Source ja Free Software toimintamallit ovat tapoja kehittää, jakaa ja lisensoida ohjelmistoja. Vaden ja Vainio (2005, 227) toteavat, että on käytännöllistä puhua vapaista ohjelmistoliikkeistä termillä Foss, joka on lyhennys termistä Free and Open Source Software. Vaden ja Vainio eivät kerro taustaa Foss termin synnylle[6]. Johdannossa käytän runkona Jon ”Maddog” Hallin puhetta koskien vapaata ohjelmistoliikettä. Hall kertoo miten ja miksi Foss-termiin päädyttiin. Hallin mukaan toimijoiden käytännölliset syyt ovat vaikuttaneet erilaisten vapaiden ohjelmistoliikkeiden syntyyn ja nimeämiseen.

Esittelen lyhyesti ohjelmistoliikkeiden historian käännekohtia ja millaisia termejä vapaisiin ohjelmiin liittyvät yhteisöt ovat käyttäneet itsestään. Kuuluisin Foss-termin puolesta puhuja on Linux Internationalin toiminnanjohtaja Jon ”Maddog” Hall[7]. Kuvaan Hallin Finlandia talolla 27.2.2004 pitämän puheen[8] pohjalta vapaiden ohjelmisto-liikkeiden terminologiaan liittyvää historiaa. Hallin puheen lainaukset perustuvat hänen

[6] Uusimmassa artikkelissaan Vainio ja Vaden (2006, 350) suomentavat FOSS:N VALO:KSI (vapaan ja avoimen lähdekoodin ohjelmistot), mutta eivät kerro termin synnystä tarkemmin. Itse käytän termiä FOSS tutkimuksessani.

[7] Maddogin lempinimi oli annettu hänelle jo nuorena hänen kiivaan luonteensa vuoksi. Käytän jatkossa hänestä nimeä Hall.

[8] Englanninkielisen puheen litteraatiosta Hallin lainaukset suomentanut Juha Leminen.

vastaukseensa yleisökysymykseen, jossa häneltä kysyttiin miksi hän käyttää vapaisiin ohjelmistoihin liittyen Free and Open Source Software termiä. Hall aloitti vastauksensa vapaan ohjelmistoliikkeen historian alusta:

JH: Sanon Free and Open Source Software, koska ensin oli Free Software.

Richard Stallmania on kutsuttu viimeiseksi hakkeriksi. Stallman alkoi kehittämään käyttöjärjestelmää Unixille Gnu-hankkeessa. Hankkeen suunnitelma tehtiin syksyllä 1983 ja koodin kirjoittaminen alkoi tammikuussa 1984[9]. Vuoden 1985 lopulla Stallman perusti Free Software Foundationin Gnu-hankkeen ympärille (Moody 2001, 40). Samana vuonna Stallman kehitti käyttäjänoikeuden (copyleft) tietokoneohjelmiin. Käyttäjänoikeus oli tekijänoikeuden vastakohta, jotta vapaan tietokoneohjelmien käyttäjillä olisi mahdollisuus kopioida ohjelmaa, muokata sitä tai myydä sen alkuperäisiä tai muokattuja versioita. Ohjelmistoissa käyttäjänoikeus ilmenee, kun niissä eksplisiittisesti määritellään käyttäjälle General Public License (GPL) lisenssi. Stallman (1999, 56) on erotellut ohjelmiston vapauden lajit seuraavasti: vapaus käyttää ohjelmistoa, vapaus muuntaa ohjelmistoa, vapaus jaella ohjelmistoa, vapaus parantaa ohjelmistoa ja julkistaa tiedot kaikille halukkaille käytettäväksi[10]. Lokakuussa 1985 Stallman perusti Free Software säätiön tukemaan Free Software-liikettä ja Gnu-projektia. Stallmanille ohjelmistojen vapaus merkitsi poliittista ohjelmanjulistusta, jonka tavoitteena oli luoda eettisiä arvoja liittämällä ohjelmistojen vapaus sananvapauden merkityksessä poliittisiin keskusteluihin ja yhteiskunnallisiin ongelmiin.

1990-luvun alussa suomalainen tietotekniikan opiskelija Linus Torvalds oli tyytymätön olemassa oleviin käyttöjärjestelmiin ja kirjoitti koodia luodakseen uuden käyttöjärjestelmän. Linuxin syntytarina on esitetty useissa kirjoissa[11], joten kerron lyhyesti vain sen pääkohdat. Kun Linus oli kirjoittanut ohjelmistonsa lähdekoodia, hän ilmoitti siitä Internetin Minix-uutisryhmässä ja antoi koodinsa vapaasti kehitettäväksi. Tämän jälkeen useita kehittäjiä ympäri maailmaa alkoi yhteistyössä kehittää Linuxia. Linuxin kehityksen pääareenoiksi muodostuivat sähköpostilistat ja CVS-tietovarasto. Linuxin kehitys jakautui erilaisiin modulaarisiin osiin. Linus itse kuitenkin vastasi

[9] Stallmanin haastattelu vuodelta 2006 Internetistä http://dialspace.dial.pipex.com/town/parade/df04/Richard_Stallman_Interview.pdf, tarkistettu 30.3.2006

[10] Tarkemmin näistä Stallmanin ohjelmiston vapauden lajeista, katso Silvonen (2003, 120-123).

[11] Linuxin syntytarina on esitetty mm. kirjoissa Kapinakoodi, Just For Fun, Linuxin tarina.

ytimen kehityksestä ja sääteli uuden koodin lisäämistä ohjelmistoon. Linusin kirjoittamasta koodista tuli ydin Stallmanin Gnu-ohjelmistoprojektille. Tästä ratkaisusta on peräisin kiista siitä millä nimellä Linux-käyttöjärjestelmää tulisi kutsua. Ohjelmiston nimen tulisi osoittaa tunnustusta eri tahoille, jotka vaikuttivat työllään Linux ohjelmiston syntyyn. Tämä kiista oli yksi syistä, jonka vuoksi syntyi kaksi erilaista arvomaailmaa ohjelmiston vapaudelle ja lähdekoodin avoimuudelle. Hallin mukaan Linux yhteisön keskeiset vaikuttajat eivät olleet tyytyväisiä Free-sanaan:

> JH: Eric Raymond, Bruce Perens ja monet muut eivät pitäneet Free Software käsitteestä, koska kaupallinen yhteisö oli hämmentynyt termistä. He ajattelivat sen tarkoittavan vapaa kuluista, vapaa rahasta, ilmainen. Kuuntelin tätä ja sanoin, että asiassa on väärinkäsitys. Paras asia, minkä voimme tehdä on opettaa ihmisille mitä todella tarkoitamme, eikä muuttaa termiä. Mutta he sanoivat tarvitsevansa jotain välittömästi päästäkseen yli pelosta, että kaikki ohjelmistot ovat vapaata. Heidän mielestään se (Free Software-käsite) vahingoittaa ohjelmistoja kahdella tavalla.

Hall kuvaa puheessaan miten Yhdysvalloissa paheksutaan ilmaisia asioita. Hän jatkaa, miten Perens ja Raymond saivat idean uudesta nimestä vapaille ohjelmistoille:

> JH: Sitten tuli idea Open Sourcesta. Kun puhuttiin Open Sourcesta, ihmiset kiinnostuivat ja kysyivät mitä se tarkoittaa.

Näin kaupallinen maailma kiinnostui termistä Open Source. Open Source liikkeen keulakuvaksi nousi Linus Torvalds. Termin avulla Open Sourcesta tuli strategia yhdistää avoin lähdekoodi kaupallisiin intresseihin. Bruce Perens esitti Open Source määritelmän kirjassa Open Sources (Perens 1999). Raymond ja Perens olivat perustaneet Open Source Initiativen jo helmikuussa 1998. Stallman ei kuitenkaan ollut tyytyväinen Open Source termiin (Stallman 2002), joten jotain oli tehtävä. Hallin mukaan oli saatava nämä kaksi leiriä yhteen, joten ratkaisu olisi käyttää termiä FOSS. Hallin mukaan FOSS kuvaa ohjelmistojen kaikkia puolia yhdessä fraasissa. Siinä yhdistyvät ohjelmistojen vapaus ja lähdekoodin avoin saatavuus. Hall kommentoi vastauksensa lopussa, että on sama mitä termiä käytät, kunhan vain käytät vapaata käyttöjärjestelmää:

> JH: Tarvitsemme kuitenkin jonkin nimen…Gnu / Linux on ok, käyttöjärjestelmä Linux ytimelle on ok, Linux on ok. Kunhan vain käytät sitä, niin se on ok.

Tutkijakollegani[12] kanssa lähetimme Hallille sähköpostilla kysymyksen myöhemmin vuonna 2004. Kysyimme häneltä, kuka käytti ensimmäisenä Foss-termiä. Samalla hän selvitti näkemystään, että avoimen ohjelmiston todellinen arvo on siinä, että toimintamallilla saataisiin aikaan käyttäjän haluamia asioita:

---

[12] Tutkija Juha K.Siltalan kanssa lähetimme Hallille sähköpostin Juha Lemisen sähköpostiosoitteesta 14.10.2004.

> JH: Kuten monet asiat, on hämmentävää sanoa kuka sanoi ensin, mitä ja milloin. Muistan että Richardin argumenttien takia termin Open Source käyttäminen oli väärin, koska se jätti pois käsityksen vapaudesta. Ihmiset käsittivät väärin Free Softwaren. Heille se merkitsi ilmaista ohjelmistoa. Jopa tänä päivänä ihmiset puhuvat Free Softwaren arvosta rahaa säästävänä, koska ei tarvita moninaisia lisenssejä omistetuille ohjelmille. Olen ymmärtänyt että FOSS:n TODELLINEN ARVO on kyky räätälöidä ohjelmisto käyttäjän tarpeiden mukaan. Siksi FOSS ei ollut mielestäni kompromissi vaan täsmällinen kuvaus jollekin, joka on samalla vapaa sekä avoin. Ja siksi aloin käyttämään sitä. En voi sanoa että olisin varmasti ensimmäinen, mutta en muista kuulleeni että kukaan ennen minua olisi käyttänyt sitä.[13]

Berry (2004) on osoittanut että tämä julkinen kiistely ohjelmistotoiminnan malleista esitellään erilaisista diskursiivisista lähtökohdista liikkeiden puhemiesten toimesta. Ohjelmistokoodi on solmukohta (nodal point), jonka luonteeseen argumentit usein liittyvät. Se on tuotu julkisen kiistelyn kohteeksi, jolloin siitä on tullut poliittinen keskustelu, joka liittyy ohjelmistojen omistusoikeuksiin sekä tiedon ja jaetun luovuuden vapaaseen jakamiseen. Diskurssiteorian valossa avoin-suljettu keskustelu edustaa antagonistista poliittista kiistaa (ystävä–vihollinen), joka ei ole hedelmällinen vapaassa ohjelmistotoiminnassa, vaan keskustelu tukahtuu Microsoftin ja avointen toimijoiden välillä[14]. Mouffen (2000) näkemyksen mukaan konfliktit ja antagonismit on tuotava julkiseen kiistelyyn sen sijaan että ne tukahdutettaisiin. Agonistinen malli tarjoaa vaihtoehdon tälle mallille. Agonistisessa mallissa kiistakumppanit jakavat yhteisen poliittisen viitekehyksen, mutta ovat eri mieltä sisällöllisistä määrittelyistä. Radikaali demokratia tarjoaa positiivisen vision vapaalle (libre) kultuurille (Berry ja Moss 2006.) Berry ja Moss (2006) yhdistävät *vapaisiin yhteisöllisiin intresseihin*[15] (libre commons) käsityksen radikaalista demokratiamallista. Tällainen agonistinen demokratian malli perustuu käsitykseen argumenteista, jotka tuottavat jatkuvia kamppailuja, joiden tarkoituksena on demokratian paikallinen takaaminen. Konfliktien jatkuva olemassaolo on demokratian elinehto. Berry ja Moss (2006) tarjoavat ns. kolmannen tien: vapaiden yhteisöllisten intressien radikaalille demokraattiselle projektille. Todelliset oikeudet vapaan kulttuurin liikkeille määrittyvät poliittisen demokratian päämäärien ja agonistisen debatin kautta. Demokraattisen politiikan tulee muuntua antagonisesta agonistiseksi (Mouffe 2005). Uudenlainen käsitys demokratiasta liittyy mitä suuremmissa määrin uudenlaisen julkisen tilan (public sphere) toimintapojen määrittelyihin. Väitän, että

---

[13] Jon Hallin sähköposti Juha Lemiselle 14.10.2004. Isot kirjaimet viestissä Hallin itsensä käyttämiä.

[14] Media on tuottanut julkisuudessa Microsoftin ja Linuxin välillä tätä ystävä-vihollinen vastakkain-asettelua. *”Microsoft tulee joskus mukaan Linuxiin, koska me tulemme murskaamaan heidät”* ja *”He ovat peloissaan. Luulen heidän vihdoin myöntävän, että me olemme vihollinen”* tyylisiä kommentteja Linus Torvaldsin väitetään sanoneen Microsoftista (Tikka 2003).

[15] Turunen (2006, 428) suomentaa Commons sanan yhteisölliseksi intressiksi.

ohjelmistoliikkeet, ohjelmistoprojektit, ohjelmistokehittäjät, Internetin kehittäjät ja Commons-liikkeet kamppailevat agonistisilla areenoilla erilaisten protokollien, toimintapojen ja oikeuksiensa puolesta. Normien vastavuoroisuus toimijoiden kesken kehittyy näillä avoimilla foorumeilla.

Tämän ohjelmistotoimintaan liittyvän diskursiivisen näkemyksen kautta johdan tutkimukseni tutkimusongelman. Päähuomio kohdistuu siihen, miten vapaan ohjelmistoprojektin arkipäiväinen toiminta Internet-välitteisesti ja neuvoteltavat valtasuhteet aktualisoituvat ohjelmistoprojektin erilaisilla foorumeilla. Diskurssianalyyttisesti tarkastelun kohteeksi nousevat erilaiset argumenointitavat ja käytännöt, joissa tuotetaan subjektit (ohjelmoijat argumenttien esittäjinä) ja objektit (vapaa ohjelmisto, tutkimustyön sovellutukset ja vapaan ohjelmistoprojektin käytännöt). Vapaa ohjelmistoprojekti ja sen käytännöt ovat sidoksissa toiminnan materiaalisuuteen, jonka kautta argumentit tuodaan yhteisölliseen prosessiin koeteltaviksi. Näitä kamppailuja esitän tutkimuksessani Internet-välitteisen toiminnan, keskustelujen sekä Lukan ryhmän jäsenten haastattelujen kautta. Keskeinen lähtökohta on hahmottaa ryhmän ohjelmistoprojektien kehitystä, eli mitä Lukan projekti saavutti ja miten se muuttui vuosien aikana. Empiirisesti johdettu tutkimusongelma on seurata, sitä miten projekti puolustaa vapaata toimintamalliaan käyttäen erilaisia argumentteja projektissaan vapaudestaan ja toisaalta ohjelmiston markkinointia koskien. Tutkimuksessa pyritään esittämään, millaisia erilaisia diskursiivisia argumentteja käytetään ja miten toimintatavat muuttuivat vuosien kuluessa Hyperstruktuuriryhmän toiminnassa.

Tutkimukseni on valmistunut Helsingin yliopiston Toiminnan teorian ja kehittävän työntutkimuksen yksikössä, jossa työskentelen Reijo Miettisen tutkimusryhmän tutkimusavustajana. Tutkimusryhmässä on jo kymmenen vuoden ajan tutkittu tiedepohjaisia innovaatioita ja tutkimustyön organisointia. Professori Miettisen tutkimus-ryhmän tutkimustyön seuraaminen jo useamman vuoden ajan on innoittanut minua tekemään kenttätyötä tutkimuskohteeni parissa.

2. TUTKIMUKSEN TEOREETTINEN VIITEKEHYS

Steven Weberin (2005, 180) mukaan Open Source[16] termiä käytetään eri yhteyksissä eri tavoin. Weber luettelee muutamia käyttötapoja. OS on määritelty suljetusta tai yrityslähtöisestä mallista poikkeavaksi tutkimus- ja kehitystyön metodiksi. Liikeyritykset käyttävät OS:a kuvaamaan uutta liiketoimintamallia toimintaympäristössään uusille tulos- ja palkintamekanismeille. OS määritellään myös yhteisöksi, jolla on verkostomainen tapa toimia yhdessä yhteisen hyödykkeen kehittämiseksi. OS:a on jopa pidetty uutena tuotteistamisrakenteena tietotalouksille. Toisaalta OS:a on teoretisoitu informaation vapautta perusoikeutena pitävänä poliittisena filosofiana.

Tässä luvussa pyrin määrittelemään teoreettisia viitekehyksiä, joihin vapaa ohjelmistotoiminta liittyy. Määrittelen FOSS tutkimuksen kentästä alueita, jotka ovat kiiinnostavia oman tutkimuskohteeni kannalta. Ensiksi esittelen, miten innovaatiota on käsitteenä teoretisoitu ja miten voidaan hahmottaa ohjelmistoinnovaation kehittämiseen liittyvää oikeudellista säätelyä. Toiseksi tulkitsen aikalaisanalyyttista tapaa ymmärtää Linuxia ja sen moninaista kiertoa hyödykkeenä. Kolmanneksi tulkitsen ohjelmisto-yhteisöjen vaihdannan kulttuuria lahjataloutena.

2.1 Innovaation käsite, ohjelmistokeksinnön oikeudellinen säätely ja innovaation toimintakulttuuri

Mikä tai mitä on innovaatio ja miten se tuotetaan yhteistyössä? Hyysalon (2006) mukaan innovaatio määritellään monissa yhteyksissään eri tavoin. Julkisessa keskustelussa innovaatio-sanan käyttö näyttäisi tarkoittavan yhteistoiminnallista kehystä mille tahansa luovalle toiminnalle. Monesti innovaatio voi koostua monestakin innovatiivisesta osasta. Tällaisia osia ovat innovaation tekniset, sosiaaliset tai organisatoriset osatekijät. Viime aikoina hajautetun ja avoimen innovaation käsitteet ovat saaneet jalansijaa innovaatiokeskustelussa. Innovaation syntyyn yhdistetään usein eri alojen tiedon ja osaamisen yhdistäminen, joita innovaation kehittäjät antavat toisilleen yhteistyösuhteissa. Innovaatiotoiminnan rahoittamiseen ja tutkimustulosten tuotteistamiseen kaupallisille markkinoille liittyy useita intressitahoja. Valtio, markkinat ja yhteisöt kiinnittyvät erilaisten säätely- ja tukijärjestelmien kautta innovaatioiden tutkimus- ja tuotekehitystoimintaan.

[16] Jatkossa käytän lyhennettä OS.

Vapaassa ohjelmistotyössä on kysymys vapaiden resurssien jakamisesta, jonka tuloksena on pyrkimys yhteisölliseen hyvän tuottamiseen. Miettisen (2006, 10) mukaan ohjelmistojen avointa kehittämismallia voidaan pitää uutena, Internetin mahdollistamana jaetun luomisen muotona. Ohjelmistoihin liittyvään yhteisen tiedon ja hyvän jakamiseen liittyy useita säätelyjärjestelmiä. Omistusoikeuksiin liittyvä lainsäädäntö kehittyy hitaammin kuin moninaiset yhteisölliset tavat toimia informaatiota sisältävien kohteiden parissa.

Yhteisölliset intressit (commons) ovat luonteeltaan kollektiivisia. Jo Rooman lain aikaan yhteisölliset intressit oli tiedostettu; niitä kuvasivat termit ”*res extra commercium*” tai ”*res communes*”. Yhteisölliset intressit on usein kuvattu informaatioaikakaudella julkisina yhteisöille vapaina alueina (public domain), vapaina ”kokoontumispaikkoina” Internet-välitteiselle toiminnalle. Yhteisöllisten intressien piiri on nähty perustuvan ainakin osittain altruistiseen toimintaan yhteisöissä. Turusen (2006) mukaan kuitenkin perinteiset immateriaalioikeudet, kuten patentit, tekijänoikeudet ja tavaramerkit, korostavat yhden toimijan tekijänoikeudellista valtaa. Ne aiheuttavat informaatioaikakaudella esteitä innovaatioiden vapaalle kehittämiselle. Turunen väittääkin, että digitaalitalouden dynaamisuuteen tarvittaisiin yhteisöllistä, luovaa toimintaa mahdollistavaa lainsäädäntöä, koska monet Internetin välityksellä kehitettävät kulttuuriset tuotteet ovat monien tekijöiden yhteisesti tuottamia. Nämä kulttuuriset tuotteet ovat monella tavoin lainsäädännön piirissä. Turunen määrittelee ohjelmiston neljänlaiseksi oikeudelliseksi kohteeksi: patenttien, tavaramerkkien, standardien ja Internet-osoitteiden nimistön oikeudellinen säätely immateriaalilainsäädännössä saattaa rajoittaa ohjelmiston kehittämistä. Ohjelmistossa on useita osia, jotka funktionaalisuutensa perusteella ovat erilaisia tietokoneperustaisia keksintöjä. Turunen määritteleekin neljänlaisia piirteitä ohjelmistokeksinnöissä:

(1) Lähdekoodi (source code). Ohjelmoijien tietokonekielellä kirjoitettu lähdekoodi.
(2) Objektikoodi (object code). Lähdekoodi käännettynä tietokoneen luettavaksi.
(3) Algoritmi (algorithm). Matemaattiset ilmaukset lähdekoodissa.
(4) Ohjelman visuaalinen ulkoasu. Ohjelman tapa ilmetä näytöllä ja sen visuaalinen tapa loppukäyttäjälle. (Turunen 2006, 218-219).

Castells ja Himanen (2004) määrittelevät nk. suomalaisen innovaatiomallin perustan vertaamalla suomalaista mallia Silicon Valleyn toimintatapoihin. Mallin ensimmäinen osatekijä ovat lahjakkaat ihmiset (talented people). Suomalaisessa innovaatiomallissa

perustan ihmisten osaamiselle muodostavat julkinen, vapaa ja korkealaatuinen koulutusjärjestelmä. Suomen korkeakoulujärjestelmä on myös hyvin kattava. Nokia on tyyppiesimerkki siitä, kuinka tutkimus- ja kehitysyhteistyötä tehdään yhteistyössä yliopistojen kanssa. Toinen tärkeä osatekijä on innovaatioprosessin rahoitus. Valtio on keskeinen rahoittaja innovaatioiden tutkimus- ja kehitystyölle. Tärkeimmät valtion rahoittajatahot ovat Sitra ja Tekes, jotka toimivat Kauppa- ja teollisuusministeriön alaisuudessa. Tekesin rahoitus oli merkittävä tekijä esimerkiksi Nokian kehittäessä Gsm-standardia matkapuhelininnovaatioonsa. Kolmas keskeinen menestyksen tae on vahva kansallinen innovaatiokulttuuri. Castells ja Himanen mainitsevat Linus Torvaldsin esimerkkinä siitä kuinka positiivinen innovaatiokulttuuri rohkaisi hänet luomaan Linuxin. Valtio, liiketoiminnan tahot ja luovat ihmiset yhteistoiminnassa tekevät mahdolliseksi innovaatiokulttuurin menestystarinat. Neljäs ja tärkeä osatekijä on yrittäjyys, joka merkitsee idean kaupallistamista markkinoille. Castellsin ja Himasen mukaan on olemassa useita malleja saada idea markkinoille. Yleisin kaupallistamisen tapa on yritysvetoinen malli. Muita kaupallistamisen malleja ovat julkisen sektorin tai kansalaisvetoiset mallit saada idea tuotteistettua käytettäväksi kaupallisilla markkinoilla.

2.2 Linux kiertävänä hyödykkeenä aikalaisanalyyttisesta näkökulmasta

Informaatioaikakauden keskeisin teoreetikko Manuel Castells määrittelee informaatioaikakauden prosessin virtana tilassa sidottuna informaation kiertoon kapitalistisessa yhteiskunnassa ja sen tuotantomuodoissa. (Castells 2004, 36-37). Castellsin verkostoyhteiskunnan teoriassa käytännöt ovat verkostoituneita, ja siksi myös toiminnan tila vaihdon kulttuurissa on verkostoitunut. Verkostoituneiden käytäntöjen informaatiovirta prosessoituu hallittavaksi erilaisten kommunikaatioteknologioiden välittämänä. Castellsille verkostoyhteiskunnan tila ilmenee kolmenlaisissa informaation vaihdannan ympäristöissä: (1) paikoissa, joihin ihmisten toiminnan konteksti on sijoitettu, (2) materiaalisissa kommunikaatioverkostoissa, jotka yhdistävät näitä toimintoja sekä (3) informaatiovirran sisällössä ja sen geometriassa, joka tulee esiin erilaisissa verkostotoiminnan funktioissa ja tarkoituksissa. Castellsin mukaan aikakausien murroskohtien muutoksille on keskeistä, että uuden sosiaalisen rakenteen muodostuminen on kytketty ihmisten toiminnan materiaaliselle perustalle ja sen vaihdon kulttuurille.

Lee ja LiPuma ovat kehittäneet oman tavan ymmärtää informaatioaikakauden vaihdannan kulttuuria. He kutsuvat käsitystään kierron kulttuuriksi (cultures of circulation), jossa kiertävät hyödykkeet ovat eri tavoin performatiivisia. Lee ja LiPuman (2002) mukaan performatiivisuus on kulttuurisen ilmiön yhteydessä kytketty merkityksen syntymisen prosesseihin, joissa kierto ja vaihto on nähty lähettävän merkityksiä. Ne ovat enemmän kuin perustavia toimintoja sellaisenaan. Kun tämä erottelu ylitetään vaatii kierto uudelleen käsitteellistämisen kulttuurisena ilmiönä. Kierron kulttuurissa ei ole enää mahdollista ajatella kiertoa yksinkertaisena liikkeenä ihmisten, hyödykkeiden, ideoitten ja kuvien välillä paikasta toiseen. ”*Kierto on kulttuurinen prosessi*” Lee ja LiPuman mukaan (2002, 192) ja he määrittelevät kiertämisen prosessin seuraavasti: ”*…omine abstraktiivisine, evaluatiivisine, rajoitettuine muotoineen, jotka on luotu interaktiossa erityistyyppisten kiertävien muotojen välille ja niitä ympäröivien kuvailevien yhteisöjen välille. Nämä kuvailevat yhteisöt määrittelevät toimintatavoille kuvaukset keksiäkseen muotoja, tunnustaakseen käytäntöjä, perustaakseen instituutioita ja määritellessään rajoja perustuen omalle sisäiselle dynamiikalleen*.”

Lee ja LiPuma kutsuvat näitä rakenteellisia kiertoja siis kierron kulttuureiksi. Adrian Mackenzie on saanut vaikutteita Leen ja LiPuman kierron kulttuuria käsittelevästä teoriasta. Mackenzie (2005) hahmottelee Linuxia informaatioaikakauden hyödykkeiden kierron kulttuurissa ja sen kapitalistisissa vaihdannan muodoissa, joissa ohjelmiston koodi tai ohjelmistoprojekti ovat performatiivisia, kulttuurisia objekteja. Mackenzie esittelee Linuxista tuotettuja erilaisia kulttuurisia muotoja sekä hyödykkeitä niiden kiertäessä informaatioaikakauden virroissa. Ohjelmiston tekninen esitys on aina yhdistetty kulttuurisesti performatiiviseen esitystapaan, jossa sillä eri tuotteistetuissa muodoissaan on käyttöarvo kulttuurisesti performaativisena hyödykkeenä.

Mitä Mackenzielle sitten tarkoittaa käsitys, että ohjelmisto toiminnallisena objektina on performatiivinen? Hänen mukaansa ohjelmistokoodi ei koskaan ilmene tai ole toiminnassa ennen lukuisia käytäntöjä. Tällaisia käytäntöjä ovat lähdekoodin kaupallistumiseen ja käytön ympärille syntyvät kulttuuriset merkitykset, kun yritykset, yhteisöt ja ohjelmoijat tuottavat, kierrättävät ja käyttävät Linuxia. Toiminnallisena objektina työpöytänä palvellessaan, Linux melko kirjaimellisesti koordinoi sosiaalisia toimintoja koskien informaatio- ja kommunikaatioverkostoja. Linux erilaisissa

lähdekoodatuissa muodoissaan kiertää ihmisten välisessä ohjelmistotoiminnassa, jossa sen ohjelmistokoodin arvo on käyttöarvoltaan muuttuva sen käyttäjäyhteisöille. Ohjelmiston koodi eri toimijoiden käyttöintressien toiminnan tuloksena sisältää enemmän tai vähemmän sosiaalisia tai kulttuurisia arvoja erotettuna taloudellisista arvoista[17].

Miksi siis Mackenzielle on tärkeää, että vapaa ohjelmisto on itseensä viittaava objekti? Ohjelmiston voima on sen kaksinaisfunktionaalisessa luonteessa. Kun ohjelmistoa käytetään samalla kun sitä yhtäaikaisesti kehitetään, ohjelmistovirheet löydetään helposti. Silloin ohjelmiston käyttöarvo ilmenee koodissa ja kymmenissä, sadoissa tai tuhansissa sen käyttäjien viesteissä ja heidän kirjoittamissaan koodin pätkissä ja korjausehdotuksissa. Koodissa näkyy myös sen kehittäjiensä määrän taito ja halu taistella vapaan lähdekoodin kehittämisen puolesta. Koodi saa yhteisöllisyydestä ja siihen kuuluvien yksilöiden osallistumisesta sen tuottamiseen käyttöarvonsa. Vähämäki (2006, 282-283) tulkitsee Karl Marxia seuraten käsitystä tavaramuodosta ja sen käyttöarvosta. Yksilön valmistama tuotteiden esineellinen luonne ilmenee toisille tehtyinä tuotteina luonnostaan yhteiskunnallisina ominaisuuksina. Vähämäen mukaan työn tavaramuodon tuotteistuminen on peräisin siitä hetkestä, jona ihmiset alkavat työskennellä toisiaan varten. Kun yksilön tekemä työ saa yhteiskunnallisen muodon on mahdollista työskennellä toisiaan varten. Silloin heidän työnsä muuttuu yhtäläiseksi, yhteismitalliseksi.

2.3 Ohjelmoijayhteisö lahjataloutena

Steven Weber on yksi viime vuosien keskeinen Open Source toimintamallia eritellyt tutkija. Weberin (2004) mukaan kuusi Open Source toiminnan keskeistä edellytystä sen menestymiseksi ovat seuraavat:

1. Hajallaan oleva tieto voidaan koota vain olosuhteissa, jossa tietoon on rajoittamaton pääsy, idea ei ole omistettu tai suljettu.
2. Kriittisen käyttäjien massan on pidettävä tuotetta arvokkaana.
3. Tuote hyötyy laajasta vertaishavainnoinnista ja arvioinnista, jolloin sitä voidaan parantaa luovilla aloitteilla virheiden korjaamisella (ts. virheiden korjaamisen aste ylittää virheiden tuottamisasteen).
4. Tuotteen käyttämisestä on myönteisiä verkostovaikutuksia.

[17] Mackenzie (2006) käsittelee ohjelmistoa sosiaalisena objektina ja prosessina. Hän tutkii toimijuuden palautuvuutta liittyen erilaisiin ohjelmistoprosesseihin (emt, 19).

5. Yksilö tai pieni ryhmä voi asettua johtoon ja tuottaa sisällöllisen ytimen, joka on lupaava kehittyäkseen joksikin todella hyödylliseksi.
6. Tuotteen rakentamisen ympärille voi rakentua toistuvassa vuorovaikutuksessa oleva vapaaehtoisten yhteisö.

Kaikki Weberin OS toiminnan edellyttämät piirteet liittyvät yhteisön sosiaaliseen toiminnan dynamiikkaan. Vapaita ohjelmistoyhteisöjä on kuitenkin tutkittu usealla eri tavalla. Näkökulmia ovat olleet myös ymmärtää ohjelmoijien sisäisen motivaation ja ulkoisen motivaation suhdetta ohjelmoijien toiminnan perustana. Toisaalta on teoretisoitu erilaisia sopimussuhteita, esimerkiksi psykologinen sopimus toimijoiden välillä. Pääasiassa ilmiötä on kuitenkin tutkittu kahden erilaisen tieteenalan näkökulmasta (Millar, Choi, Russell ja Kim 2005): (1) perinteisen taloudellisen hyödyn ajattelumalleilla, sekä (2) antropologisen tutkimuksen näkökulmasta. Yksi keskeinen tapa, jolla Eric Raymond käsitti ideaalisen Open Source toiminnan ohjelmistonkehittäjäyhteisöissä on lahjatalouden käsite (Raymond, 1999). Ohjelmistotyössä on kysymys erilaisista tavoista vaihtaa informaatiota ja informaation kautta siirtyviä osaamisresursseja itse koodin ja tiedon muodostumiseen. Raymondin mukaan vertaistunnustus ja maine ovat keskeisiä motivaation lähteitä ohjelmoijille. Ohjelmoijat ovat kehittäneet monimutkaisen vertaisarviointijärjestelmän, jossa on samankaltaisia piirteitä kuin tieteellisessä vertaisarviointijärjestelmässä.

Miten sosiologiassa on käsitelty luottamusta ja vastavuoroisuuden normeja? Miksi yksilö tekisi pyyteettömiä tekoja haluamatta korvausta? Sosiologisen teorian varhainen käsitys lahjasta on peräisin durkheimilaisen koulukunnan antropologisista tutkimuksista Marcel Maussilta. Durkheimin (1980) mukaan oli aina ensin sosiaalinen yhteisöllisyys, joka rakentuu jaetusta tiedosta ja kokemuksista. Sosiaaliset tosiseikat mahdollistivat yhteisön sosiaaliset faktat ”sui generis”, jotka ilmenevät uskonnollisina rituaaleina, lakeina, sääntöinä, jne. Yhteisön edustamat arvot tulivat ilmi kollektiivisina representaatioina tai esityksinä. Durkheimin mukaan yhteisön moraalinen konsensus syntyi aina vähitellen[18]. Durkheimille yhteisöllisyys ei ollut kuitenkaan prosessi sarjassa tapahtuvia toimintoja, vaan yhteisöllisyys syntyi verkostomaisesti erilaisissa uskomusjärjestelmissä.

[18] Ilmosen ja Jokisen (2002, 53) mukaan Durkheimin vastike luottamukselle oli käsitys moraalisista yhteisöistä ja niiden toimintatavoista. Niissä ilmeni ekspressiivistä solidaarisuutta, koska niissä luotiin sosiaalisia siteitä. Toisaalta he kritisoivat (emt, 35), että Durkheim toisaalla tutkimuksissaan sotki luottamuksen tuttuuteen.

Jälkimodernissa maailmassa eri teknologioiden ja kulkuneuvojen välittäminä yhteisöt voidaan ymmärtää verkostoiksi, joissa toimijat ovat kontaktien muodostamien solmujen päässä toisistaan. McNeill ja McNeillin (2005, 475) mukaan verkostoitumisen keskeisin muoto on ollut läpi ihmiskunnan historian erilaisten symbolien käyttö verkostojen luomiseksi. Verkostot välittivät sovittuja merkityksiä ja johtavat ajan myötä yhteistyöhön ja yhteenottoihin alati suurempien ihmisjoukkojen kesken.

Lahjataloudessa yksilöt lahjoittavat resurssejaan vastavuoroisesti. Lahjan antamisen mekanismit on korvattu sosiaalisilla siteillä. Marcel Mauss oli Emile Durkheimin siskonpoika, joka omaksui lahjateoriassaan näkökulmakseen antropologisen elementarismin Durkheimin ajattelusta (Cheal 1988, 2-3). Poiketen poliittisen taloustieteen vaihdon käsitteestä, Mauss esitti lahjan käsitteen antropologisten tutkimustensa pohjalta, jotka koskettivat vaihtotaloutta alkuperäiskansojen yhteisöissä. Maussin mukaan näiden yhteisöjen vaihdannan muoto oli lahjanvaihto. Sosiaalisiin suhteisiin kytkeytyvän voimavaran ylläpitoa leimasi oman edun tavoittelun ja altruistisen toiminnan yhdistelmä, joka ilmeni erilaisina vastavuoroisina vaihdon muotoina. Kuten Mauss toteaa (1999, 42); ”*Symmetristen ja vastavuoroisten lahjojen järjestelmä ei ole lainkaan vaikeaa käsittää, jos ymmärrämme sen olevan ennen kaikkea henkisten siteiden verkosto. Järjestelmä perustuu tietyllä tavalla sielun omaaville tavaroille sekä ihmisille ja ihmisryhmille, jotka kohtelevat toisiaan tietyssä mielessä esineinä. Kaikki nämä instituutiot paljastavat yhden ja saman sosiaalisen faktan tai alueen, tietyn mentaliteetin.*”

Hän kutsui näiden yhteisöjen talousmuotoa lahjataloudeksi. Sen yhtenä keskeisenä piirteenä on, että jotakin palvelua tai hyödykettä ei vaihdettu välttämättä suoraan toiseen palveluun tai hyödykkeeseen, vaan takaisinmaksuun saattaa mennä vuosia. Maussille lahjanvaihto oli totaalisten suoritteiden järjestelmä. Totaalisten suoritteiden järjestelmä tarkoitti, että kaikki kulttuurissa olevat materiaaliset esineet kuuluivat vaihdon piiriin. Mauss käsitti vaihdon Durkheimia seuraten totaaliseksi sosiaaliseksi faktaksi yhdistäessään erilaiset käytännöt ja niiden kulttuurilliset merkitykset vaihdon piiriin. Godelierin (1999) mukaan lahjoilla on myös antajansa persoonallinen voima, jonka salaisuutta Mauss pyrki ratkaisemaan. Lahjanvaihtoa leimasi oman edun tavoittelun ja altruistisen toiminnan ristiriita. Alkuperäiskansoilla oli erilaiset termit lahjanvaihdon tavoille ja niiden arvottamiselle. Maussin ratkaisu kuvaamaan lahjan antajan

persoonallista voimaa oli *hau*, asioiden henki[19] (Mauss 1999, 143-144). Godelierin (1999, 151) mukaan ratkaisu ymmärtää *haun* käsitettä Maussin teoriassa on selvittää yhteisöjen sosiaalisen järjestyksen juuria ja erilaisten sosiaalisten siteiden syntyä. Godelierin vastaus lahjaesineeseen sisältyvän persoonallisen voiman tulkitsemiseksi oli kysymys sosiaalisesta siteestä ja siihen luottamisen luonteesta. Tämä sosiaalinen side on verrattavissa luottamuksen käsitteellistämiseen sosiaalisena siteenä vastavuoroisessa toiminnassa aikamme kulttuurissa (Ilmonen 2004, 1). Kuitenkin Maussin tulkintoja kohtaan on vuosikymmenien aikana esitetty kritiikkiä usealta taholta, mm. *haun* käsitteestä, joka on keskeinen vaihtotapahtuman käsitteellistämisen kannalta[20].

Firthin kritiikki jo pian Maussin kirjan ilmestymisen jälkeen oli, että Mauss sekoitti alkuperäiskansojen *haun* ja *utun* käsitteet (Godelier 1999, 143-144). *Hau* merkitsi latinan spiritusta, joka alkuperäiskansoille oli tuuli tai sielu. *Utu* oli puolestaan monimuotoisempi käsite kuvaamaan vaihdon ja vastavuoroisuuden monia eri puolia ja velvoitteita. *Utu* voidaan ymmärtää myös hinnaksi. Hyöty (profit) olisi ollut parempi sana kuvaamaan *hauta*, mutta sen aikaisessa Maori-yhteisössä ei sellaista sanaa ollut (Godelier 1999, 53). Keskeinen käsite maoreille oli *mana*, joka kuvasi antajan kunniaa ja arvovaltaa, jolla antaja sitoi persoonaallisen voimansa lahjaan. Douglasin mukaan maorien *hau* oli jonkinlainen vastine Adam Smithin näkymättömän käden käsitteelle (Douglas 1999, 17). Holvas (2000, 116) puolestaan toteaa, että haun käsitteelle oli keskeistä *palautuvuus:* ”Hau ei tee sitä tai tätä oman tahtonsa mukaan, vaan kiertävien tavaroiden voima tähtää aina yhteen asiaan: itse prosessin pyörittämiseen, tavaroiden kierrättämiseen, asioiden *palautuvuuteen*.”

Eric Raymond kuvasi ensimmäisen kerran 1990-luvun lopulla OS liikkeen teoreettisena keulakuvana kirjassaan avoimen ohjelmoijayhteisön toimintaa lahjatalouden metaforalla. Raymondin mukaan OS-toiminnan lahjataloudessa sosiaalinen status määräytyy sen mukaan miten pyyteettömästi ohjelmoija osallistui projektiin. Tärkeintä ohjelmoijalle ei ole se, mitä hän pystyy itse kontrolloimaan vaan mitä hän antoi muille osallistumalla vapaaseen ohjelmistokehitykseen. Bergquistin ja Ljungbergin mukaan (2001) lahjataloutta ei ole tärkeää käsitellä ainoastaan avoimuuden luomisena ohjelmointi-kulttuuriin, vaan merkittävää on, että lahjatalous organisoi ihmisten välisiä suhteita

[19] Katso myös Holvas 2000.
[20] Katso Ilmonen 2004 tarkemmin kritiikin luonteesta.

tietyllä tavalla. Raymondille tämä ymmärrys kulttuurin logiikasta ja käytäntöjen jakamisesta on vapaan ohjelmistotoiminnan dynamiikan ydin. Bergquistin ja Ljungbergin (2001) mukaan lähtökohdallisesti käsitys ohjelmistokulttuurista lahja-taloutena kertoo yhteisön kulttuurisesta järjestyksestä, jossa arvot ja normit yhteisön toiminnassa syntyvät erilaisten asetelmien ja toimijoiden toiminnan tuloksena. Ideoitten jakaminen toiminnan keskeisenä muotona antaa lisää voimaa lahjan vaihdolle ja kiertämiselle. Bergquist ja Ljungberg (2001) ymmärtävät Maussin väittävän, että lahja esittää vaihdon prosessia, mutta myös luo valtasuhteita ihmisten välille. Siksi idea lahjan antamisesta yhteisön toiminnan tapana on keskeinen, jotta ohjelmistoyhteisön sosiaalistumisprosessi monimuotoisuudessaan tulisi ymmärretyksi.

Open Sourcen toiminnan piirteitä on verrattu myös tieteellisen tutkimustyön luonteeseen. (Bergquist 2003). Siinä on samoja piirteitä kuin tieteellisessä vertaisarvioinnissa (Raymond 1999, Bezroukov 1999). Hagstrom (1982) yhdisti tieteellisen tutkimustyön ja julkaisuprosessin lahjatalouden ideaan. Latourille ja Woolgarille (1979) ei-rahallisen vaihdon metafora tieteellisissä laboratorioissa oli ansion kierto (cycle of credit).

Lahjatalouden idea on lähtökohta ymmärtää ohjelmistokulttuuria. Hakkerikulttuurin alkuaikoina lahjatalous oli osuvampi määritelmä, mutta nykyisin ohjelmoijien motivaatiot ja vaihdon muodot ovat moninaisempia. Mielestäni kuitenkin lahja sekä luottamus ovat käsitteitä, jotka kuvaavat ohjelmoijien välistä toimintaa, jossa kunnioitetaan vastavuoroisuuden normeja erilaisilla tunnustamisen[21] (recognition) mekanismeilla. Kuitenkin Maussin lahjateorian esittely osoitti, että jo primitiivisessä kulttuurissa oli erilaisia termejä vaihtotalouden sosiaalisille muodoille. 2000-luvulla vaihtotalouden prosessit perustuvat ”immateriaalisen” informaation vaihtoon, jonka omistusoikeudelliset kiistat Internetin foorumeilla toimijoiden kesken ovat luonteeltaan monimutkaisempia. Siaperan (2006) mukaan monikulttuurisuuden (multiculturalism) ja tunnustamisen tavat tulisi ymmärtää avoimena kamppailujen kenttänä, jossa jatkuva kamppailujen tila takaa demokratian määrittelemisen myös online-välitteisessä toiminnassa.

[21] Tunnustamisen politiikka on filosofi Charles Taylorin (Taylor 1992, 25-73) aloittama keskustelu tunnustamisen tavoista monikulttuurisessa yhteiskunnassa.

2.4 Luottamuspohjaiset verkostot vapaassa ohjelmistotoiminnassa

Miten lahjatalous on mahdollinen ja millainen vaihdon verkosto muodostuu ohjelmistoprojektien ympärille digitaalisessa maailmassa Internet-välitteisesti? Vapaassa ohjelmistoprojektissa on avoimet foorumit projektin kommunikaatiolle ja työtavoille. Ohjelmistokulttuurissa tätä mallia kutsutaan peer-to-peer malliksi, jossa kehittäjät / käyttäjät kommunikoivat ilman välittäviä tahoja. Kommunikaatiofoorumit toimivat ”*solmukohtina*”, joihin erilaisia osaamisen muotoja keskittyy. Vapaassa ohjelmistoprojektissa on pyrkimyksenä laajentaa kehittäjäkuntaa erilaisiin suhde- ja kontaktiverkostoihin. Immateriaalisten kontaktiverkostojen toiminnan jatkuvuus ja kasvoton kanssakäyminen perustuvat luottamuksellisuuteen ohjelmoijien välisessä yhteistoiminnassa. Zeitlyn (2003, 1290) on esittänyt väitteen, että etnografinen tutkimus pystyisi osoittamaan OS yhteisöjen tutkimiselle sopivaksi käsitteiksi lahjatalouden ja samankaltaisuuden yhteisymmärryksen (kinship amity). Tutkimuksessani ymmärrän vapaan ohjelmistotoiminnan vaihdon verkostoksi, jossa sosiaaliset siteet mahdollistavat vaihdon kulttuurin. Keskustelu luottamuksesta tai luottamuksellisuudesta liittyy tähän yhteyteen. Sosiaalisen siteen arvoituksellisuus perustuu vaihdon luottamuksellisuuden tapaan, joka ilmenee toimijoiden tulevissa toimenpiteissä. Ilmosen (2005, 49) mukaan sosiaalisen suhteen luottamuksellisuus auttaa pääsemään käsiksi sosiaalisen verkoston osapuolten hallussa oleviin tiedollisiin, taidollisiin, poliittisiin ja taloudellisiin varantoihin. Vastavuoroisuuden luonteessa on huomioitava kuitenkin toimijan halukkuus osallistua ja hänen valmiuteensa auttaa ja antaa osaamisresurssejaan toisten käyttöön. Ilmosen (2005, 57) mukaan myös lahjateoria liittyy sosiaalisen suhteen positiiviseen ja negatiiviseen vallankäyttöön. Graeber (2001) kutsuu puolestaan vastavuoroisuutta mekanismiksi, jossa tasapainoillaan avoimen ja suljetun vaihdon välillä. Ihmisten toiminta jälkikäteen voi muuttaa vastavuoroisuuden luonnetta. Luottamus lähestyy positiivisessa vallankäytön muodostamassa sosiaalisessa suhteessa solidaarisuutta, tuen antamista silloin, kun luottamuksen osoittaja kaipaa sitä. Toisaalta tuen antajalla on negatiivista valtaa. Hänellä on tahtoessaan valta pyrkiä aiheuttamaan hankaluuksia. Kumpikin tapa ymmärtää vallankäyttö luottamussuhteessa lähentää sen lahjatalouteen. Lahjoja ei anneta pelkästä anteliaisuudesta. Niihin kytkeytyy myös vastalahjan vaatimuksia, jotka ovat eri tavoin artikuloitu vaihdon prosessissa.

Millaisia ovat sitten tunnustamisen välineet ohjelmistotyön lahjataloudessa? Tunnustamisen välineet ja tavat syntyvät jokaisessa vapaassa ohjelmistoprojektissa tavoiksi ja käytännöiksi osoittaa kunnioitusta. Lahjatalous on markkinoita näkyvämpi, koska se on riippuvainen julkisesta arvostuksesta, kunnian jakautumisesta ja yhteisöllisistä sanktioista. Luottamus voi ilmentyä ainakin kahdella tasolla: luottamuksena instituutioihin tai yksilöihin. Ilmosen (2000) mukaan epäluottamuksen dynamiikka on rakenne, joka syntyy nopeasti osapuolten välille, mutta purkautuu hitaasti. Luottamuksen synty on perustunut mahdollisesti oletukseen, että uskomme toisen sanaan tai lupaukseen.

Tiedon ja informaation vaihto ovat keskeisiä toimintatapoja ohjelmistoyhteisöille. Vaihdon muodot ja ohjelmoijien erilaiset toimintatavat herättävät uudenlaisia kysymyksiä sosiaalisten siteiden luonteesta. Luottamuksellinen (trustworthy) vaihtosuhde sosiaalisena siteenä on kytkeytynyt tunnustussuhteisiin osaamisen ja informaation vaihdossa. Informaatiota vaihdetaan ohjelmistoprojektissa useilla erilaisilla toimintavoilla. Luottamuksen lähde on toiminnan avoimuus ja sitoutuminen vapaaseen toimintamalliin. Luottamusta on myös monella tasolla. Tasoja ovat esimerkiksi ohjelmoijan luottamus lähdekoodin luonteeseen, vapaan tietomallin rakentamiseen ja toisten ohjelmoijien pyyteettömään panokseen projektissa. Gallivanin (2001, 280) mukaan johtamisen ja sosiologian teorioissa on esitelty viisi erilaista luottamuksen lajia. Gallivanin mukaan ne ovat seuraavat:

1. Tietoon perustuva luottamus: luottamus perustuen osapuolten välisen pitkään kestäneen toiminnan perustalta.
2. Luonteenpiirteisiin perustuva luottamus: luottamus toisen persoonallisiin ominaisuuksiin.
3. Institutionaalisuuteen perustuva luottamus: luotettava ympäristö, jonka toimijoihin voi takuuvarmasti luottaa.
4. Oikeudellisiin syihin perustuva luottamus: luottamus laillisiin menettelytapoihin.
5. Nopea (swift) luottamus: hauras luottamuksen laji, joka ilmenee vauhdikkaasti virtuaalisissa työryhmissä tai tiimeissä.

Gallivan (2001, 294) mukaan ohjelmistoprojektissa luottamuksen tasot ovat erilaisia toimijoiden kesken. Keskeiset (core) kehittäjät, reuna (peripheral) kehittäjät, virheviestien lähettäjät (posters) muodostavat keskenään erilaisia verkostoituneita luottamussuhteita. Ohjelmistotyön kehittäminen perustuu ideoitten vaihtoon lähdekoodin

luomiseksi. Prosessin aikana lähdekoodia muutetaan, poistetaan ja kirjoitetaan uutta tilalle. Oma roolinsa on pelkillä käyttäjillä, jotka raportoivat ongelmista käyttökokemuksissaan. Suuressa projektissa muodostuu näin ollen satoja tai tuhansia erilaisia sosiaalisia siteitä käyttäjien ja ohjelmoijien välille.

Tässä luvussa olen kuvannut, kuinka lahjatalous, luottamuksen verkostomainen ominaisuus sekä käsitys kiertävästä materiaalisesta hyödykkeestä ovat erilaisia tapoja käsitteellistää ohjelmistotoiminnan materiaalisen vaihdannan ulottuvuutta, jossa sosiaalinen ja teknologinen ovat toisiinsa kietoutuneita. Lahjatalous ja luottamus ovat sosiologisia käsitteitä, joiden avulla voidaan ymmärtää aikamme materiaalista vaihtotaloutta ja sen kytkeytymistä erilaisiin tapoihin luoda uusia luottamuksellisia sosiaalisia siteitä, jotka liittyvät innovatiivisten ideoiden omistamiseen ja materiaalisuuteen. Koska Internet-välittyneisyys perustuu luonteeltaan yhteisen kohteen kehittämiseen, siihen liittyvä luottamuksellisuus on luonteeltaan materiaalisen haurasta. Sosiaalisen siteen luottamus ei ole karvan varassa informaationaalisessa yhteiskunnassa, vaan sitä ylläpitää luottamus ohjelmistokoodiin ja yhteisöllisiin toimintatapoihin. Aktiivinen luottamus[22] syntyy jälkitraditionaalisessa maailmassa, kun uskotaan yhteisen materiaalisen kohteen kehittämiseen ja vaalitaan toiminnan kohteen edustamia arvoja.

3.VAPAA OHJELMISTOPROJEKTI REKURSIIVISEN LIITTOUTUMISEN MUOTONA

Kun ohjelmistoyhteisöt ymmärretään kehittyviksi sosiaalisiksi yhteisöiksi, niitä on usein kutsuttu käsitteellä virtuaalinen yhteisö. Latham ja Sassen (2005, 9) kehittävät uuden tavan ymmärtää virtuaalisia yhteisöjä. Heidän käsityksensä on digitaalinen muodostuma, jonka lähtökohtana on sosiaalisen muodon ja muodostumisen prosessin yhdistäminen. Heidän mukaansa digitaaliset muodostumat tulisi ymmärtää ontologiselta asemaltaan sosiaalisina asioina. Internetin välittämänä näiden digitaalisten muodostumien prosessit ovat osittain esillä, julkisesti avoimena informaationa. Näitä prosesseja täytyisikin havannoida sekä seurata käytössä ja toiminnassa (Latham ja Sassen 2005, 28). He tarkoittavat muodostumalla lähtökohdallisesti neljänlaisia asioita:

1. Muodot ovat varhaisessa kehityksen vaiheessa.

[22] Katso aktiivisen luottamuksen käsitteestä esimerkiksi Giddens 1996, 252-253.

2. Niiden synty ei ole osoitettu perustamistapahtumissa, virallisissa konstituutioissa tai säännöissä. Ne ovat sekoitus epämuodollisia osasia ulottuen verkosto-suunnitelmista julistuksiin.
3. Jokainen sosiaalinen muoto on muuttumiskelpoinen sopivissa konteksteissa, tekijöissä, suhteissa ja johdonmukaisuudessaan tästä hetkestä seuraavaan.
4. Ymmärrys digitaalisista muodostumista on aluillaan ja tulee muuttumaan, kun analyysit olemassa olevista ja uusista syntyvistä muodostumista kumuloituvat.

Vapaita ohjelmistoja kehittävien yhteisöjen keskeinen piirre on, että niiden ohjelmointikulttuurissa työvälineet ja työn kohde eli ohjelmiston tuottaminen ovat kiinteästi kytkeytyneitä toisiinsa. Tämä rekursiivinen kietoutuminen (recursive loop) johtuu ohjelmoijan kaksinaisesta roolista ohjelmiston tuottamisessa. Nämä ovat ohjelmistokehittäjän ja -käyttäjän roolit. Yksi keskeinen ohjelmoijien motivaatio on, että he kehittävät ohjelmistotyökaluja omaan käyttöönsä. Ohjelmistotyöläiset kehittävät omia ratkaisuja lisätä ja parantaa omia vapaita ohjelmistotyökalujaan. Tuotettavaa vapaata ohjelmistoa ei voi erottaa vapaan ohjelmistotyön sosiaalisista käytännöistä. (Mackenzie 2001, 547-548).

3.1 Rekursiivisen yhteisön käsite

Kelty (2005) puolestaan käyttää rekursiivisuuden ideaa nörttien (geeks)[23] tutkimisen perustana. Keltyn idea on keskeinen tutkimustapani kannalta, joten esittelen tiivistetysti sen pääpiirteet. Rekursiivisuus käsitteenä tarkoittaa toistumista, palautumista, takaisin tulemista[24]. Ohjelmoijien välisessä toiminnassa se ilmenee diskursiivisten argumenttien kautta, jotka on suunnattu ohjelmointiteknisen toiminnan uudelleen rakentamiseen jatkuvassa prosessissa.

[23] Kelty käyttää ohjelmoijista termiä nörtit (geeks), koska hänen mukaansa se on ohjelmoijien itse käyttämä ja hyväksymä termi yhteisönsä jäsenistä.

[24] Sosiologisessa teoriassa rekursiivisuus teoretisoitiin 1980-luvulla Giddensin ja Bourdieun aikalaiskäsityksissä. Giddensin rakenteistumisteoriassa rekursiivisuus tulkitaan sosiaalisten järjestelmien ominaisuudeksi, jossa rakenne on samanaikaisesti sekä toimijan toiminnan lähtökohta, että tulos joko uusinnettuna tai muunnoksena. Bourdieulle (1990, 52) habitus merkitsi sekä käytäntöjen tulosta (opus operatum) ja käytäntöjen muotoa (modus operandi).

Keltylle rekursiivinen yhteisö (recursive publics) on nörttien liittoutumisen muoto kamppailla puolustamiensa teknisten ja laillisten näkökulmien puolesta. Kelty määrittelee liittoutumisen käsitteen ymmärtämällä sen Bruno Latourin liittoutumisen (collective) käsitteen perustalta sekä Castellsin verkostoyhteiskunta käsityksen pohjalta. Keltyn (2005, 187) mukaan hän ei tulkitse Internetiä pelkästään tieteen ja teknologian tutkimuksen näkökulmasta verkostona, joka yhdistää sosiaaliset ja tekniset koosteet vaan myös ”todellisena” verkostona, jonka teknisten ja laillisten muotojen kehittyminen kuvataan tiheänä kuvauksena (thick description) siihen osallistuvien toimijoiden argumenttien kautta. Argumentit ja erilaiset toimintatavat ilmenevät Internetin erilaisilla foorumeilla toimijoiden puheen ja toiminnan valossa. Ryhmä yksilöitä jakaa useimmiten yhteisiä arvoja, vain liittoutuakseen toistensa kanssa koska heillä on yhteinen huoli tilansa mahdollisuuksista liittoutua kehittääkseen yhteisöllisiä toimintatapoja. Internetin teknologinen kehitysprosessi on samankaltainen kamppailujen kenttä, samalla tavoin kuin erilaiset avoimet foorumit Internetin sisällä. Ohjelmoijien sosiaalinen yhdessä toimimisen muoto koostuu teknisten käytäntöjen sekä toiminnan diskursiivisten argumenttien yhteistuloksena. Avoimuus on käytäntö ja käsite josta rekursiivisuus on riippuvainen. Jos ei ole pääsyä näkemään ohjelmiston lähdekoodia tai muuta vapaan teknologian kehitysprosessia ja sen protokollia, niin prosessit eivät ole avoimia, eikä erityinen rekursiivinen yhteisöllisyys ole olemassa. Ohjelmistotoimijoiden sosiaalisen ryhmän toiminnan tavat ovat luonteeltaan rekursiivisia liittoutumisen muotoja, joille Internet on tämä liittoutumisen muoto. Heidän toimintansa Internetissä ymmärretään kamppailujen areenaksi. Rekursiivinen yhteisö on käsitteellisenä välineenä tärkeä, koska se kuvaa tapaa jolla ohjelmoijat käsittävät sosiaalisen olemassaolon tapansa teknisten käytäntöjen läpi yhtä paljon kuin diskursiivisten argumenttiensa kautta. Ohjelmiston kirjoittamisen käytäntö on verbaalisen argumentin ja materiaalisen käytännön välissä. Julkinen tila ohjelmoijien toiminnassa on rekursiivinen, koska tällaisella alueella on kyky keskustella ja mahdollisesti uudelleen valmistaa protokollia. Ohjelmoijat kamppailevat pitääkseen Internetin ja sen sisällyttämät vapaat toiminnan muodot rekursiivisen yhteisöllisyyden puolesta radikaalisti avoimina. Ohjelmoijat jakavat todellisuuden yhteisenä kuvitelmana siitä mitä yhteisö tai sosiaalisuus merkitsee. Vapaat ohjelmistoprojektit tai muut vapaat toimimisen areenat Internetissä ovat luonteeltaan

kamppailun areenoja, jotka määrittelevät yhteisön toimintatapojen arvoja ja käytäntöjen toimivuutta jatkuvissa, palautuvissa ja loppumattomissa prosesseissa[25].

Keltyn käsitteen ”Recursive Publics” monitulkinnallisuuden sekä sen vuoksi, ettei käsitteelle ole hyvää käännöstä suomeksi, kysyin Christopher Keltyltä sähköpostitse käsitteeseen selvennystä. Hän vastasi seuraavasti[26]:

> Viitaten kysymykseesi, termi julkinen merkitsee samaa kuin Habermas käyttää (avoin (Öffentlickeit) saksan kielellä), mutta se on myös suora vastaus Charles Taylorin ja Michael Warnerin työhön, johon on viitattu artikkelissa. Se ei ole julkinen kuten Dewey määrittelee, vaikka siinä on mielenkiintoisia yhteneväisyyksiä (tämä on sama suunta kuin johon Latour vie termiä).
>
> Joka tapauksessa tärkein syy siihen, että käytän termiä, on että haluan korostaa tärkeää eroavaisuutta Habermasiin. Habermasilla ja monille muille poliittisille teoreetikoille, julkisen diskursiivisuus tulee esiin puhumisessa ja kirjoittamisessa ja on hyvin tavanomaista laatua (esim. kasvokkain vuorovaikutuksessa ja lehtien julkaisuissa, pamfleteissa ja traktaateissa). Haluan selittää kuinka ohjelmisto ja verkosto ovat myös julkisen diskursiivisuuden muoto samankaltaisessa mielessä, ja koska ne ovat erityyppisiä välineitä (tarkoittaen välineitä jotka tukevat toista mediaa, kuten puhetta, chattia, printtaamista, sähköposti, jne), siksi kiistat ovat rekursiivisia----jokaisella keskustelun tasolla on keskustelu joka on sen tason alapuolella, ulottuen kykyihin chatata reaaliajassa, alas yksityiskohtiin fyysisen tason yli, joita verkko kuljettaa (puhelinlinjat, langattomat signaalit, matkapuhelimet, jne) Tämä ”recursive publics” ei koske ainoastaan julkisen diskursiivisuutta, mutta välittömästi (tai läheisesti) mahdollisia tiloja joissa on mahdollisuus erityisen julkisen olemassaololle...niin kauan kunnes se ”pysähtyy”(halts).

Veikko Pietilä kommentoi[27] minulle termiä ”Recursive Publics” ja toi ilmi ongelmia, joita Keltyn määrittelyt termistä pitävät sisällään. Pietilän mukaan Keltyn ongelma on, ettei hän tee erottelua Habermasin Publikum ja Offentlichkeit termien välillä. Habermasin (2004) teoriassa Publikum oli ”yhteisö”, joka keskusteli Offentlichkeitin alla. Pietilä kehoittaakin suhtautumaan Keltyn inspiraation lähteenä, eikä niinkään miettimään ”Recursive Publics” sanan tarkkaa suomennosta. Asetan työssäni kyseisen termin teoreettiseksi välineeksi, jonka avulla tulkitsen aineistoani Hyperstruktuuriryhmän tapahtumista. Tutkimukseni on haaste on soveltaa Keltyn termiä tutkimukseni aineistoon. Ilmosen (2005, 50) mukaan moniaineksinen käsite voidaan

---

[25] Keltyn teoretisointi on uusi, siihen on kuitenkin viitannut esimerkiksi Coleman (2005). Sosiaalisen kuvitelman teoriassa on esitelty samankaltaisia elementtejä jo aiemmin (Gaonwar & Povinelli 2002).
[26] Christopher Keltyn sähköposti 26.4.2006.
[27] Vaihdoin Veikko Pietilän kanssa sähköpostiviestejä Recursive Publics termistä 28.4.2006 ja 8.5.2006. Pietilä on viestintätieteen emeritusprofessori Tampereen Yliopistosta. Hän on suomentanut Jurgen Habermasin kirjan Julkisuuden rakennemuutos. Pietilä on kehittänyt publikum termille uusiosanan julkiso, jota Mika Renvall käyttää myös uudessa Dewey (2006) suomennoksessaan. Julkiso sanan käyttäminen ei ole levinnyt kuitenkaan laajalle suomalaisessa yhteiskuntatieteellisessä tutkimuksessa, vaan se on lähinnä viestintätieteilijöiden käyttämä termi.

valita ohjaamaan tutkimusta: ”*Tämä tarkoittaa sitä, että seurataan Derek Layderin ohjetta, jota hän suosittaa adaptiivisen teoriankehittelyn ajatusmallissaan. Sen mukaan jokin käsite pitää valita teoreettisin perustein konkreettisen tutkimuksen ohjaajaksi*.” Sovellan Keltyn käsitystä ohjelmoijista ja rekursiivisesta julkisesta tilasta tutkimalla erilaisia kamppailuja vapaan ohjelmistoprojektin toiminnassa. Kuvaan projektin erilaisia vapaan toiminnan alueita ja niissä tapahtuvia kamppailuja, joiden kautta projekti määrittelee vapautensa uudestaan rekursiiviseksi yhteisöksi. Kelty kuvaa näitä argumenttitapoja, ensinnäkin teknologisina käytäntöinä, joita ovat mm. hakkerointi, verkostoituminen, koodin kirjoittaminen sekä diskursiivisina argumentteina, joita esimerkiksi kuvaavat oikeudet, identiteetit ja toimijoiden väliset suhteet. Diskursiivisuus tulee ilmi myös erilaisina kielellisinä argumentteina. Ohjelmoijien oikeus puuhailla (tinkering) ilmenee erilaisina vapaata puhetta ilmentävinä käytäntöinä liittyen vapaisiin ohjelmistoihin.

3.2 Diskursiivisuuden materiaalinen luonne käytännöissä

Kelty (2005) ei määrittele ”*diskursiivisen*” argumentin luonnetta tekstissään, joten otan tutkimukseeni avuksi diskurssiteoriaa myötäilevän käsityksen termistä. Yritän tutkimuksessani tulkita miten diskursiivisuus on kytketty vapaista ohjelmistoista käytettyihin argumentteihin sekä käytäntöihin.

Diskurssiteorian keskeiset kehittäjät Laclau & Mouffe (1985, 108) määrittelevät diskursiivisuuden Wittgensteinin kielipelitematiikan kautta. Wittgensteinin myöhäistuotannon (Wittgenstein 1981, 24-25) seinän rakentamista kuvaavan esimerkin kautta he kuvaavat diskursiivisuuden *materiaalista* luonnetta. Merkittävä näkökulma Wittgensteinin kielipelien käsitteeseen on, että se on kytketty ”elämänmuotoon”: ”*Sanan kielipeli on määrä tähdentää tässä sitä, että kielen puhuminen on osa* ***toimintaa*** *tai elämänmuotoa* (emt 1981, 36). Kielen erityinen käyttö heijastuu toimimisen ja elämisen tapaan (emt 1981, 32): ”*Kielen kuvitteleminen merkitsee elämänmuodon kuvittelemista*”[28].

[28] Schatzki (2001) liittää tämän Wittgensteinin seinän rakentamisen esimerkin käytäntöihin ja sosiaalisen järjestyksen käsitteeseen. Wittgensteinin käsitykseen säännöistä ja sääntöjen seuraamisesta liittyy laaja keskustelu, jota ei ole tässä yhteydessä sen laajuuden vuoksi mahdollista käydä. Wittgenstein toteaa (1981, 35), että: ”Säännön seuraaminen, tiedotuksen esittäminen, käskyn esittäminen, shakkierän pelaaminen ovat tapoja (käyttöjä, instituutioita).”

Erilaiset käytännöt liittyen materiaaliseen tekemiseen kytkeytyvät täten kielipeliin. Kirjoitetut dokumentit, puhe, ideat, konkreettiset käytännöt, rituaalit, instituutiot ja empiiriset objektit kontekstissaan määrittyvät Laclaulle ja Mouffelle diskursiivisiksi (Howarth 2005, 333-334) (Smith A. 1998, 85-86). Tämä diskursiivisuuden määritelmä liittää sen kiinteästi käytäntöjen käsitteellistämisen luonteeseen. Sosiaaliset käytännöt tuotetaan artikulaatioprosesseissa. (Laclau ja Mouffe 1985, 109, 113).

Bridgman ja Willmott (2006) tulkitsevat teknologian materiaalisuutta ja organisaation muutosta myös diskurssiteorian näkökulmasta. Heidän mukaansa (2006, 123) Laclaun ja Mouffen diskurssiteoria keskittää huomiomme identiteettien, suhteiden ja objektien väliseen olemassaoloon, johon sisällytetään myös teknologian materiaaliset ominaisuudet. Institutionaalisessa kontekstissa ne on konstruoitu objekteiksi, joihin vaikuttavat niiden väliset vuorovaikutussuhteet ja niiden teknologisten ominaisuuksien tuottajat ja käyttäjät. Tämä kerrostuminen (sedimentation) ilmentää miten teknologiat ja instituutiot on artikuloitu ja hegemonisesti kytketty diskursseihin jotka esittävät niiden välisiä suhteita.

Oma näkökulmani on ymmärtää näitä erilaisia diskursiivisia argumentteja vapaan ohjelmistoprojektin toiminnassa. Wittgensteinin seinänrakennusesimerkki tapahtuu tässä ja nyt, hetkellisessä ja kasvollisessa toiminnassa. Olennaista on se, että miten käsitys *diskursiivisuudesta* voidaan käsitteellistää uudella tavalla Internet-välitteisen toiminnan kontekstissa. Internetille on luonteenomaista toiminnan tekstuaalisuus ja välittyneisyys, johon liittyy mahdollisuus käyttää aikaa ja harkintaa vuorovaikutuksessa. Tämä mahdollistaa uudenlaisen tavan toimia strategisesti.

3.3 Ohjelmoijayhteisö informaatioaikakauden etnografisena tutkimus-kohteena

Geertzin (1973, 22) mukaan antropologin tutkimuksen kohde ei ollut kylä, vaan kohteena oli tutkia kylän asukkaiden kulttuuria antropologin viettäessä aikaansa kylässä. Ohjelmistotyössä tapahtumat ovat sijoittuneet Internet-välitteisiin käytäntöihin myös ohjelmistoryhmän toimistolla. Yuwei Lin (2005) on tuonut ilmi, että vapaa

ohjelmistoyhteisö ei ilmene ainoastaan ohjelmistokoodin vaihdossa vaan se tuotetaan erilaisissa yhteisöllisissä ongelmanratkaisukäytännöissä. Keskeistä Linin mukaan vapaiden ohjelmiston ympärillä tehtävälle tutkimukselle ja sen tutkimustavalle on tutkia ongelmanratkaisun prosesseja ja siihen liittyviä tapahtumaketjuja jokapäiväisissä käytännöissä. Virtuaalietnografian näkökulmalle keskeinen havainnointitapa tulisi olla käytäntökeskeinen tutkimusote (Lin, 2005)[29].

Tutkimukseni keskeinen lähtökohta etnografiselle tutkimustavalle ja sen metodologialle ovat erilaiset näkökulmat ohjelmiston ja ohjelmistoprojektin vapauteen projektin toiminnassa ja käytännöissä. Ohjelmistokulttuurissa vapautta on havainnoitu kolmessa erilaisessa toimintakontekstissa. Ohjelmistoprojektin käytännöt noudattavat vapaan ohjelmistoliikkeen periaatteita, joten tämä vapauskäsitys tarkoittaa sitä, että Internet-välitteisesti ohjelmistoprojektin *käytännöt ovat vapaita*. Toiseksi ohjelmistoprojektia esitellään erilaisissa tapahtumissa seminaareissa ja muissa yhteyksissä, joissa ohjelmistoprojekti esitellään *vapaana ohjelmoijien puheessa ja toiminnassa*. Kolmanneksi tutkimuksessa on havainnoitu työtä ryhmän työtiloissa, joten kolmas tapa ymmärtää vapaata ohjelmistoprojektia on miten sitä tuotetaan yhteisöllisesti *vapaana arkisessa ohjelmistotyössä* ohjelmistoryhmän työtiloissa.

Van den Boomen ja Schäferin (2005) artikkeli kertoo kuinka Open Source termi kulkee läpi yhteiskunnan eri tavoin: sitä edustavien puhemiesten, erilaisten väitteiden, tuotteiden sekä diskursiivisten strategioiden (esim. markkinointistrategia) kautta termille rakentuu erilaisia kulttuurisia merkityksiä. Tutkimuksessani ei tutkita ohjelmistokoodia sinänsä, vaan ryhmän innovaativisten ideoitten valmistamista erilaisilla ”*rekursiivisen yhteisöllisyyden alueilla*” ryhmän oman vapaan ohjelmistotyön periaatteiden mukaisesti. Hyperstruktuuri-tutkimusryhmän projektia tarkastellaan vapaana ohjelmistotyöprojektina sen erilaisilla näyttämöillä. Projektin tapahtumista vuosina 1999–2004 syntyy etnografinen kuvaus ohjelmistoprojektista tieteellisenä tutkimusprojektina, vapaana ohjelmistoprojektina Internetissä sekä tutkimus– ja kehitystoimintana teknologiayritysten kanssa. Tutkimusryhmän vapaat ohjelmistokäytännöt joudutaan tuottamaan uudestaan esimerkiksi patenttiongelmien yhteydessä. Tarkastelen ohjelmistokoodin ja ohjelmistotuotteiden valmistamisen prosessia erilaisten havainnointiaineistojen kautta,

[29] Linin (2005) artikkeli ” The Future of sociology of FLOSS”. FLOSS tarkoittaa neutraalia kannanottoa Free Software ja Open Source liikkeiden välisiin kiistoihin. Johdannossa esittelin näkemykseni vapaisiin ohjelmistoliikkeisiin liittyvästä ”diskursiivisesta” terminologiasta.

joissa projekti näyttäytyy vapaana ohjelmistoprojektina päämääränään vapaan ohjelmistotuotteen valmistaminen käyttäjäyhteisölle. Projektin julkisuuskuva välittyy ja näyttäytyy haastatteluissa, lehtihaastatteluissa sekä kaikissa julkisissa yhteyksissä, joissa projekti on esillä. Bruno Latourin (2005, 178-179) mukaan ”*on seurattava kytkentöjä, on seurattava toimijoita sellaisenaan*” sähköisen dokumentaation ympäristöissä. Tietokonevälitteisessä ympäristössä seuraamisen kohde ovat ohjelmoijien tuottamat erilaiset dokumentit, artefaktit ja niiden syntyprosessit. Keskeinen arvo ryhmän työlle oli Open Source / Free Software metodologian yhdistäminen omaan tutkimus- ja kehitystoimintaansa.

Samalla kun ohjelmistoa työstettiin avoimuuden periaatteilla, tieteellistä tutkimus-dokumentaatiota omista prototyypeistä kirjoitettiin samaan tietovarastoon. Avoimen ohjelmistotyön arvomaailman alle projektissa rakentui siis erilaisia tutkimustyön intresseihin perustuvia tiedostoja sateenvarjomaisesti tietovaraston alaisuuteen. Tutkimusryhmän tietovarastosta (CVS) muodostui keskus, jonka kautta kaikki projektin tuottama tieto kulki, samalla toimien varastona moninaisille tutkimustiedostoille. Strukturalistista perinnettä seuraten Latour (1987) kehitti teoksessaan kaksi käsitettä, jotka ovat keskeisiä paikan käsittämiselle toimijaverkkoteoriassa. Nämä käsitteet ovat metrologia (metrology) ja laskentakeskukset (centers of calculation). Koordinaatio-keskukset (Suchman 1997) ja tietämyskeskukset (Hautamäki 2003) ovat vastaavanlaisia yrityksiä teoretisoida toiminnan keskuksia erilaiselle informaatioperustaiselle yhteistoiminnalle. Barryn (2001, 202) mukaan keskukset liittyvät tieteellisen tiedon tuottamisessa erityisiin lokalisoituihin laskennan paikkoihin; kuten observoinnin, monitoroinnin, teknisten käytäntöjen tai kokeiden suorittamisen paikkoihin. Laboratorio ja työpaja ovat erityisiä ympäristöjä, jossa kokeet ja monitorointi tuottavat objekteilleen uusia tuloksia. Paikat joissa kokeilut ja tekniset innovaatiot vakiinnutetaan sijaitsevat moninaisissa ympäristöissä. Kokeiluja suoritetaan kentillä, tehtaissa, byrokratian erilaisissa koneistossa, sairaaloissa, vankiloissa ja museoissa. Dayn (2004) mukaan keskukset ovat koosteita historiallisesta tiedosta. Ne eivät ole pelkästään evidenssiä olemassa olevista materiaalisista objekteista, vaan ne konstruoivat olemassa olevaa historiallisesti. Kirjastot ovat keskeisiä historiallisia tiedon keskuksia. Kuvaan lyhyesti laskentakeskuksen idean Latourin vuoden 1996 artikkelin *Ces réseaux que la raison ignore - laboratoires, bibliothèques, collections* perusteella. Latourin ideaa laskentakeskuksesta voidaan verrata CVS-tietovaraston keskeiseen asemaan

ohjelmistotyössä. Tietovarasto on mielestäni samankaltainen funktionaalinen keskus kuin laskentakeskus. Latourin laskentakeskuksen idea kuvaa kuinka entiteetit kulkevat ylä – ja alavirtaan sitä virtausta, joka siirtää maailmaan partikuularisten entiteettien ”*laskentakeskukseksi*”. Latourille informaatio on kahden paikan välinen suhde. Toinen paikoista on aina periferia suhteessa keskukseen riippuen virtaussuunnasta. Idean täytyy muuntua representaatioksi, jotta sitä voidaan kuljettaa. Informaatio ei ole ainoastaan platoninen muoto, vaan materialisoituma kahden paikan välillä. Informaation laajentaminen ja supistaminen vaatii ammattimaista käytännön harjoittajaa. Jokainen informaation pala hyödyntää lähellä olevaa palaa, mikäli se hyödyttää rekisteriä, laskentaa, ohjelmaa ja jos se voidaan asettaa vierekkäin toisen vastaavan kanssa. Kun sitä voidaan katsoa ja tutkia, siitä tulee yhteismitallinen kaikkien siihen saakka toisilleen täysin vieraiden todellisuuden alueiden tai osasten kanssa. Latourille esimerkiksi tietokone on aktantti, joka edustaa ja tuottaa yhteismitallisuutta, koska me kaikki käytämme tietokoneita, jotka muuttuivat hiljattain fyysisesti tarpeeksi tehokkaiksi käsittelemään, yhdistelemään ja kääntämään toisistaan erillisiä piirroksia, tekstejä, valokuvia ja laskelmia. Numeeriseen muotoon muuttaminen eli digitalisointi pidentää laskentakeskusten pitkää historiaa tarjoamalla jokaiselle inskriptiolle kaikkien muiden voiman. Tämä voima ei ole kuitenkaan peräisin merkkien universumista, vaan se syntyy yhteensopivuudesta, optisesta koherenssista ja standardoinnista muiden inskriptioiden kanssa, joista jokainen on kytkeytynyt toiseensa verkoston tai lukemattomien erilaisten verkostojen kautta.

4. HYPERSTRUKTUURI-TUTKIMUSRYHMÄN VAPAA OHJELMISTOPROJEKTI SOSIOLOGIAN TUTKIMUSKOHTEENA

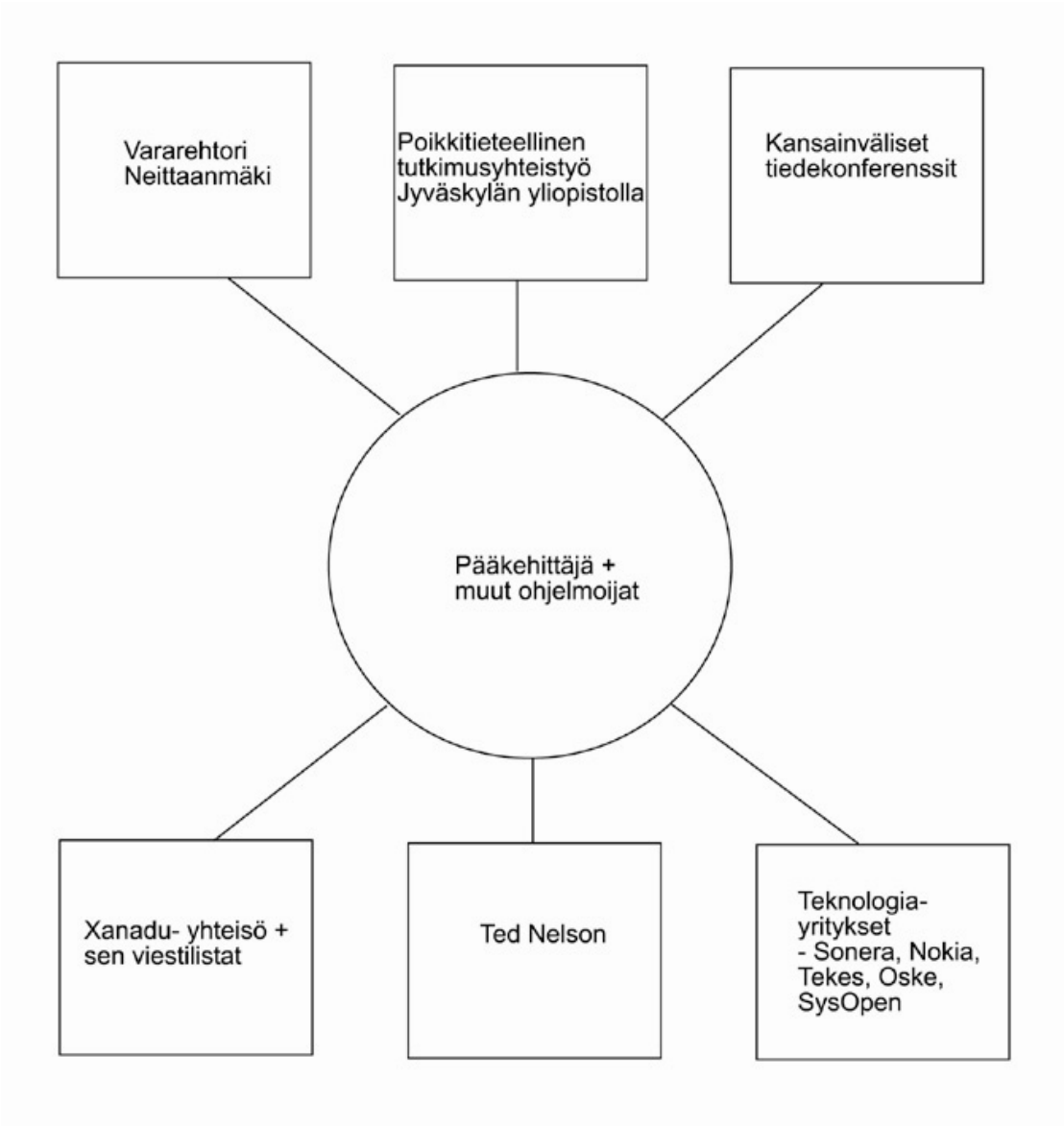
Tutkimuksessani on havainnoitu ohjelmistokulttuuria kahdenlaisesta näkökulmasta. Ohjelmistoprojektin käytäntöjä on havainnoitu ohjelmistoprojektin fyysisissä tiloissa, joissa on seurattu ohjelmoijien arjen ohjelmistokäytäntöjä ja haastateltu ohjelmoijia. Toisaalta ohjelmistoprojektia on havainnoitu seuraamalla viestiliikennettä sen moninaisilla kehitysfoorumeilla sekä Irkkiä, tietovaraston dokumenttien ja kotisivustojen kehittymistä vuosien 2001–2004 aikana. Seuraavaksi kuvaan miten ohjelmistoprojekti pyrki luomaan vapaata kehittäjäyhteisöä ja miten vaihdannan muodot projektissa syntyivät. Tutkimusryhmän toiminnalla oli kaksinainen luonne: tieteellisen tutkimus ja

teknologisen innovaation kehittäminen yhdistyvät vapaassa ohjelmistoprojektissa. Hypertekstuaalisen työympäristön visuaalista ulkoasua ja käytettävyyttä kehitettiin poikkitieteellisessä yhteistyössä mm. kognitio- ja viestintätieteilijöiden kanssa. Ryhmän tutkimustyössä oli vaikutteita kognitiivisesta psykologiasta ja hahmopsykologiasta, joiden pohjalta ryhmän tutkijat kehittivät ohjelmistonsa visuaaliseen toimintaan liittyviä ohjelmistoideoita. Pienimuotoisia käyttäjätestauksia tehtiin käyttöjärjestelmän visuaalisen ulkoasun parantamiseksi ja tieteellistä tutkimustyötä varten. Ryhmässä työskenteli alkuvuosina pääkehittäjä Tuomas Lukan johdolla kuusi tietotekniikan opiskelijaa ohjelmoijina. Projektin rahoittajia vuosien varrella Jyväskylän yliopiston lisäksi olivat Sonera, Nokia, SysOpen, Tekes ja Osaamiskeskus. Yksi projekteista, johon ryhmä osallistui oli Perheportaali-tutkimushanke, jonka tarkoituksena oli suunnitella viestintäväline perheiden käyttöön monitieteellisenä yhteistyönä yhdessä teknologia-yritysten kanssa (Siitonen, Sivunen, Feldt, Neittaanmäki M., Metsäpelto ja Pulkkinen 2002).

Seuraavaksi esittelen projektin työkulttuuria niin työtiloissa kuin Internet-välitteisillä foorumeilla. Ohjelmistoprojekti on vapaa projekti, mutta sen kotisivut, tietovarasto ja viestilistat ovat aina jonkun omistamina palvelimella. Palvelimen välittämänä projektin moninaiset sivustot ja niiden julkiset foorumit ovat esillä Internetissä. Näkökulmani hahmottaa ohjelmistoryhmän organisoitumista ja toimintaa on valita tutkimusyksiköksi vapaan ohjelmistoprojektin verkostoituminen ja yhteistyösuhteet. Yhteistyöverkossa toimijat täydentävät toistensa osaamista komplementaarisesti (Miettinen, Lehenkari, Hasu ja Hyvönen 1999). Sosiaalisen ilmiön tasolla pyrin hahmottamaan projektin kaksinapaisuuden: (1) miten projekti toiminnassaan verkostoitui institutionaalisena toimijana sekä (2) miten projektin Internet-välitteinen verkostoituminen tapahtui vapaana ohjelmistoprojektina. Giddensin (1990, 21) mukaan sosiaaliset suhteet jälkimodernissa maailmassa on nostettu ylös paikallisesta vuorovaikutuksen kontekstista ja nämä uudenlaiset sosiaaliset suhteet rakentuvat käsitteellistettäviksi ajan ja tilan uudenlaisiin vuorovaikutuksellisiin ympäristöihin.

Kaavio 1 kuvaa Hyperstruktuuri-tutkimusryhmän alkuvaiheiden yhteistyöverkostoa. Yhteistyöverkostossa on toimijoita, joiden lisäksi kaaviossa on kuvattu kaksi toimijarypästä. Poikkitieteellinen tutkimusyhteistyö käsitti yhteistyötä eri tieteenalojen tutkijoiden kanssa Jyväskylän yliopistolla. Ohjelmistotekniikan laitoksella toiminut

Hyperstruktuuriryhmä harjoitti yhteistyötä psykologian, kognitiotieteen ja viestintätieteen tutkijoiden kanssa.



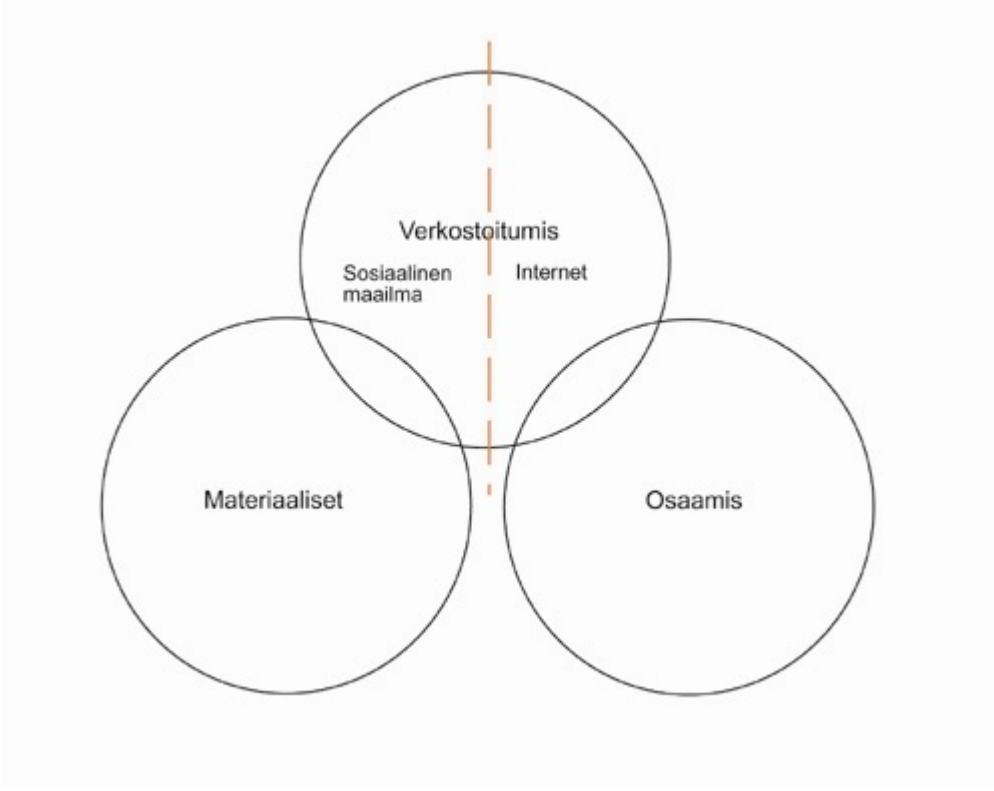
Kaavio 1: Hyperstruktuuri-tutkimusryhmän alkuvaiheiden yhteistyöverkosto

Ryhmän jäseniä osallistui vuosien aikana erilaisiin kansainvälisiin konferensseihin[30]. Tutkimuksen historialuvut kertovat enemmän yhteistyötoimijoiden roolista projektissa.

Tietokonevälitteisessä toiminnassa lähden ajatuksesta, että yliopistolla toimiva vapaa ohjelmistoprojekti koostui kolmenlaisista sen toiminnan edellyttämistä resurssisuhteista: (1) materiaaliset resurssit, joita ovat yritysten taloudellinen tuki (esim. ohjelmoijien palkka, yliopiston tarjoamat tilat sekä projektin pöytätietokoneet ja kannettavat sekä

[30] Konferensseja olivat Hypertekstiin liittyvät kansainväliset konferenssit vuosina 2001–2004 sekä Information Visualization konferenssit vuosina 2002–2003.

muut tutkimustyön edellyttämät välineet, jne.) (2) projektin jäsenten erilaiset tiedolliset resurssit, joita he pyrkivät hyödyntämään projektissa vastavuoroisesti. Näitä kutsun osaamisresursseiksi sekä (3) verkostoitumisresurssit, joita projekti pyrkii lisäämään.



Kaavio 2: Vapaan ohjelmistoprojektin resurssityypit

Tieteellinen yhteistyö eri tahojen kanssa sekä yritysyhteistyö ovat instituutionaalista verkostoitumista. Toisenlainen verkostoitumisen resurssi vapaalle kehittäjäyhteisölle on tietokonevälitteisen viestinnän mahdollisuudet, joiden kautta uusia ohjelmoijia voi liittyä yhteiseen kehitystyöhön. Vapaille ohjelmistoprojekteille on tyypillistä, että toisilleen tuntemattomat ohjelmoijat liittyvät projekteihin ja muodostavat suurempia hajautettuja kehittäjäyhteisöjä, joihin osallistuu jopa satoja ohjelmoijia. Kaavio 2 kuvaa kuinka verkostoitumisresurssi koostuu kahdesta osatekijästä. Tieteellisen tutkimusryhmän työlle tärkeitä ovat sosiaalisen maailman ja instituutioiden verkostoituminen, mutta Internetin kautta tulevat uudet kehittäjät ovat tärkeitä ohjelmiston käyttämisen, kehittymisen ja toimivuuden kannalta. Erilaisten resurssityyppien luoma perusta mahdollistaa projektin olemassaolon sekä taloudelliset edellytykset sen toiminnalle.

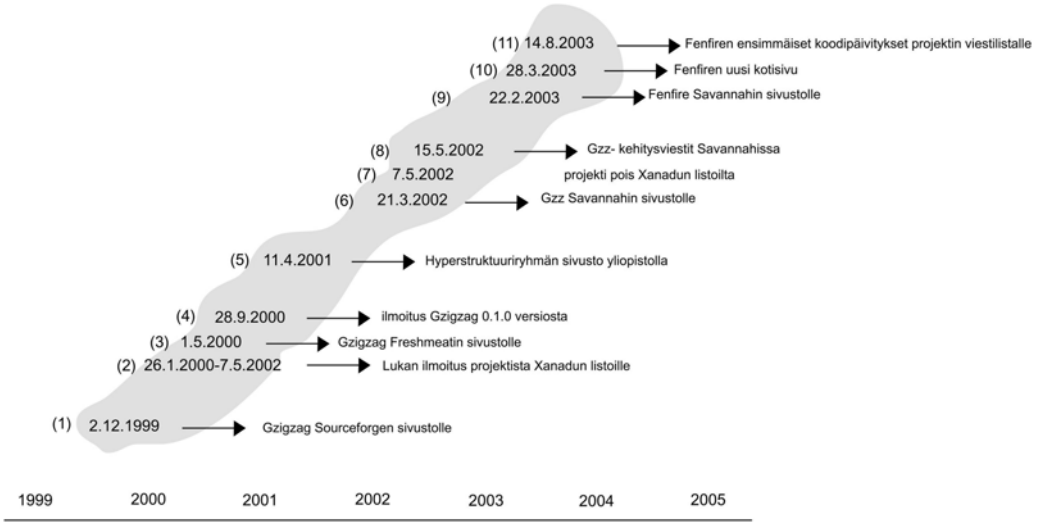
Ryhmän ohjelmistotyön arjesta on kerrottu luvussa 7. Esittelen sen vuoksi vain yleisluontoisesti projektin luonnetta. Hyperstruktuuri-tutkimusryhmän työtilat olivat ohjelmistotekniikan laitoksella Jyväskylän yliopiston Agorassa. Projektin käytössä oli kaksi huonetta ja avautuva työtila. Yhdessä huoneessa oli työtilat pääkehittäjälle ja toisessa kolmelle ohjelmoijalle. Käytännössä ohjelmoijat työskentelivät muutaman metrin päässä toisistaan. Pääkehittäjä työskenteli huoneessaan muutaman päivän kuukaudesta, joten silloin ryhmä piti kokouksia kasvokkain ja sopi työtehtävistä. Kehitystyö tapahtui pääasiassa Internet-välitteisesti. Ryhmän pääkehittäjä oli erikoistutkija Tuomas Lukka. Lisäksi ryhmässä työskenteli yksi Comas-tutkijakoulun[31] opiskelija. Ryhmän ohjelmoijat olivat titteleiltään tutkimusavustajia, jotka oli palkattu projektiin.

Projektin kehitysfoorumit ilmenivät eri tavoin Internetin välityksellä. Foorumeja on kahdenlaisia. Toiset ovat avoimien ohjelmistoprojektien sivustoja, jonne projektit voivat rekisteröityä ja säilyttää tietojaan CVS:ssä. Toisaalta projektilla on omat kotisivustonsa, joita voidaan kutsua välittyneiksi kvasi-vuorovaikutuksellisiksi paikoiksi (mediated quasi-interaction)[32]. Niiden vuorovaikutus on samankaltainen kuin median tiedottava tyyli ja ne ovat luonteeltaan monologisia. Kotisivustot kertovat projektista tietoja, kuten kuinka projektiin voi osallistua, mutta niillä ei ole suoraan vuorovaikutusareenoja. Kaavio 3 kuvaa projektin toimintaa Internet-ympäristössä. Esittelen seuraavassa lyhyesti kaaviossa 3 kuvatun Hyperstruktuuriryhmän toiminnan Internetin foorumeilla 11 keskeisen päivämäärän listana.

Gzigzag rekisteröitiin Sourceforgen sivustolle 2.12.1999 (1)[33]. Lukka ilmoitti projektin kotisivusta Xanadun keskustelulistoilla Zigzag-kehittäjille suunnatulla listalla 26.1.2000 (2)[34]. Freshmeat-sivustolle perustettiin Gzigzag-projektille sivu 1.5.2000 (3)[35]. Antti ilmoitti Gzigzagin 0.1.0 versiosta Zigzag kehittäjien päälistalla 28.9.2000 (4)[36]. Jyväskylän yliopiston sivustoille perustettiin Hyperstruktuuri-ryhmälle oma sivu, joka sisälsi

[31] Comas on lyhenne tutkijakoulun nimestä Graduate School in Computing and Mathematical Sciences.
[32] Käsitteestä tarkemmin, katso Thompson (1995, 82-87). Thompson tekee jaon kasvokkaiseen vuorovaikutukseen (face-to-face interaction), välittyneeseen vuorovaikutukseen (mediated interaction) ja välittyneesen kvasi-vuorovaikutukseen (mediated quasi-interaction).
[33] http://sourceforge.net/projects/gzigzag/, tarkistettu 8.5.2006.
[34] http://xanadu.com.au/mail/zzdev/msg02154.html, tarkistettu 8.5.2006
[35] Sivu ei enää käytössä, Freshmeatin pääsivu http://freshmeat.net
[36] http://xanadu.com.au/mail/zigzag/msg00280.html, tarkistettu 8.5.2006

ryhmän erilaisia tiedostoja 11.4.2001 (5)[37]. Gzz rekisteröitiin Savannahin sivustolle 21.3.2002 (6)[38]. Nelsonin mielestä projektin yksityiskohtaisista kehitysasioista ei pitäisi puhua Xanadun listalla, siksi projekti siirsi viestiliikenteensä pois Xanadun listoilta 7.5.2002 (7)[39]. Gzz-kehitysviestilistalle ryhmän ensimmäiset viestit lähetettiin Savannahissa 15.5.2002 (8)[40]. Fenfire-projekti rekisteröitiin Savannahin sivustolle 22.2.2003 (9)[41]. Fenfire-kotisivusto perustettiin 28.3.2003 (10)[42]. Fenfire siirtyi pois Gzz-sivustolta ja koodipäivitykset tehtiin uuteen tietovarastoon tehtiin Fenfiren-commits listan kautta 14.8.2003(11)[43].

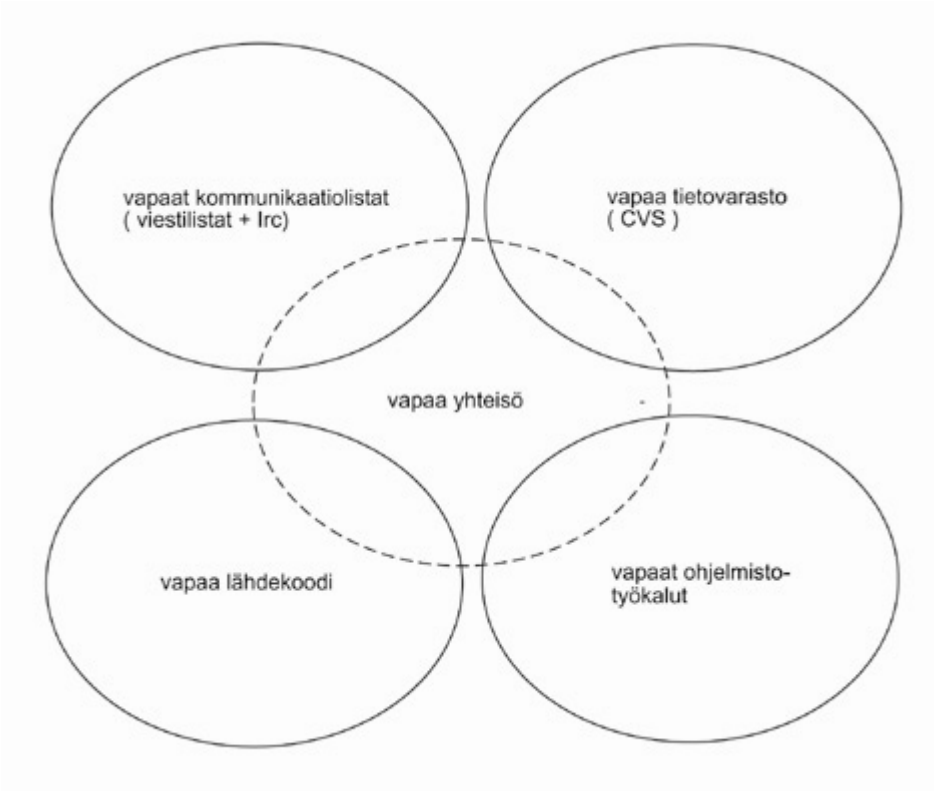


Kaavio 3: Hyperstruktuuri-tutkimusryhmän toiminta Internetissä

Miten ohjelmistoprojektin vapaus ja työn luonne sitten ilmenivät erilaisissa yhteyksissä? Ohjelmiston vapaus esitettiin ja perusteltiin erilaisissa tapahtumissa sekä julkaisuissa, jossa ohjelmistoprojekti oli esillä. Ohjelmistoprojektin arjessa ohjelmiston vapaus ilmeni ryhmän jokapäiväisissä työ- ja viestintäkäytännöissä.

---

[37] http://himalia.it.jyu.fi/, tarkistettu 8.5.2006
[38] http://savannah.nongnu.org/projects/gzz/, tarkistettu 8.5.2006
[39] http://xanadu.com.au/mail/zzdev/msg01577.html, tarkistettu 8.5.2006
[40] http://lists.gnu.org/archive/html/gzz-dev/2002-05/msg00000.html, tarkistettu 8.5.2006
[41] http://savannah.nongnu.org/projects/fenfire/, tarkistettu 8.5.2006
[42] http://fenfire.org/, tarkistettu 8.5.2006
[43] http://lists.gnu.org/archive/html/fenfire-commits/2003-08/msg00000.html, tarkistettu 8.5.2006



Kaavio 4: Hyperstruktuuriryhmän vapaat ohjelmistotekniset toimintatavat

Kaavio 4 kuvaa Hyperstruktuuriryhmän vapaan ohjelmistoprojektin teknisiä ulottuvuuksia ja sitä, miten kehittämisen vapaa malli ilmeni projektissa. Vapaat kommunikaatiolistat ja ohjelmistotyökalut sekä vapaa tietovarasto ja lähdekoodi mahdollistivat toimintaympäristön vapaalle yhteisölle. Ne olivat tekniset edellytykset vapaalle projektille, mutta projektin sosiaalinen ulottuvuus oli saada uusia jäseniä kehittäjäyhteisöön.

Ohjelmoijien työ tapahtuu pääasiassa virtuaalisesti. Virtuaalisella työllä tarkoitan työtapaa, joka ei tapahdu kasvokkain vaan tietokonevälitteisenä viestintänä. Informaationaalisen aikakauden yksi keskeinen muutos on, että ihmisten välinen viestintä tapahtuu tietoverkoissa ei-kasvokkaisessa vuorovaikutuksessa. Ihmisten välisiä puheakteja ei tässä kontekstissa ole. Vuorovaikutus ilmenee viestintäakteina, jotka sisältävät erilaista semioottista koodia kuten puheaktit ihmisten arkielämässä. Ohjelmistotyön arki ilmeni erilaisten elektronisten foorumien kautta. Ohjelmistoryhmän sosiaalisessa viestinnässä yhdistyivät fyysinen työ ryhmän toimistolla ja virtuaaliset työtavat. Viestinnän aktit, kielenkäyttö ja työtavat mukautuivat välineisiin, joilla ryhmä työskenteli. Ohjelmistotyössä työpaikalla on myös kasvotusten tapahtuvia neuvotteluja. Katkoksia yhteistyössä

sekä yllättäviä teknisiä ongelmia ilmeni, kun yliopiston verkko ei yllättäen toiminutkaan. Havainnointi osoitti, että tietovarastolla on keskeinen rooli ohjelmistotyössä. Hyperstruktuuriryhmän tietovarastoon (CVS) ryhmä kokosi erilaista dokumentaatiota. Omissa tiedostoissaan tietovarastoissa olivat lähdekoodi, päiväraportit, tieteelliset käsikirjoitukset, raportit rahoittajille ja kaikki dokumentaatio mitä projektin yhteydessä luotiin. Dokumentit olivat projektin jäsenille saatavilla tietovarastosta, sekä kaikille muille kiinnostuneille Internetin välityksellä.

4.1 Tutkimuskysymykset

Tutkimuskysymykset jäsentyvät aineistolukujen mukaan. Koko ohjelmistoprojektin elinkaarta koskeva tutkimuskysymys käsittelee Hyperstruktuuriryhmää ja sitä, miten projekti toimi vapaana Internet-välitteisesti. Miksi projekti ei onnistunut kehittämään toimivaa ohjelmistotuotetta? Mitä ryhmä saavutti innovaatioiden kehittämisessä ja millaisia tutkimustuloksia ryhmä saavutti? Mitä ohjelmoijat saivat projektista? Näihin kysymyksiin vastaan luvun 8 lopussa ja pohdintaa-osiossa. Analyysissäni kiinnitän huomioni siihen, kuinka tutkimuskysymysten teemat voidaan ymmärtää tutkimuksen teoreettisten käsitteiden eli luottamuksen, lahjatalouden ja rekursiivisen yhteisöllisyyden näkökulmasta. Lisäksi esitän tutkimukseni tuloksena syntyneen mallin, jonka esitän hypoteesina siitä, kuinka vapaisiin ohjelmistoihin liittyvää toimintaa voidaan rekursiivisen yhteisöllisyyden näkökulmasta ymmärtää.

Lisäksi tutkimuksessani on tutkimuksen aineistolukuihin perustuvia tutkimuskysymyksiä. Aineistolukuihin perustuviin tutkimuskysymyksiin vastaan sen luvun lopussa, jota tutkimuskysymys koskee. Jokaiseen aineistolukuun yhdistetyt kaksi erityistä kysymystä hahmottavat ryhmän historian keskeisiä tapahtumia.

Tutkimuskysymykset ovat seuraavat:

1. Luvussa 5: Miten projekti käynnistyi ja miten ensimmäiset ohjelmoijat liittyivät projektiin? Miten Nelsonin ideat otettiin vastaan?
2. Luvussa 6: Mistä näkökulmista Stallman käsittelee projektia? Miten keskustelussa yhdistyy vapaan ohjelmistoprojektin ja toimivan ohjelmisto-hyödykkeen näkökulmat?

3. Luvussa 7: Miten ohjelmistoprojekti toimii käytännössä arkipäivän työssä? Miten projektin vapaa toimintamalli ilmeni projektin käytännöissä?
4. Luvussa 8: Miten ryhmä toimi patenttiongelmien yhteydessä? Miksi ryhmän ja Nelsonin yhteistyö loppui?

4.2 Aineisto

Tutkimuksen aineistoksi on vuosina 2001–2005 ohjelmistoprojektista kerätty kolmenlaista laadullisen tutkimuksen aineistoa. Tutkielman metodologinen tutkimustapa on etnograafinen tapaustutkimus. Tutkielman erilaiset aineistotyypit on esitelty taulukossa 1. Haastatteluaineistoa projektista on kerätty toukokuusta 2001 lähtien. Ensimmäisen kerran ryhmän jäseniä haastateltiin joulukuussa 2001. Toukokuussa 2002 ryhmän työtä seurattiin kahden kokonaisen työpäivän ajan. Ryhmän jäsenet tavattiin vähintään kaksi kertaa vuodessa vuosien 2002–2003 aikana heidän työpaikallaan Jyväskylän yliopiston Agorassa projektin tiloissa. Viimeinen ryhmän jäsenten tapaaminen ja kokouksen havainnointi tapahtui maaliskuussa 2004. Muuta havainnointi-aineistoa on kerätty ryhmän kokouksista, tapaamisista ja julkisista esiintymisistä. Lista haastattelu- ja havainnointiaineistosta on esitetty tutkimuksen liitteessä 2. Henkilö-kohtaisia sähköpostiviestejä vaihdettiin informanttien, ryhmän ohjelmoijien, kanssa. Sähköpostiviesteja vaihdoin vuosien 2001–2005 aikana viiden ryhmän ohjelmoijan kanssa. Kahden informantin kanssa sovin etukäteen vierailu- ja haastatteluajankohdista. Ohjelmoijien kanssa vaihdetut sähköpostit koskivat ryhmän ohjelmistoprojektin tilannetta ja ohjelmoijien omia työtehtäviä. Haastattelujen yhteydessä vietin yleensä koko päivän projektin tiloissa ja keräsin havainnointiaineistoa projektin luonteesta sekä tutustuin ryhmän ohjelmoijiin. Ohjelmistoryhmän jäseniin ystävystyminen ja yhteydenpito mahdollistivat tiheän kuvaamisen tutkimuksen etnografiselle tutkimusotteelle.

Toisen projektin etnografiaan liittyvän aineiston muodostivat projektia koskevat dokumentit. Näitä dokumentteja on kahdenlaisia: Internet-dokumentit ja kirjalliset dokumentit. Projektin tietokonevälitteinen kommunikaatio ja sen dokumentaatio Internetissä oli kaikkien saatavilla. Projektin erilaisilla kotisivuilla oli uutisia, tietoa ja tiedostoja projektista.

Taulukko 1: Ohjelmistoprojektista kerätyt aineistot vuosina 2001-2005

Aineisto	tyyppi	kuvaus	määrä
1. Haastattelut	Ryhmän ohjelmoijien haastatteluja	Ohjelmoijien haastattelut ryhmän tiloissa vuosina 2001 –2004	15
2. Osallistuva havainnointi + julkiset esiintymiset	Ryhmän kokouksien ja esiintymisten havainnointia	Ryhmän kokouksien, ja ohjelmiston esittelytilaisuuksien havainnointia	10
3. Internet-aineistot	Ryhmän erilaiset Internet-sivustot	Erilaiset ”lokit” , sähköpostilistat, Cvs ja kotisivustot Internetissä	Ryhmän työn seuraaminen Internetissä 2001–2005
4. Muu aineisto ryhmän työstä	Ryhmän kirjallinen dokumentaatio + henkilökohtaisia sähköpostiviestejä ryhmän jäsenten kanssa	Erilaiset materiaaliset dokumentit ryhmän työstä ja yhteydenpito sähköpostitse	30 henkilökohtaista sähköpostiviestiä + ryhmän tutk.suunnitelma Akatemialle

Keskeinen projektin dokumentaation keskus on CVS (Concurrent Version Systems), joka sisälsi projektin avoimen dokumentaation Internetin kautta. Projekti tallensi reaaliaikaisen viestintänsä (IRC) tiedostoiksi vapaasti luettavaksi Internetin välityksellä. Projektin materiaalisilla dokumenteilla tarkoitan projektin julkisuuskuvaan liittyviä lehtijuttuja ja muita vastaavia ryhmän omia kirjallisia dokumentteja. Projektin jäsenet antoivat luottamuksellisesti tutkimuskäyttöön esim. tutkimussuunnitelman Suomen Akatemian rahoitushausta.

4.3 Aineiston tulkinnasta ja tutkimusprosessista

Sosiologisen tutkimustavan metodiikkaan suuresti vaikuttaneet Georg Simmel ja Erwing Goffman asettivat yhteiskunnan tutkimiselle metodologisen ohjenuoran tutkia sen vuorovaikutuksen kudoksia ja niiden aiheuttamaa muutosta, joissa uudet sosiaalisen toiminnan muodot ilmenisivät. Goffman lainasi Simmeliä väitöskirjansa alkutekstissä tuodakseen ilmi samaistumisensa Simmelin metodologiseen näkemykseen sosiologian tutkimustavasta (Manning 1992, 19-21). Keskeistä Simmelin sosiologiselle metodille ja sen tutkimustavoille oli ajatus, että koska sosiaalinen elämä on jatkuvassa muutostilassa,

sen kuvaaminen on vaikeata. Pienetkin muutokset sosiaalisen elämän keskellä vaikuttavat sosiaaliseen elämään tämän keskuksen reunoilla. (Smith G., 1989).

Vapaassa ohjelmistotyössä tutkijan huomion kohde ei ole työn ja yhteisön mystifiointi ”*virtuaaliseksi*” yhteisöksi, vaan kohteeksi muodostui ohjelmistoprojektin havainnointi jatkuvassa prosessissa. Hyperstruktuuriryhmän vapaa toimintamalli esitettiin erilaisissa yhteyksissä ryhmän puheessa, toiminnassa ja käytännöissä. Kuinka tutkia siten vapaan ohjelmistotyön vuorovaikutuksen ja toiminnan kudoksia sosiologisesti? Tutkimus-prosessia jäsentävät selkeät osat: aluksi aiheen valinta, teoreettisen viitekehyksen muodostaminen ja tutkimusmetodien valinta, seuraavaksi aineiston keruu ja sen koodaaminen sekä muistiot ja lopulta aineiston analyysi muodostuu suhteessa tutkimuksen teorioihin. Tutkimusprosessin osat ovat hyvinkin erillisiä, ja siksi tutkimustyö käytännössä tapahtuu lomittaisessa järjestyksessä, joka tuo tutkijalle usein hedelmällisiä havaintoja. Laadullisen aineiston analyysissa tutkija muodostaa teemoja yhdistämällä nämä omaan ymmärrykseensä aineistosta. Layderin (1998) adaptiivisessa teoriassa (adaptive theory) tutkimusprosessi muistuttaa aineistokeskeistä tutkimusotetta. Tutkiminen on prosessinomaista vuoropuhelua teorian, aineiston ja kirjoitusprosessin välillä. Denzin (2004, 2-3) puolestaan kuvaa onlinetutkijan tee-se-itse tutkijaksi (bricoleur), joka yhdistelee erilaisia tiedonkeruun tapoja, käyttää uusia tutkimusvälineitä ja kehittelee vanhoja tutkimusmetodeja muokkaamalla uudenlaisia ongelmanratkaisun keinoja empiirisen tutkimustyöhön. Online-välitteiset tekstuaaliset viestit ovat eksplisiittistä tietoa erilaisista Internet-perustaisista toimintatavoista: sähköpostiviestit, Irkki-keskustelut ja kotisivujen uutisosiot kertovat tarkan ajan milloin viesti on lähetetty. Nämä viestien kokoamisen tavat luovat erilaisia synkronisia ja asynkronisia ”*lokikirjoja*” ohjelmistoyhteisön sosiaalisesta maailmasta. Ohjelmistotuotannon prosessissa ohjelmoijan toiminnasta jää jälki erilaisille tiedon paikoille, jotka ovat luettavissa Internet-välitteisesti. Ohjelmistotyön välineiden ja niiden dokumentaation kautta työ ja sen materiaaliset resurssit pyritään valjastamaan hyödyllisiksi ja käytettäväksi laajemmalle, hajautetulle kehittäjien piirille. Samalla ohjelmoijille muodostuu ohjelmistoyhteisössään sosiaalisten suhteiden ja kommunikaatiotapojen verkosto työyhteisöksi.

Tutkimuksen aineistotriangulaatio on muodostunut erilaisista laadullisista tutkimusaineistoista. Tutkimuksen lähtökohtana on käyttää erilaisia ohjelmistoprojektin

dokumentaationäytteitä, ohjelmoijien haastattelunäytteitä tai muuta projektiin liittyvää dokumentaatiota kuvaamaan vapaan ohjelmistoprojektin ideologiaa ja käytäntöjä. Niiden avulla kuvataan projektin kehittämiä ideoita ohjelmistoinnovaatioista ja tutkimustyöstä sekä luodaan tulkintoja siitä miten ohjelmoijat kokivat oman suhteensa vapaan ohjelmiston tekemiseen. Ensimmäisellä haastattelukerralla jokaiselta ryhmän ohjelmoijalta kysyttiin taustoja siitä miten hän oli päätynyt projektiin, mikä hänen roolinsa oli projektissa ja miten hän koki projektin tilanteen. Seuraavilla haastattelukerroilla ohjelmoijia haastateltiin projektin sen hetkisen tilanteen mukaan. Haastatteluteemoina olivat vapaa projekti, projektin sosiaalinen organisoituminen, työnjako ja miten projektin ideat sekä yhteistyö muihin toimijoihin kehittyivät. Haastattelujen yhteydessä vietin projektin tiloissa lähes koko työpäivän havainnoimalla ohjelmoijien työtä. Päivän aikana tein paikalla olevien ohjelmoijien kanssa 20–30 minuutin henkilöhaastattelut. Kokouksissa ja esittelytilaisuuksissa toimin havainnoijana, enkä osallistunut keskusteluihin vaan nauhoitin tilaisuudet joko ääni- tai kuvanauhalle. Erilaisten laadullisten aineistojen kautta on myös siteerattu projektista tehtyjä lehtikirjoituksia. Niistä välittyy kuva siitä miten projekti haluttiin esitellä eri yhteyksissä. Aineistona on käytetty luvussa 6 Richard Stallmanin ja Gzigzag-ryhmän välistä keskustelua. Jon Hallin puhetta vapaisiin ohjelmistoyhteisöihin liittyvästä terminologiasta on käytettä runkona johdannossa.

4.4 Tapaustutkimuksen validiteetti ja yleistäminen

Kenttätyön metodi koostuu erilaisten aineistojen keräämisestä ja niiden käytöstä tutkimuksessa. Tutkijan erilaisilla metodeilla keräämät dokumentit muodostavat empiirisen materiaalin. Moninaisten aineistojen käyttö muodostaa aineistotriangulaation, jolla voidaan parantaa ja luoda monipuolisempi kuva tutkittavasta ilmiöstä. Tutkimustapani on yksittäinen tapaustutkimus (single case study), jonka aineisto on kerätty kenttätyömetodilla. En ole tietoinen vastaavasta esimerkkitutkimusta, jossa olisi osallistuvalla havainnoinnin metodilla ja laajalla kenttätyöllä havainnoitu tutkijaryhmän työskentelyä liittyen vapaaseen ohjelmistotoimintaan, ja siksi luokittelen tapauksen ainutkertaisen tapaustyypin kategoriaan.

Aineiston laadullinen tulkinta perustuu huolelliseen tapauskuvaukseen. Tutkimuksen aineiston ja keräämismenetelmien tarkalla raportoinnilla on pyritty osoittamaan tutkimuksen luotettavuus. Tärkeintä on se, että tutkija selvittää keskeiset kenttätyöhön liittyvät menetelmät, toiseksi tutkijan tulisi selvittää miten aineistoja on käytetty ja miten tutkimuksessa on päädytty tekemään johtopäätöksiä aineistosta käsin. Lukijan täytyy pystyä seuraamaan tutkimuksen tekemisen prosessia tutkijan tekstissä kuvaamien tietojen perusteella. Tutkimuksen sisäisen validiteetin parantamiseksi voidaan käyttää monenlaisia aineistoja, jotka on kerätty erilaisilla havainnointitavoilla.

Grönfors (1982) toteaa, että laadullisessa tutkimuksessa on usein kiinnostuttu ainutlaatuisuudesta, ei yleistettävyydestä. Huolellinen kuvaus tutkimuskohteesta lisää tapauksen ulkoista validiteettia ja sen mahdollista vertailtavuutta muihin tapaustutkimuksiin koskien vapaita ohjelmistoprojekteja. Etnografisen tutkimuksen validiteetti perustuu myös tutkimusprosessin yksityiskohtaiseen kuvaamiseen. Ohjelmistoprojektin historia on esitetty retrospektiivisesti vuoteen 2001 asti, kesästä 2001 asti projektista on kerätty aineistoa eri tavoin. Keskeinen havainnointipaikka tutkimuksessa oli Internet, joka voidaan ymmärtää samanaikaisesti tutkimuksen kohteena, tutkimusvälineenä ja aineiston lähteenä. Internet-aineisto on ollut vapaasti saatavilla julkisilla foorumeilla. Tutkimuksessani on pyritty noudattamaan hyvää Internet-tutkimuksen eettistä tapaa, joka perustuu Kuulan (2006, 169-199) käsitykseen eettisestä Internet-tutkimuksesta. Tutkimuksessani olen pyrkinyt myös refleksiiviseen tutkimusotteeseen käyttämällä erilaisia empiirisiä aineistoja lukujen runkona sekä teoreettisessa osuudessa käymällä vuoropuhelua ohjelmoijia ja heidän kulttuuriaan koskevien erilaisten teoreettisten suuntausten välillä.

Laadullisen tutkimuksen yleistäminen on ongelmallista, koska kysymys mitä, miten ja mihin yleistetään aiheuttaa yleistämisen käsitteelle erilaisia tutkimusotteen metodiikkaan liittyviä rajoituksia. Tärkeä kriteeri yksittäisen tapauksen kuvaamiselle on se, että siitä saatavien tutkimustulosten avulla voidaan rikastaa teoriaa ja tutkia sen rajoja. (Koskinen, Alasuutari ja Peltonen 2005, 166-168, 263-272). Yksi laadullisen tutkimuksen elementti on se, että tutkija itse asiassa aina tutkii tapausta tai tapauksia, ja hän itse määrittelee mikä on hänen tutkimuksessaan tapaus. Sisäinen validiteetti tapaustutkimuksessa perustuu aineiston ja tapahtumien mahdollisimman tiheään kuvaamiseen. Williams (2001) toteaa, että kuvailevassa etnografisessa tutkimusotteessa itse tutkimusprosessin

aikana sekä tapauksen kuvailun aikana esitetään erilaisia väitteitä kohteesta, jotka ovat luonteeltaan yleistäviä. Williams (2002) tarjoaa laadullisen tutkimuksen kuvailevalle tutkimustavalle moderatiivisen yleistämisen (moderatum generalization) käsitettä. Williamsin mukaan moderaatiivinen yleistäminen on peräisin kulttuurisesta johdonmukaisuudesta ja se luo perustan induktiiviselle päättelylle elämismaailmastamme käsin. Payne ja Williams (2005) kehittävät yhdessä ajatusta moderatiivisesta yleistämisestä eteenpäin. He kuvaavat, kuinka tämä yleistämisen tapa on moderatiivinen kahdella tapaa. Ensinnäkin koskien tutkimuksen kohdetta, johon moderatiivinen tulkinta on kohdistettu. Yleistäminen ei siis toimi laajamittaisesti kulttuureissa sosiologisina väittäminä, eikä yleistäminen toimi pitkän ajallisen jakson ja kulttuurien välisten etäisyyksien välillä. Toiseksi yleistäminen on vain moderatiivisesti pysyvää, mikä tarkoittaa sitä, että tulkinnan on oltava avoin muutokselle. Tämä näkökulma on tärkeä, koska se korostaa moderatiivisen yleistämisen hypoteettista luonnetta. Moderatiiviset yleistykset ovat kokeiltavia väittämiä, jotka voidaan vahvistaa tai kumota tutkimuksellisen evidenssin lisääntyessä. Payne ja Williams (2005, 306-307) luettelevat useita ontologisia tasoja, joilla yleistäminen voi tapahtua sosiologisessa tutkimuksessa[44]. Tutkimukseni lopussa esitän yleistävän hypoteesin, joka koskee vapaan ohjelmistotyötoiminnan *sosiaalista rakennetta* yhteiskunnallisessa todellisuudessa.

5. NELSONIN JA LUKAN YHTEISHANKKEEN ALKU

Tuomas Lukka tutustui ensimmäistä kertaa Ted Nelsonin vallankumouksellisiin ideoihin tietokoneen käyttöliittymän kehittämisestä syksyllä 1998:

> Noin kaksi vuotta sitten luin verkosta Ted Nelsonin suunnitelmaa Zigzag-systeemille. Silloin se vaikutti ihan mielenkiintoiselle, mutta unohdin asian aika pian. Vuosi sitten luin samat sivut uudelleen ja silloin päässäni napsahti. On huomattava, että jotkut perinteisemmät tieteenharjoittajat varmaankin vääntäisivät tuosta vähän toisenlaisen napsahduksen - tieteen alalla uudenlaiset ajattelutavat eivät muitten saman alan edustajien kesken ole aina erityisen tervetulleita. Mutta kuitenkin, siis tajusin sillä hetkellä, että tuo design, Zigzag, voisi muodostaa pohjan paljon järkevämmälle tavalle käyttää tietokoneita. (Tuomas Lukka seminaariesitelmässään Agoran avajaisissa Jyväskylässä 22.9.2000)

Nelson ja Lukka tutustuivat sähköpostin välityksellä vuonna 1999. Sähköposti-viestien vaihto muuttui yhteistyöksi suunnitella Zigzag-ohjelmistoa. Lukka perusti Jyväskylän yliopistolla opiskelijaprojektina tutkimushankkeen *”Uudenlaiset tiedonhallinta-*

[44] Williamsin (2002, 138-139) käsitys metodologisesta pluralismista pyrkii jatkamaan Max Weberin kumoamaa nomoteettisen ja idiograafisen tutkimustavan eroa.

järjestelmät” kehittämään Zigzagia. Opiskelijatyöprojektissa toteutettiin Jyväskylän yliopiston tietotekniikan laitokselle Zigzag-tiedostorakennetta hyödyntävä ohjelma[45]. Lukka innostui projektin aikana toimimaan pääohjelmoijana. Lukan ja Nelsonin välille syntyi suullinen sopimus, että Lukka kehittäisi Zigzagista vapaata ohjelmistoversiota, johon Nelson säilyttäisi ”*directors cut*” oikeuden[46]. Lukka kertoi miten hän päätyi aloittamaan projektissa pääohjelmoijana :

> Mutta sitten tosiaan, sitten lähetin Tedille sähköpostia että haluaisin toteuttaa tämän ja sitten... Siinä oli alussa pikkusen väärinymmärryksiä. No sitten päädyttiin tähän, että rupeen töihin (-) Ja siitä sitten lähti. Ja tämä oli ensin opiskelijaprojekti. Mä halusin koettaa mitä opiskelijat saisi tästä aikaan ilman että itse teen, itse koodaan sitä. Mutta sitten niin kuin tajusin sen syksyn lopussa, että miten olisi pitänyt tehdä alusta lähtien. Sitten alettiin tehdä uutta versiota Javalla. Eka versio oli C++:lla.
> (Tuomas Lukan haastattelu 28.5.2001)

Lukka kutsui Nelsonin vierailemaan Jyväskylän yliopistolle ja tiivis yhteistyön vaihe alkoi. Nelson vieraili vuoden sisällä kolme kertaa Jyväskylässä: joulukuussa 1999, maaliskuussa 2000 ja elokuussa 2000[47]. Näiden kolmen vierailun seurauksena Lukka innostui toteuttamaan teknologisesti aivan erilaista paradigmaa edustavaa ohjelmistoa kuin pääasiassa loppukäyttäjien käytössä olevat tietokoneen käyttöliittymät, esim. Microsoftin Windows. Maaliskuussa 2000 Nelson vieraili Jyväskylässä kahden viikon ajan ja suunnitteli prototyyppiä Lukan kanssa. Vierailun aikana Nelson piti myös luentosarjan Xanadun perusideoista. Nelson ei itse osannut ohjelmoida, joten hän ja Lukka yhdistivät ideoitten toteuttamiseksi kaksi erilaista näkökulmaa. Nelson kertoi ideoistaan, jotka Lukka suunnitteli vapaassa ohjelmistoprojektissa toteutettavaksi. He kehittivät yhdessä suuntaviivat ohjelmistoprojektille, jossa Lukka toimisi teknisenä pääkehittäjänä. Maaliskuun vierailusta muodostui käännekohta Lukan käsitykselle

---

[45] Lukan johdolla sovelluksen toteuttivat Linux-ympäristöön C++ -ohjelmointikielellä sekä Gtk+, Gdk- ja GLib-kirjastoilla ja Berkeley-tietokannalla Heikki Maaranen ja Vesa Parkkinen.

[46] Ryhmän oma historiadokumentaatio tiedosto määrittelee Lukan ja Nelsonin sopimuksen perustuvan Nelsonin Directors Cut oikeuteen. Directors cut termi on peräisin elokuvakulttuurista ja tarkoittaa ohjaajan viimekätistä oikeutta omaan luovaan työhön. Ryhmän oma versio historiansa tärkeimmistä tapahtumista määrittelee sopimuksen luonteen seuraavasti: ”*He contacts Ted Nelson and after some emails, they agree that Tuomas can develop a free version, if Ted gets to do his "director's cut*". Verkossa http://himalia.it.jyu.fi/ffdoc/fenfire/history/Milestones.rst, tarkistettu 3.5.2006.

[47] Nelson luennoi ja työskenteli Jyväskylän yliopistolla 5.3–19.3.2000 ja 20.8–31.8.2000. Joulukuun 1999 tarkoista päivänmääräistä ei ole dokumentoitua tietoa. Vuoden 2000 elokuun vierailun yhteydessä Nelson antoi ainakin yhden haastattelun, joka on julkaistu Tekniikan Waiheita lehden numerossa 2 / 2002. Henry Oinas-Kukkonen: World Wide Webin kummisedän näkemyksiä ja kokemuksia uuden median maailmasta, sivut 16-26.

Nelsonin ideoista. Lukka ymmärsi Nelsonin ideoitten pohjalta[48], että dokumenttien kytkeytyvyys toisiinsa on avainasemassa visuaalisessa käyttöliittymässä :

> Mulla itsellä oli hyvin, hyvin rankkaa, rankkaa aikaa henkisesti silloin kun mä niinku menin läpi sen paradigman muutoksen…sen jälkeen huomasin että ajattelen asioista aika eri lailla, eli että se kytkeytyvyys on olennaista.(Tuomas Lukan haastattelu 28.5.2001)

Vuonna 2000 Gzigzag-projektin toiminta käynnistyi hankkeena Jyväskylän yliopistolla sekä Internetissä Xanadu-kehittäjien sähköpostilistalla[49]. 26. tammikuuta 2000 Lukka ilmoitti Gzigzag-projektin olemassaolosta keskeisellä Xanadu-projektin ylläpitämällä keskustelulistalla. Huhtikuussa 2000 projekti rekisteröitiin[50] Sourceforgen sivustolle Internetiin vapaaksi ohjelmistoprojektiksi. Free Software arvojen mukaisesti ohjelmiston kehittäjiksi kaikki olisivat tervetulleita[51]. Projekti sai nimen Gzigzag. Nimi muodostui Lukan nimettyä sen Gnu–akronyymin (Gnu`s not unix) mukaisesti erotukseksi Zigzagista[52]. Nelsonin maaliskuun ja elokuun 2000 kahden viikon pituiset vierailut Lukan isännöiminä olivat tiivistä yhteistyöskentelyn aikaa saada Gzigzag-projekti käyntiin. Lukan johtama projekti aloitti toimintansa Jyväskylän yliopiston Agoraan vuonna 2000 perustetussa tutkimuskeskuksessa, jonka perustana on poikkitieteellinen tutkimus ihmisen ja teknologian välillä. Ryhmä nimesi itsensä Hyperstruktuuri ryhmäksi[53]. Tutkimuksen painopiste oli ihmisen ja tietotekniikan suhde. Projektin alussa päämäärät jakautuivat kahteen osaan: projektin puitteissa tehtiin tieteellistä tutkimusta saman aikaan kun ohjelmistoa kehitettiin avoimen lähdekoodin ohjelmistoksi. Soveltava tutkimus sai alkunsa Soneran rahoittamassa Perheportaalihankkeessa, jossa tutkittiin tietotekniikan vaikutusta perheiden arkielämään[54]. Hankkeen tarkoitus oli Jyväskylän yliopiston ja Soneran yhteishankkeena selvittää miten viestintäteknologiaa voitaisiin hyödyntää perheen arjen hallinnassa. Gzigzag-projekti pyrki valmistamaan teknologisen

---

[48] Katso myös Raisa Korkin haastattelu Tuomas Lukasta: Tekoäly ja Xanadu lennättävät korkealle. Vaasan ylioppilaslehden artikkeli vuonna 2000 saatavana Internetistä, http://lipas.uwasa.fi/yolehti/06-00/23.html, tarkistettu 3.5.2006.

[49] http://www.xanadu.com.au/mail/zzdev/, tarkistettu 13.11.2005.

[50] http://sourceforge.net/projects/gzigzag/, tarkistettu 13.11.2005.

[51] Ryhmän ensimmäisessä artikkelissa mainitaan että kiinnostuneet voivat liittyä ryhmän Free Software projektiin ja sen sähköpostilistoille (Lukka ja Ervasti 2001, 150). Katariina Ervasti oli vuosina 2000–2001 ryhmässä työskennellyt hypertekstien tutkija, joka ei ohjelmoinut, mutta osallistui hypertekstuaa-listen innovaatioiden suunnitteluun.

[52] Keskustelu Lukan kanssa ryhmän nimen taustasta 8.3.2002, ei nauhoitettu.

[53] Tuomas Lukan ja Katariina Ervastin artikkeli Hyperstruktuuri-ryhmän tutkimustyöstä ilmestyi Cybertext yearbook 2000 kirjassa, sivut 141-151.

[54] Lukka ei kuitenkaan halunnut suunnata kaikkea kiinnostustaan tutkimukseen Jyväskylän yliopistolla vaan työskenteli myös puolipäiväisesti Modirum Oy:ssa tietotekniikka-alan konsultoijana ja tuotteiden rakentajana vuosina 1999–2002.

Perheportaali-sovelluksen koekäyttöön perheille. Gzigzagin ohjelmoijat suunnittelivat ja kehittivät yhdessä muiden tieteenalojen tutkijoiden kanssa prototyyppiä[55].

Luentosarjan jälkeen elokuussa Lukka kutsui kotiinsa tietotekniikan opiskelijoita vapaamuotoiseen tilaisuuteen, johon Nelsonin ja Lukan ideoista kiinnostuneet olivat tervetulleita vaihtamaan ajatuksia Lukan ja Nelsonin kanssa. Projektin alussa ohjelmoijat löytyivät yksi kerrallaan eri tavoin ja ensimmäiset opiskelijat työskentelivät projektissa kesällä 2000. Lukka kertoi haastattelussa siitä miten ensimmäiset tietotekniikan opiskelijat liittyivät projektiin mukaan. Hän kuvaa tätä käyttämällä Nelsonin vertausta:

> Niin, Ted, mä en muista ketä hän lainasi. Olisiko se ollut Ross Perot[56] tai joku tällainen sano että, ” *Eagles don't flog, we have to find them one by one*”. Ja tosiaan kaikki on löytynyt eri lailla. Ei oo, ei oo mitään sääntöä siinä. No tämähän on siis tosiaan erilaista kun jossain Open Source-projekteissa, koska niissähän se taas menee sillä lailla että ihmiset ottaa yhteyttä niin, tämä on enemmän tutkimusprojekti tässä yhteydessä. (Tuomas Lukan haastattelu 28.5.2001)

Projektin lähtökohtana oli vapaa ohjelmistokehitys, johon ohjelmoijat sitoutuivat. Vapaan ohjelmistokehityksen metodeilla projekti toteutti ohjelmistoideoita, ohjelmiston lähdekoodia ja sovelluksia Free Software-projektissaan. Free Software-projekti tarkoitti sitä, että ryhmä käyttäisi työvälineinään vapaita ohjelmistoja. Lisäksi kehitettävä ohjelmisto sen sovellutuksineen oli muiden saatavilla Internetissä vapaiden ohjelmistolisenssien käytäntöjen mukaisesti. Lukka kertoi avoimen ohjelmisto-kehitykseen sitoutumisen syntyneen ryhmän jäsenten keskuudessa heidän omasta vapaasta valinnastaan:

> On yhteistä niin kuin tällainen tietynlaisen avoimen, siis avoimen siis vapaan softan filosofia. Eli se että siis kaikki käyttää Linuxia ja niin edespäin. Vaikka se ei oo kyllä mitenkään pakko tässä projektissa, koska me käytetään Javaa eli tää toimii yhtä hyvin myös muissa käyttöjärjestelmissä, mutta se on vaan nyt sattunut näin. Ideologisista syistä varmaan. (Tuomas Lukan haastattelu 28.5.2001)

Syyskuussa 2000 Lukka osti projektille kotisivun Internetistä[57]. Ryhmän kotisivulla tiedotettiin uutisosiossa ohjelmiston edistymisestä. Ryhmän kotisivulta oli ladattavissa myös uusin versio ohjelmistosta. Aika syksystä 2000 kevääseen 2001 oli tiivistä kehitystyön aikaa ja prototyypin kehitysversio eteni. Projekti tutki Zigzag-idean

---

[55] Psykologian ja viestinnän tutkijat olivat tehneet hankkeeseen esiselvityksen. Siitonen (toim.): Mitä hyötyä tieto-ja viestintätekniikasta on perheelle ? : Perheportaali-projekti 1.vaiheen esitutkimus.

[56] Kyseessä oli Nelsonin käyttämä versio Ross Perotin, Electronic Data Systems Corporationin perustajan, yhtiön tunnuslause 1960-luvulta löytää tehokkaimmat ja luotettavimmat työntekijät ”*Eagles don't flock; you find them one at a time*”. Nelson (Literary machines 1987, sivu 1/22) kertoi Xanadu projekteistaan ja niiden työntekijöistä muokaten Perotin moton muotoon “ *Eagles don’t flock, you have to find them one by one*.”

[57] www.gzigzag.org, joka ei ole enää nykyään olemassa.

käyttömahdollisuuksia ja sen koodausta toimivaksi kokonaisuudeksi. Kehitysversio eteni vuoden aikana ensimmäisestä versiosta 0.6.1 versioon:

TAULUKKO 2: GZIGZAGIN OHJELMISTOVERSIOIDEN JULKAISUPÄIVÄT

Versio	Julkaisupäivä Internetissä
0.1 pre 0	14.9.2000
0.1 pre 1	21.9.2000
0.1 pre 2	23.9.2000
0.1	26.9.2000
0.1.1	27.9.2000
0.1.2	28.9.2000
0.1.3	9.10.2000
0.1.4	24.10.2000
0.1.5	1.11.2000
0.3.0	20.11.2000
0.4.0	10.12.2000
0.5.0	21.1.2001
0.5.1	24.1.2001
0.6.0	8.4.2001
0.6.1	23.8.2001

Keväällä 2001 ryhmä suunnitteli ohjelmistokoodinsa uudelleen kirjoittamista, jonka jälkeen julkaistaisiin 0.8 versio. Kesällä 2001 osa ryhmän jäsenistä osallistui Tanskassa Hypertext 01 konferenssin Zigzag-työryhmään, jossa pidettiin seitsemän esitelmää[58]. Esitelmöijiä olivat Nelson, hänen läheisimmät yhteistyökumppaninsa (Marlene Mallicoat, Andrew Pam ja Ed Harter), englantilaisen kehittäjätiimin jäsen Les Carr sekä Tuomas Lukka ja hänen tutkimusryhmänsä jäseniä. Zigzag-tietomallia kehittäviä projekteja oli vuonna 2001 toteutuksen alla Englannissa, Australiassa, Japanissa ja Suomessa.

Ryhmän ohjelmistoprojektin seuraaminen[59] alkoi toukokuussa 2001 pääkehittäjä Lukan haastattelulla ja ryhmän kokouksen havainnoinnilla. Haastattelussa Lukka kertoi projektin lähtökohdista. Hän kertoi lyhyesti ryhmän ensimmäisistä jäsenistä ja projektin tavoitteista kehittää avoimena ohjelmistoprojektina uudenlainen graafinen käyttöliittymä. Lukan vallankumouksellinen idea oli uudistaa tietotyöläisen käyttöliittymä myös visuaalisesti:

> Niin sitten mä törmäsin tuohon Zigzag-systeemiin ja tajusin että tuossa on, tuossa on niin kuin hirvittävää potentiaalia. Eli siis se että muutetaan koko tietomalli, mikä on kaiken pohjalla. Ei olekaan tiedosto, tiedostojärjestelmät ja hakemistot ja niin edespäin. Vaan onkin nämä solut ja dimensiot ja se on moniulotteinen järjestelmä.
> (Tuomas Lukan haastattelu 28.5.2001)

[58] Työryhmän sivu Internetissä, http://www.sigweb.org/conferences/ht-conferences-archive/ht01/workshops.html#w4, tarkistettu 3.5.2006.
[59] Havainnoijana ja haastattelijana Jyväskylän yliopiston Agorassa tutkija Jussi Silvonen 28.5.2001

Tavoitteekseen projektille Lukka asetti tehdä läpimurto tietokoneen käytettävyyden parantamiseksi. Tietokoneitten käyttöjärjestelmien edistys oli Lukan mielestä ollut vähäistä:

> Edistystä ei vielä ole oikeastaan tehty. Tai se edistys on menossa vähän vastaavaan kun tämä todistus siitä, että jänis ei saa koskaan kilpikonnaa kiinni. Edistys on tavallaan pienempiä, pienempiä askelia sitä tiettyä pistettä kohti mutta ei koskaan saavuta sitä yhtä pistettä, sitä todellista läpimurtoa.
> (Tuomas Lukan haastattelu 28.5.2001)

Haastattelussa Lukka kertoi, että ryhmän ja Nelsonin yhteistyössä oli vaikeuksia. Ensimmäiset ongelmat yhteistyössä olivat ilmenneet ja siksi oli epäselvää milloin Lukka ja Nelson tapaisivat seuraavan kerran:

> Nyt saa nähdä kuinka kauan menee seuraavaan, koska meillä on, tällä hetkellä hän on sanonut että hänellä ei ole aikaa tähän projektiin. Hän on tuskastunut siihen, että tässä ei ole saatu hänen mielestään tarpeeksi nopeasti jotain sellaista aikaan mitä hän haluais, mutta katsotaan. Elikkä hän ei itse ohjelmoi niin hän ei hahmota mikä on vaikeaa ja mikä on helppoa. (Tuomas Lukan haastattelu 28.5.2001)

Ryhmän rahoitustilanne vuonna 2001 oli hyvä. Projektin tärkeimmät rahoittajat olivat Sonera ja SysOpen. Lukka totesi Digitoday-verkkojulkaisun haastattelussa[60], että ryhmän rahoitustilanne vuodelle 2002 näytti myös lupaavalta:

> Ulkopuolinen rahoituksemme kahteen kehitysprojektiin tälle vuodelle on 400 000 markkaa, ja näyttää siltä että työ jatkuu ensi vuonna. Henkilöstövahvuus saattaa kuitenkin vaihdella.

16. joulukuuta vuonna 2001 ohjelmistoryhmän viisi Lukan alaisuudessa toimivaa ohjelmoijaa haastateltiin Agorassa[61]. Ryhmän jäsenille projektin avoimuus oli tärkeää projektin alusta lähtien. Nelsonin projekteja ei oltu aikaisemmin toteutettu avoimen lähdekoodin projekteina. Tuukka kertoi ryhmän olevan ensimmäinen, joka toteutti Nelson ajatuksia vapaassa projektissa ja kuinka Stallmanin käsitys ohjelmistojen vapaudesta oli hänelle tärkeä:

> Eihän Nelsonin projektikaan oo ollu niinku vapaata ohjelmistoa, ennen kuin me alettiin tekemään tätä. Ilmeisesti, no niin, sen verran uus juttu kuitenkin. Mutta sillä tavalla että jotenkin tuntuu että me tehdään sitä tän perinteen pohjalta, mikä on kuiteski jo muodostunut. Niin että me jollain tavalla tehdään jotain, mistä olis enemmän hyötyä, kuin jos me vaan omassa kopissa pakerrettais prototyyppiä. Kuitenkin koko ajan on niinku ajatuksena, että kaikki saa nähdä tän ja tästä pitäis olla kaikille hyötyä… Ja tosiaan meillä vissiin kaikilla on aika silleen henkilökohtainen näkemys myöskin. Se on, että ohjelmistojen vapaus on tärkeätä, samalla tavalla mitä Stallman puhuu. (Tuukan haastattelu 16.12.2001)

---

[60] Reijo Holopainen : Sonera ja SysOpen rahoittavat radikaalia Zigzag käyttöjärjestelmää, uutinen Digitodayssa 9.4.2001, verkossa http://www.digitoday.fi/showPage.php?page_id=9&news_id=5382, tarkistettu 3.4.2006.

[61] Haastattelijoina Agorassa tutkija Jussi Silvonen ja Juha Leminen.

Toisaalta ryhmän ohjelmoijista Vesa korosti ymmärtävänsä Torvaldsin näkökannan ohjelmistojen kehittämiseen. Samalla Tuukka myös kommentoi kysymykseeni, että Open Source toimintatavan uskottiin tuovan projektille tulevaisuudessa enemmän kehittäjiä.:

> Mä oon oikeastaan henkilökohtaisesti omaksunut Torvaldsin tän näkökannan tähän vapaaseen softaan, joka tuntuu olevan semmonen vähän enemmän hällä väliä kun Stallmanin käsitykset. Mä en oo niin fanaattinen sen suhteen kuin jotkut saattaa olla, ainakaan enää. (Vesan haastattelu 16.12.2001)
>
> Haastattelija: Kuinka konkreettista se periaatteessa se Open Source- juttu on. Kuinka paljon siitä on hyötyä, onko sitä vielä vai se on enemmän tulevaisuudessa?
>
> Tuukka:(-) niinku ne perinteet että tehdään samalla tavalla kuin kaikki muutkin on tehnyt, että ne on lähteny sellasesta pienestä. Ja sit on niinku olemassa sellanen, että me tiedetään että on semmonen realistinen mahdollisuus. Että jos me saadaan jotain järkevää aikaan niin silloin saadaan myös niinkun muita tekijöitä, koska on niinku helppo ymmärtää, että pienellä porukalla me ei saada sillä tavalla hirveästi aikaan. Mutta silti tuntuu järkevältä tehdä sitä.

Luvun lopuksi käsittelen yhteenvedon omaisesti luvun 5 tapahtumia. Ryhmän ohjelmoijat Lukan johdolla tuntuivat olevan lumoutuneita Nelsonista ja hänen ideoittensa sovellettavuudesta ohjelmistotuotteeksi. Sennettin mukaan (2006, 25-27) Silicon Valleyssa monet nuoret opiskelijat lähtivät tavoittelemaan mainetta ja kunniaa 1990-luvun alkupuolella Bill Gatesin jalanjäljissä. Suomalaiselle opiskelijalle Gzigzagin houkutukset olivat myös hyvät: saada pientä palkkaa, hyötyä ohjelmointiopiskeluunsa ja oppia Lukalta sekä Nelsonilta. Lukan ryhmän ohjelmoijan mukaan ryhmän toteuttamat innovaatiot Gzigzag projektissa olivat 2/3 Lukan käsialaa ja Nelsonin panos oli 1/3 projektin taustalla. Projektissa vuosina 2000–2001 toiminut Kimmo kertoi, että Nelson esiintymistapa luennoilla veti kiinnostuneita ohjelmoijia puoleensa koska ”*että se oli niinku kuitenki enemmän semmosta science fictionia siis tämä Tedin tapa ilmasta asioita*.” Projektin alkuvaiheissa Lukka kehitti Gzigzag-projektin pääohjelmoijana Nelsonin ideoita teknologisesti pidemmälle. Lukan haave oli kehittää tietotyöläiselle jokapäiväisiin tiedonhallinnan ongelmiin työkalu, johon tietokoneen käyttäjä yhdistäisi kätevästi itselleen haluamansa käyttötavat omien tietojensa hallintaan. Lukka halusi parantaa World Wide Webin puutteita, joita olivat mm. hyperlinkkien hauraus, tiedon luokiteltavuus ja tiedostojen linkitettävyys. Xanadun perusideoista Lukka kehitti oman utopistisen visionsa luoda vapaassa ohjelmistoprojektissa teknologinen sovellus, jonka käyttöarvollinen lupaus oli kehittää uudenlainen tietokoneen käyttötapa hallitsemaan tietokoneen käyttäjän informaatiovirtaa ja tiedon varastointia. Nelsonin ja Lukan

yhteistyön alkua voidaan tulkita lahjanvaihtona, jonka perustalta syntyi suullinen sopimus vapaasta kehitysprojektista.

6. VAPAAN OHJELMISTOLIIKKEEN PERUSTAJA KOMMENTOIMASSA GZIGZAGIN OHJELMISTOVERSIOTA

Heinäkuun 14. päivän aamuna vuonna 2001 järjestetyssä tilaisuudessa Richard Stallman arvioi Gzigzag-ohjelmistoversiota 0.63 Jyväskylän yliopiston Agorassa järjestetyssä esittelytilaisuudessa. Gzigzag ohjelmiston versio 0.63 oli tilanteessa, jossa suunniteltiin lähdekoodin uudelleen kirjoittamista 0.8 version aikaansaamiseksi lähitulevaisuudessa. Tilaisuuteen osallistui Stallman ja kuusi Gzigzag ryhmän jäsentä[62]. Osallistuin tilaisuuteen havainnoijana ääninauhoittamalla 75 minuutin keskustelun. Tilaisuuden lähtökohtana oli Gzigzag-ohjelmistoversiosta valkokangaalle heijastettu esitys, jolla pääkehittäjä Lukka esitteli visuaalisesti ohjelmiston toiminnallisuutta[63]. Aluksi Lukan esitellessä ohjelmistoideaa esitti Stallman tarkentavia kysymyksiä ohjelmistoideaan. Näistä kysymyksistä edettiin Gzigzag-ohjelmistoprojektia laajemmin koskeviin kysymyksenasetteluihin. Analysoimani keskustelutilanne oli luonteeltaan itsestään kehittyvä. Keskustelu eteni dialogisesti ilman ennakkosuunnitelmaa. Stallman roolissaan esiintyi asiantuntijana. Free Software-liikkeen perustajana hän edusti kommenteillaan arvoja, jotka hän oli luonut 30 vuoden uransa aikana ohjelmistokehitystyössä yliopistomaailmassa, yrityksissä ja Free Software-liikkeen johtohahmona. Dialogissa arvioitiin ohjelmistoideoiden vahvuuksia ja heikkouksia, uhkia ja mahdollisuuksia. Gzigzag-ryhmä sai ohjelmistoideoittensa kehitettävyyteen arvokasta palautetta. Ohjelmistoprojektin menestys hyödyllisenä tuotteena on riippuvainen käyttäjän ohjelmistolle asettamista vaatimuksista. Yhteistyö tulevien käyttäjien ja teknologian

[62] Juha Leminen ja Jussi Silvonen osallistuivat tilaisuuteen Stallmanin ja kuuden Gzigzag - ryhmän jäsenen lisäksi.

[63] Grönfors (1982, 88) kuvaa kenttätyön alkuvaiheen tyypillistä vaihetta, jolle ominaista on havainnointi ilman osallistumista. Esittelytilaisuudessa havainnoin ryhmän toimintaa ensimmäisen kerran tutkielmani aikana, joten lähtökohdat tutkimuskohteen hahmottamiseen perustuivat tämän demotilaisuuden informatiiviseen erittelyyn jälkikäteen kenttämuistiinpanojen ja tilaisuuden nauhoituksesta tehtyjen muistiinpanojen perusteella. Tilaisuuden keskustelun englannin kielisestä litteraatiosta on tehty suomen kielisiä käännöksiä tämän analyysin empiiriseksi esimerkkiaineistoksi.

muiden kehittäjien kanssa on välttämätöntä, jotta palaute tuotteen kehittymisestä hyödynnetään tuotekehityksessä ja suunnittelussa.

Tilaisuuden alussa Lukka esitteli Gzigzag-ohjelmiston perusideoita. Esittelyn jälkeen Stallmanin ja Lukan dialogissa etsittiin erilaisia merkityksiä Gzigzag-ohjelmiston tuotteistamiselle. Millainen ohjelmisto olisi käytännöllisiltä kehitysmahdollisuuksiltaan sen lähestyessä kehitysversiota 1.0? Käyttäjien ja kehittäjien roolit ohjelmiston hyötykäytössä ovat erilaiset. Käyttäjien taitotaso vaihtelee ohjelmointitaitoisesta tietotyöläisestä normaalin loppukäyttäjään. Ohjelmistotuotteen helppokäyttöisyys ja toimivuus olisi tärkeätä, jotta ryhmän projektiin liittyisi tulevaisuudessa vapaitten ohjelmistojen kehittäjiä.

Käyttämästäni litteraatioaineistosta on tehty sisällönanalyysi. Keskustelun englanninkielisen litteraation puheenvuorot on numeroitu. Keskustelussa käytettiin puheenvuoroja 325 kappaletta. 74 minuuttia 45 sekuntia kestäneen esittelytilaisuuden aikana puheenvuorojakauma oli seuraava:

Stallman	156
Lukka	130
Tuukka	33
Antti	3
Rauli	3

Keskustelu eteni fragmentaarisesti aiheesta toiseen Stallmanin johdatuksella. Hän lausui mielipiteitään esittelydemosta, sekä kommentoi Lukan kommentteja keskeyttämällä hänet usein. Analyysissäni olen käyttänyt pääasiassa Stallmanin arvioivia kommentteja, koska keskustelu keskittyi Lukan ja Stallmanin välille. Puheenvuoron perässä on esitetty sulkeissa alkuperäisen englanninkielisen puheenvuoron numero. Luvussa 6 on kaksi puheenvuoroa, jotka kuvaavat tilaisuuden alussa tapahtuvaa Lukan esittelyä. Luvuissa 6.1 ja 6.2 on tematisoitu otsikon alle luvun aiheeseen liittyvä keskustelu.

Lukka aloitti esittelynsä viittaamalla ohjelmiston visuaaliseen ulkoasuun:

> TL: ...ovat loistava tuotos nykyisestä tilanteestamme, missä me olemme kehittämässä systeemiämme, kuten tämä esimerkkinäkymä(-).(3)

Lukan mukaan dokumenttien identiteetti säilyisi ryhmän kehittämän pysyvän tunnisteen avulla. Tunnisteen avulla dokumenttien suhteet säilyisivät tiedonhallintajärjestelmässä.

Lukka esitti esimerkin Xanadun perusidean soveltamisesta käytäntöön Gzigzag tiedon-hallintamenetelmällä. Valmista ohjelmistoa testattaisiin Lukan mukaan arkisessa käytössä kun ohjelmistoversio 0.8 saataisiin valmiiksi:

> TL: Xanadussa on hieno lisäetu... kun työskentelee ryhmänä muuttaen jotain pitkää artikkelia ja näkee mitä muut tekee, kaikki nämä muutokset helpommin. Kun saa todellisen tekstin identiteetin missä voi verrata erilaisia versioita(71)... ja kun olemme tässä pisteessä, alamme käyttämään systeemiä itse, tehden sillä päivittäisiä asioitamme. (168)

6.1 Hyödyllinen ohjelmistotuote vai tieteellinen tutkimustyö?

Stallmanin keskeisimmät argumentit keskustelun kuluessa koskivat ohjelmistotuotekehityksen mahdollisuuksia ja riskejä Free Software projektissa. Tällainen esimerkki keskusteluaineistosta oli Stallmanin kommentti tieteellisen tutkimuksen ja ohjelmistotuotekehityksen yhdistämisen vaaroista. Lukka kertoi ensin projektia toteutettavan kahdesta erilaisesta näkökulmasta:

> TL: Periaatteessa kaksi lähtökohtaa, mahdollisesti teemme molempia, toisaalta teemme mahdolliseksi toimia niin paljon kuin mahdollista jo olemassa olevien asioitten kanssa jotka ovat jo valmiina. Mutta toisaalta haluamme tutkimusprojektina nähdä mitä,,(61)

Stallman keskeytti Lukan kommentin todeten tieteellisen tutkimuksen mahdollisesti häiritsevän käytettävän ohjelmiston kehittämistä. Hänen mielestään vaara tieteelliseen tutkimukseen keskittymisessä on tiedostettava projektin päämääriin pyrittäessä:

> RMS: Siinä on eroa, siinä on ristiriitaiset roolit tutkimuksen ja hyödyllisen ohjelmisto-paketin kehittämisen välillä(62)... Pelkään että halunne tehdä tutkimusta saattaa aiheuttaa muuten hyödyllisen ohjelmiston[valmistuvan] ei käytettäväksi. Se on harmi ja se voidaan ehkä pelastaa, mutta toivon että se [ohjelmistonkehityksenne] ei epäonnistu, mutta se on todellinen tuhoisa vaara, josta teidän täytyy olla tietoisia.(64)

Stallmanin mielestä oli olemassa riski, jos projektin tavoitteita ei aseteta tärkeysjärjestykseen. Kun ryhmä esitteli keskustelun aikana useita alkuvaiheessa olevia ideoitaan, niin Stallman kehoitti ryhmää keskittymään vain yhteen tavoitteeseen kerrallaan. Hänen mukaansa projektien paralleelinen kehittäminen olisi turmiollista ohjelmistotuotteiden valmistamiselle:

> RMS:(-)Älkää yrittäkö tehdä niin monia tutkimusprojekteja ennen kuin teette jotain hyödyllistä, koska se ei ole tapa tehdä jotain hyödyllistä, se on hirveä virhe. ...ajattelen että teidän täytyisi tehdä käyttäjille hyödyllinen työväline editoimiseen ja sitten yrittää muita ideoita ohjelmistokieleen. Jos yritätte liian monia ideoita, ette ikinä tee hyödyllistä työvälinettä ja se ei ole hyvä asia.(95)

Stallman korosti potentiaalisen loppukäyttäjän näkökulmaa Free Software ideologian pohjalta. Hänen kritiikkinsä kohdistui kehitys- ja käyttäjäympäristön kontekstualisointiin, jolla hän tarkoitti, että on mahdollista kohdata riskejä käyttöönotossa avoimessa ohjelmistotyöprojektissa. Käytettävyysnäkökulmasta Stallmanin lähtökohta oli, että tietokoneen käyttäjän käyttöympäristöön ei voi tehdä suuria muutoksia. Tällaisen lähtökohdan huomioiminen ohjelmiston suunnittelussa olisi ensiarvoisen tärkeää Gzigzagin ohjelmistoideoille:

> RMS: Jos ajattelette korvaavanne tavat. Miten ihmiset tekevät asioita, te implisiittisesti pyydätte heitä lopettamaan tekemästä mitä he haluavat. Ja te ette ymmärrä, miten negatiivinen asia se on.(116)

Stallmanin mukaan ohjelmistojen loppukäyttäjät eivät suostuisi suuriin muutoksiin. Graafisessa käyttöliittymäympäristössään. Loppukäyttäjän käyttötottumukset pitäisi huomioida tarkasti ohjelmistojen suunnitteluvaiheessa:

> RMS: Ette tunnu ymmärtävän, että jos kerrotte henkilölle että hänen täytyy käyttää näitä kaikkia yhdessä, kerrotte tälle henkilölle, että hänen täytyy kääntää koko elämänsä ylösalaisin tai hän ei voi käyttää ohjelmia. Ja useimmat ihmiset sanovat : ”*Ei kiitos, en halua kääntää elämääni ylösalaisin*”. Se on kriittinen suunnittelupäämäärä tehdä nämä välineet toimimaan yksilöllisesti. Absoluuttisesti kriittinen. Luulen ettette ymmärrä tätä.(196)

Stallman varoitti myös liian suurista suunnitelmista muuttaa asioita suhteessa nykyiseen ohjelmistonkehitysparadigmaan:

> RMS: Jos olette päättäneet ottaa ison askeleen... Vaikka tuottaisitte jotakin hyvää, niin ihmiset eivät käytä sitä, koska olette luoneet käytännölliseen esteen. Vaihtaneet kokonaan uuden tavan tehdä asioita, joka on valtava käytännöllinen este.(210)

Stallman kehoitti ryhmää ohjelmistoideoissaan yhdistämään kaksi käyttäjän tietotyössä yhdistettävää dokumenttien hallintatapaa:

> RMS: Kehotan teitä ajattelemaan kahta esitystapaa yhtä aikaa. Kun ihmiset suunnittelevat, kun ihmiset kirjoittavat. He aina miettivät molempia, miltä data näyttää ja miltä teksti tulee näyttämään. Jos alatte miettimään niitä yksittäin, mutta haluatte lisätä tulevaisuuden välineen...mitä haluaisin, niin että se näyttäisi tekstiltä, silloin kykenette suunnitteleman järjestelmän missä asiat voidaan joustavasti kätevästi liikutella.(187)

6.2 Laboratorioesineitä, ideoita norsunluutornissa vai toimiva vapaa ohjelmisto?

Stallman yritti kysyä ryhmältä miten heidän mielestään tieteellisen tutkimustyön ja ohjelmointityön välillä voisi olla toisiaan tukeva tehtävänasettelu. Stallman korosti jo

toistamiseen, että projektin olisi valmistettava vakaa ohjelmistoversiota, muuten projektin epäonnistumiselle olisi olemassa liian suuri vaara. Ideoinnista olisi siirryttävä pikaisesti tuottamaan vakaa versio seuraavassa ohjelmistonkehitysvaiheessa. Stallman kehoitti ryhmää aloittamaan ohjelmiston valmistamisen pikaisesti pääkehittäjä Lukan johdolla. Hän kohdisti kommenttinsa Lukalle henkilökohtaisesti. Stallmanin mukaan vapaan ohjelmistotyön ideat olivat monien ohjelmistokehittäjien kiinnostuksen kohteena, ja siksi ideat olisi tuotettava itse hyödylliseksi vapaasti lisensoitavaksi ohjelmistoksi mahdollisimman pian:

> RMS: ...jos mietitte ideoita abstraktisti ettekä toimi primitiivisellä tasolla, teette puolet työstä. Mutta toinen puoli työstä on saada se todella toimimaan, ja jos ette tee sitä, teillä ei ole mitään. Se on erilainen työ, mutta yhtä tärkeä opiskelijoiden[64] oppia, et halua heidän oppivan vain käyttämään mielikuvitustaan vaan todella saamaan asiat toimimaan ja hyväksi käyttää.(166)...jos teillä on idea, te voitte vain tutkia ideaa ja sitten joku toinen kirjoittaa sen, kuka tämä joku toinen on, onko se patentoiva ohjelmistokehittäjä? Teidän täytyy tuottaa hyödyllisiä ohjelmia tai kuka sen tekee? (166)

Stallman tiivisti keskeisen ohjelmistotuotantoa ja ohjelmistoidean valmistamisprosesseja koskevan peruskysymyksen idean tuotteistamiseksi hyötykäyttöön. Idean tuotteistaminen vaatisi paljon raakaa työtä:

> RMS: Mutta kun alatte tekemään niitä, ne ovat paljon työtä. Ja miksi niin pitkään kuin on helppo perustaa monia eri projekteja, mutta ette saa niitä koskaan tehtyä. Ne ovat vain leluja, ne ovat vain laboratorioesineitä, eivät käytettäviä hyödykkeitä, jotka todella vaikuttaisivat yhteisöömme. Se on paljon suurempi työ kuin mietitte ja jos ette tee sitä, tulette aina olemaan norsunluutornissa ja työtänne ei tulla koskaan käyttämään, joten tehkää tämä yksi osa ja ehkä editori ekana osana. (272)

Lukka esitti tähän vastakysymykseksi, että millaisia erilaisia projekteja Stallmanin mielestä on Gzigzag-hankkeessa. Stallman vastasi:

> RMS: Kolme projektia: ohjelmointikieli, tapa editoida muistiinpanoja ja tekstiä sekä mediapalvelin, nämä ovat kolme projektia. (229)

Näistä Stallmanin mukaan yksi olisi valmistettava pikaisesti vapaata ohjelmistoyhteisöä hyödyttäväksi ohjelmistotuotteeksi. Hän arvioikin ohjelmistodemoa vaihtoarvon eli toimivan hyödykkeen markkina-arvon perusteella.

> RMS: Jos haluatte tuotteistaa jotain joka on merkittävä osa tulevaisuuden vapaata maailmaa, teidän on parasta keskittyä tekemään yksi asia, joka on hyvin kehitetty, vakaa ja toimii, eikä pelata vain ideoilla. (206)

Tilaisuuden lopussa Stallman kysyi auttaisiko ryhmä Gnu-projektia; ehkä he voisivat yhdessä kehittää ohjelmistoideaa tai Emacs-tekstieditoria:

---

[64] Ryhmän muut jäsenet olivat statukseltaan opiskelijoita, joihin Stallman viittaa Lukalle kommentissaan.

> RMS:(-) Emacsista on kaksi versiota. Gnun ja toinen. Auttakaa Gnu-projektia kehittämään versiotamme Emacsista(359)…Joka tapauksessa auttakaa Gnu-projektia pitämään editoriamme tärkeimpänä tukemalla meitä.(362)

Tilaisuus kuitenkin loppui ja yhteistyökysymys jäi ilman vastausta.

Mitä mieltä Gzigzag-ryhmä oli Stallmanin kommenteista? Ryhmän Irkissä keskusteltiin saman päivän iltana aamupäivän Stallman-tapaamisesta. Ryhmän ohjelmoija Rauli kertoi Irkissä tilaisuudesta ryhmän saksalaiselle etäjäsenelle Benjalle, joka ei ollut osallistunut demotilaisuuteen. Pääkehittäjä Lukka osallistui myös keskusteluun hetkeä myöhemmin. Rauli totesi Stallmanin toistettavasti kertoneen perustotuuksia ohjelmistoprojektin kehityksestä. Raulin mukaan Stallmanin viestittämä perustotuus oli:

> RR: jos ette tee [ohjelmistoa] kilpailukykyiseksi olemassa olevien ohjelmistojen suhteen, kukaan ei ala käyttämään sitä.

Raulin mukaan Stallman painotti että tulisi saada jotain käytettävää valmiiksi mieluummin kuin vain tehdä tutkimusta. Pääkehittäjä Lukan mielestä Stallman antoi erilaisia hyviä neuvoja. Stallmanin esittämät erilaiset perspektiivit olivat Lukan mukaan seuraavanlaisia:

> TL: Kolme asiaa. Päämäärien asettaminen, vapauden tärkeys, saatava valmiiksi jotain, joka todella auttaa ihmisiä tekemään asioita [ohjelmistolla]. Meidän täytyy valmistaa jotain, jota ihmiset päättävät käyttää jotain tarkoitusta varten, eikä kääntää kokonaan käyttötarkoituksia.

Samassa Irkki-keskustelussa Benja esitti teoreettisen idean tiedostodokumenttien tunnistamiseksi. Tähän pääkehittäjä Lukka kommentoi viestittäen Stallmanin kommentin ideoinnista tulleen ymmärretyksi: *Tuo oli norsunluutorni näkökulma ;)*

Luvun lopuksi käsittelen yhteenvedon omaisesti luvun 6 tapahtumia. Stallman antoi ryhmälle hyviä neuvoja koskien ryhmän avoimia toimintatapoja ja kuinka heidän pitäisi priorisoida projektinsa tavoitteita. Ryhmän jäsenten keskustelu samana iltana Irkissä osoitti, että Stallmanin neuvoja oli kuunneltu. Projektin tulevaisuus näyttäisi miten Stallmanin neuvoja hyödynnettäisiin käytännössä. Stallmanin neuvot liittyivät myös siihen, että projektia tulisi hallita paremmin. Monet kommentit oli osoitettu suoraan pääkehittäjä Lukalle. Stallmanin mielestä ohjelmistolla oli mahdollisuuksia tulla toimivaksi hyödykkeeksi vapaan kehittäjäyhteisön toimesta. Hän kritisoi erityisesti tutkimustyön ja toimivan ohjelmistohyödykkeen tuottamisen ristiriitaista suhdetta. Stallmanin mielestä toimiva ohjelmistotuote saataisiin aikaan, jos ryhmän jäsenet

pystyisivät priorisoimaan kehitystavoitteitaan. Vaikka hypertekstuaaliset ideat saivat kritiikkiä, oli ryhmän jäsenten luottamus niiden kehittämiseen suuri vielä tilaisuuden jälkeenkin.

7. GZZ-RYHMÄN OHJELMISTOTYÖN ARKEA HAVAINNOIMASSA TOUKOKUUSSA 2002

Gzz-projektin havainnointi suoritettiin 20–21.5.2002 seuraamalla ryhmän työtä Jyväskylän yliopiston Agorassa. Havainnoinnin lähtökohtana oli seurata ohjelmistotyön arkea. Ohjelmistotyön mikrososiologisen kuvauksen kohteena olivat ohjelmoijat ja heidän työtapansa tiimityössä. Tässä luvussa kuvaan millainen on Jyväskylän yliopiston Agorassa ohjelmoijien työpäivä, kun ohjelmiston lähdekoodia kehitetään yhteistyönä. Kenttähavainnoinnin näkökulmana oli pyrkimys havainnoida miten ohjelmoivat työskentelevät sosiaalisessa vuorovaikutuksessa ja miten koodin tekemisen tavat Internetvälitteisesti yhdistyvät ohjelmistotyössä. Ohjelmistotyön edellytys on kytkeytyä tietoverkkoon ja viestintävälineisiin, joiden avulla lähdekoodia luodaan. Ohjelmistotyön etnografinen analyysikuvaukseni etenee työn välineistä niiden käyttötapoihin, vuorovaikutukseen verkossa ja ohjelmistokoodin kehittämiseen. Luku jatkuu esimerkin avulla kuvauksella Gzz-ryhmän työstä maanantai-iltapäivänä 20.5.2002. Luvussa kuvataan myös miten ohjelmoijien kasvokkain neuvottelut yhdistyvät verkon kautta tapahtuvaan lähdekoodin valmistamiseen.

Havainnoinnin kohteena kahden työpäivän aikana olivat Gzz-ryhmän jäsenet: Tuukka, Vesa, Janne ja Tero. Gzz-ryhmän työtilat sijaitsivat Jyväskylän yliopiston Agoran rakennuksen 4. kerroksessa ohjelmistotekniikan yksikön kolmessa vierekkäisessä tilassa. Ryhmän tilat olivat ohjelmistotekniikan huone, jossa sijaitsivat Tuukan ja Jannen päätteet. Huoneesta avautuvassa pienessä avoimessa tilassa työskentelivät Vesa ja Tero. Viereinen huone oli ryhmän pääkehittäjän Tuomas Lukan huone. Agorassa työpisteissään kahden päivän havainnoinnin aikana työskentelivät: Tuukka 11 tuntia, Vesa 12,5 tuntia, Janne 7 tuntia ja Tero 6 tuntia. Frekvenssi ohjelmoijien työjaksojen kestoista yhtämittaisena työskentelynä päätteen ääressä olivat 5 minuutista–5 tuntiin.

Kenttähavainnointi suoritettiin erilaisilla havainnointivälineillä. Videokameralla, MiniDisc-nauhurilla ja havainnointipäiväkirjaan tehdyillä merkinnöillä tallennettiin kahden päivän tapahtumia. Ryhmän työn etnografiaa kuvaavat havainnointipäiväkirjan merkinnät, joihin kenttähavainnoinnin analyysi perustuu. Kenttähavainnoinnissa kirjattiin kahdenlaisia faktoja: deskriptiivisiä ja refleksiivisiä faktoja. (Creswell 1998, 128-129). Deskriptiivisillä faktoilla kuvataan työtä. Työn sosiaaliset järjestykset ilmenevät työtapahtumina kronologisessa aikajärjestyksessä työpaikalla. Nämä elementit muodostavat kokonaiskuvan siitä, millaisia työn jaksoja ohjelmistotyössä esiintyy. Refleksiivisellä havainnoinnilla kuvataan tietoa, jota havainnoitsija on luonut osallistuvalla havainnoinnilla kyselemällä ohjelmoijilta heidän työstään. Refleksiivinen havainnointi sisältää havainnoijan omat tulkinnat, jotka kuvaavat huomioita kenttähavainnoinnin aikana Agorassa.

Tietotyön aikakausi on muuttanut etnografisen analyysin kohdetta ja luonnetta. Virtuaaliyhteisöjen tutkimus sosiologiassa on metodien ja erilaisten teorioiden kehittelyn alkuvaiheissa. Knorr-Cetina ja Bruegger (2002) ovat tutkineet osallistuvan havainnoinin, haastattelujen ja pankkijärjestelmän elektronisen dokumentaation perusteella millainen globaali virtuaaliyhteisö muodostuu pankkimaailmassa. Ajallinen koordinaatio käsitteen avulla he esittävät millaisista päällekkäisistä, jatkuvista ja epäjatkuvista tiedon vaihtamisen muodoista globaali virtuaalinen pankkiyhteisö muodostuu. Tässä luvussa pyrin samalla tavoin kuvaamaan millaisista päällekkäisistä, jatkuvista ja epäjatkuvista tiedon vaihtamisen muodoista Gzz-ryhmän työ koostuu. Yhteisön koolla on merkitystä tutkimusasetelmien asettamisessa ja metodisten analyysimallien valitsemisessa. Toimintansa laajentuessa paikalliset virtuaaliyhteisöt kehittyvät globaaleiksi tiedon vaihdannan verkostoiksi. Paikallisen virtuaaliyhteisön lähtökohta on sen perustuminen sekä kasvokkain että elektronisesti välittyneisiin sosiaalisiin suhteisiin. Tutkimus-asetelmana virtuaaliryhmän tutkiminen ja sen mahdollinen kehittyminen vaatii oman näkökulmansa. Sosiaalisen pääoman käsitteeseen liittyvä ymmärrys vahvojen ja heikkojen sidosten merkityksestä muuttuu yhteisöissä niiden toimintatapojen muuttuessa ja hierarkkisuuden kasvaessa. Gzz-ryhmä toimi pienenä virtuaalisena työyhteisönä, jonka toimintaa määrittelivät ryhmän vahva solidaarisuus ja tiedon avoimuus.

Gzz-ryhmän ohjelmoijat työskentelivät deadlinen kynnyksellä keskittyneesti omilla päätteillään. 90 % havainnointiajasta ohjelmoija työskenteli päätteen ääressä sormet

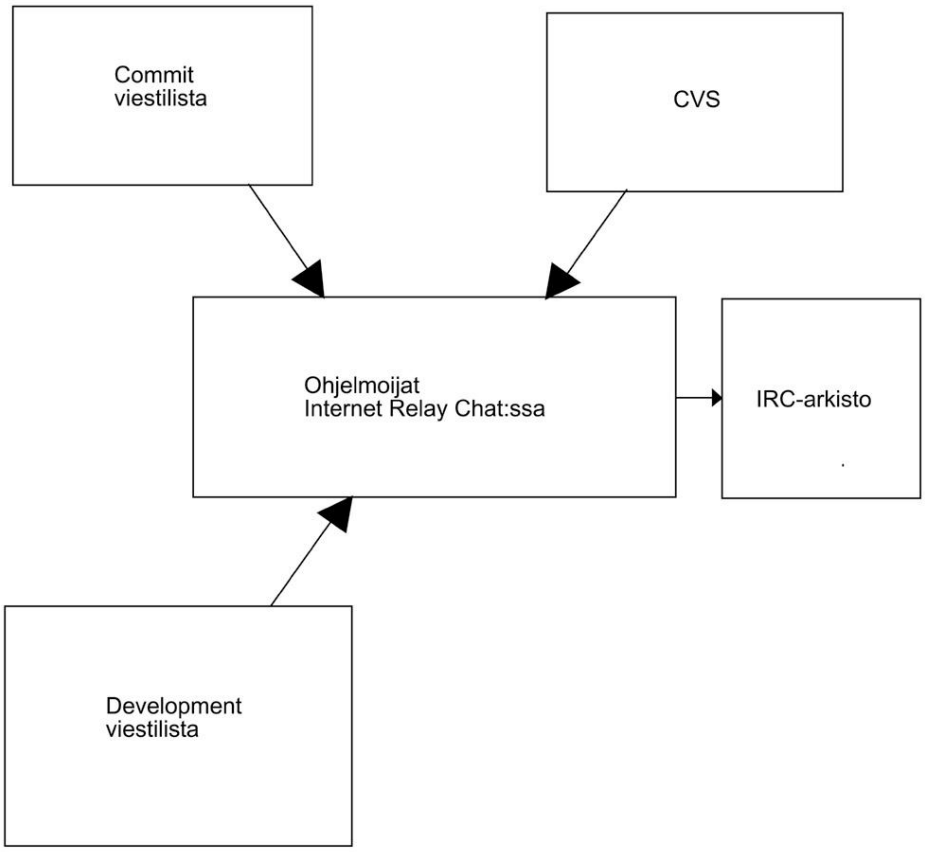
näppäimistöllä ja hiirellä. Työn keskeyttivät lyhyet neuvottelunomaiset keskustelut ja puhelut sekä arkiset työpäivän rutiinit, kuten esimerkiksi ruokatauko, kahvin juominen päätteen ääressä ja lyhyet keskustelut. Ohjelmointityön innovaatioprosessissa tiedon tuottaminen suoritetaan tietoverkossa kirjoitettuna tekstinä. Ryhmän työn havainnoinnin merkittävin kohde oli näyttöpääte, josta välittyvät havainnoijalle ohjelmistoprosessin sosiaalisen vuorovaikutuksen tavat. Tietotyön etnografisessa havainnoinnissa elektronisilla dokumenteilla on merkittävä rooli. Ryhmä dokumentoi tutkimustyönsä Irkkiviesti-liikenteen ja kehitystyöhön liittyvän sähköpostiliikenteen. Etnografisessa analyysissa dokumentit nousevat ohjelmistotyöprosessin tulkinnan kohteeksi ja apuvälineiksi.

7.1 Gzz-ryhmän työvälineet ohjelmistotyössä

Tietotyössä ohjelmoijalla on erilaisia työtehtäviä. Tietotyön itsenäiset prosessit liittyvät ohjelmointiin, jossa lähdekoodin kirjoittaminen tapahtuu ihmisen ja koneen välisenä vuorovaikutuksessa itsenäisenä prosessina. Ryhmän työtavoissa tietotyön muodot yhdistyvät ihmisten jakamina kokemuksina tietoverkon välityksellä erilaisten välineiden avulla tapahtuvana vuorovaikutuksena. Ryhmätyössä on myös kokouksia ja keskusteluja, jotka käydään kasvokkain vuorovaikutuksessa ilman teknologiaa. Gzz-ryhmän työ tapahtui pääasiassa Internet-välitteisesti erilaisilla viestinnän välineillä. Gzz-ryhmässä työskenteli yhdeksän kehittäjää, jotka kaikki omasivat oikeuden tehdä muutoksia lähdekoodiin. Gzz-ryhmän avoimen toimintamallin keskeiset työvälineet mahdollistivat erilaisia Internet-välitteisiä työkäytäntöjä. Luvussa 7.1 esittelen ryhmän työvälineitä ja niiden käyttöä. Kaavio 5 kuvaa ryhmän toiminnan kommunikaatio-verkostoa ja sen linkittäytymistä erilaisten automaattisten toimijoiden välittäminä. Ryhmän tärkein lähdekoodin tekemisen ja viestimisen väline on CVS-ohjelmisto[65]. Concurrent Version Systems (CVS) on suosittu versionhallintajärjestelmä ohjelmisto-projekteissa. Gzz-ryhmässä yhdeksän kehittäjää ylläpiti avoimesti ohjelmistoprojektiin tehtäviä muutoksia. Gzz-ryhmän projektin dokumenttien varasto oli saatavilla Internetissä Savannahin ylläpitämällä palvelimella muille kehittäjille. Tietovarastossa olivat kaikki ryhmän erilaiset dokumentit ja sinne lisättiin dokumentteja päivittäin. Tietovarastossa olevia dokumentteja voidaan muuttaa, voidaan luoda uusia dokumentteja tai dokumentteja voidaan ottaa pois käytöstä, esim. lähdekoodi. Nämä kolme toimintoa

[65] Jatkossa käytän Cvs:stä termiä tietovarasto.

on erotettu erilaisella väripohjalla ohjelman toimesta. Ilmoitus tietovarastoon tehtävistä muutoksista tapahtuu tietovaraston Commit-keskustelulistalla. Commit-viesti on tiedostojen muuttamiseen kytketty sähköpostiviesti. Samalla hetkellä kun viesti lähetettiin ryhmän listalle viestissä ilmoitetut muutokset tapahtuivat lähdekoodissa.



Kaavio 5: Gzz-ryhmän ohjelmistotyövälineitten ja kommunikaation verkosto

Ryhmän tietovarasto sisälsi 1978 tiedostoa[66]. Kesäkuun 19. päivän jälkeen 1039 tiedostoa oli muutettu. Ohjelman kehitysversio oli 0.8alpha1[67]. Dokumentti 1 on esimerkki ryhmän CVS-commit viestistä 20. toukokuuta vuonna 2002. Viestissä lähettäjä Lukka ilmoitti, miten hän oli muuttanut lähdekoodia. Viestissä ilmoitettiin mihin tiedostoon muutos oli tehty, mitä se koski ja mitä koodille oli tehty. Clean up tarkoitti, että tiedostoa oli ”puhdistettu” vioista. Näiden tietojen avulla ja viestin ajankohdan huomioon ottaen ryhmän muut ohjelmoijat ottivat viestin huomioon työssään.

[66] Dokumenttien määrä 15.8.2002.
[67] Versionumero 29.8.2002.

Toinen keskeinen työtapa oli Internet Relay Chat[68], joka on reaaliaikainen viestintä-järjestelmä verkossa. Irkin kehitti suomalainen opiskelija Jarkko Oikarinen vuonna 1988. Reaaliaikainen viestiminen verkossa tapahtuu kirjautumalla Irkki- kanavalle. Tyypillisin Irkki-viestin muoto on yksirivinen viesti, joka välittyy muille kun viestin kirjoittaja lähettää viestin listalle. Irkki-lista näytöllä kertoo keskusteluun osallistuvien nimimerkit.

From: Tuomas J. Lukka <address@hidde
Subj: [Gzz-commits] (no subject)
Date: Mon, 20 May 2002 05:50:38 -040

Update of /cvsroot/gzz/gzz/gzz
In directory subversions:/tmp/cvs-serv17918/gzz
Modified Files:
    Cell.java
Log Message:
Cleanup
gzz/gzz Cell.java,1.10,1.11
Mon May 20 05:50:37 EDT 2002

Dokumentti 1: Tuomas Lukan viesti Gzz-commit listalla 20.toukokuuta 2002.

Irkki-kanavaa käytetään henkilökohtaiseen kahden toimijan väliseen keskusteluun, mutta kanavalla keskustelevat myös yhteisöt reaaliaikaisesti. (Herring 2002, 120-123). Seuraava ryhmän ohjelmoijan kommentti kuvaa ryhmän Irkin käyttöä ohjelmistotyössä:

> Saatan roikkua Irkissä 24 tuntia vuorokaudessa, mutta en välttämättä lue koko ajan tai osallistu Irkkiin. Joskus luen viestit jälkikäteen, joskus en. (Tuukan kommentti 15.8.2002)

Ohjelmoija työskentelee viestimällä Irkki-kanavan dialogissa. Ohjelmoija seuraa Irkki-kanavaa ohjelmoinnin ohessa. Muita Irkki-kanavia käytetään keskinäiseen viestimiseen kahden ihmisen välillä pienemmän työtehtävän suorittamiseen. Esimerkiksi havainnoinnin aikana Tuukka ja Vesa käyttivät työssään lisäksi kahdenkeskistä Irkki-viestintää. Dokumentti 2 on esimerkki Gzz-ryhmän Irkin käytöstä. Ryhmän viestiminen tapahtui reaaliaikaisena yhdeksän kehittäjän ja mahdollisten satunnaisten vierailijoiden osallistuessa keskusteluun ryhmän Irkki-kanavalla. Ryhmän yksi jäsen oli saksalainen kehittäjä Benja. Niinpä keskustelua Irkki-kanavalla käytiin suurimmaksi osaksi englannin ja suomen kielellä, mutta joskus harvoin kanavalla keskusteltiin myös saksaksi. Irkki-keskustelussa käytetään apuna lyhenteitä ja hymiöitä. Esim. jos kanavalla ei ollut viestejä, ohjelmoija lähettää viestin ”*ayh*”, tarkoittaen ”*Are you here*?” kutsuen muita keskusteluun. Jokaisella ryhmän jäsenellä on Irkki-kanavalla oma

[68] Jatkossa käytän Internet Relay Chatista siitä suomen kielessä käytettyä erisnimeä Irkki.

nimilyhenteensä. Dokumentissa 2 alapuolella vegai on Vesa, stugge on Tuukka ja tuomasl on Lukka. Dokumentti kuvastaa viestiliikenteen etenemistä 20.5.2002 kello 16–17 välillä. Viestit etenevät kronologisessa aikajärjestyksessä keskusteluna vastaavasti ohjelmoijan näytöllä:

2002-05-20 13:47:45 Stugge: mistä noi ulottuvuudet ennen on löytynyt Clientissä? PlaneView sai ne ainakin näkymärakenteesta
2002-05-20 13:48:32 Stugge: niin se saa ne muka edelleen
2002-05-20 13:48:41 Stugge: toi on ihan rikki ton puolesta
2002-05-20 13:49:08 tuomasl: täällä taas
2002-05-20 13:49:14 Stugge: no nii
2002-05-20 13:49:19 tuomasl: (Stugge) joo, mutta näkymärakennetta ei ole
2002-05-20 13:49:26 Stugge: niinpä =)
2002-05-20 13:49:34 tuomasl: (Stugge) jos oli jotakin outoa, niin olisitte voineet soittaa ;(
2002-05-20 13:50:04Z Stugge: no enpä olisi osannut kysyä että mitä outoa
---- hetkeä myöhemmin ----
2002-05-20 13:53:56 tuomasl: itse asiassa kyllä nälkymän muuten pitäisi toimia
2002-05-20 13:53:59 tuomasl: ainoastaan toiminnat
2002-05-20 13:54:04 tuomasl: (Stugge) pistä vakiodimensiot
2002-05-20 13:54:17 Stugge: (tuomasl) ai mistä?
2002-05-20 13:54:34 tuomasl: luo uudet jos ei muuta
2002-05-20 13:54:48 tuomasl: tai aja vanha klientti ja laita printtaamaan ne dimensioitten id:t
2002-05-20 13:55:02 Stugge: no mistä Client ne on ennen saanut
2002-05-20 13:55:18 tuomasl: rakenteesta
--- hetkeä myöhemmin ---
2002-05-20 13:56:50 tuomasl: arkkitehtuurin enkapsulaatio alkaa näkyä 8-)
2002-05-20 13:56:58 tuomasl: ollaan menossa hyvään suuntaan

Dokumentti 2: Gzz-ryhmän Irkki-viestiliikennettä maanantaina klo 16–17

Sähköposti on väline, joka perustuu viestin lähettämiseen ja vastaamiseen tietoverkossa. Sähköposti ei toimi samoin reaaliaikaisena viestintävälineenä kuten Irkki. Sähköposti toimi myös työmuotona Gzz-ryhmän keskustelulistoilla. Viesti lähetetään listalle, joka on perustettu tiettyä käyttötarkoitusta varten. Listoista käytetään nimitystä keskustelulistat (discussion lists). Ensimmäistä keskustelulistaa MsgGroup ylläpiti ARPA:n henkilöstö vuonna 1975 (Herring 2002, 115). Ryhmä käytti sähköpostilistoja kehitystyössään kahdella tavalla. Ensinnäkin tekniset kehitysviestit tulivat Gzz-commit listalle viestin muodossa samaan aikaan kun muutos tuli lähdekoodiin. Toiseksi kehityskeskusteluun liittyvät viestit olivat Gzz-dev kehityslistalla. Dokumentti 3 kuvaa toukokuun 2002 viestejä, jolloin kehitystyössä lähetettiin vähemmän viestejä. Ryhmä oli aloittanut toimintansa 15. toukokuuta 2002 Savannahin kotisivun kautta:

May 26, 2002
Re: [Gzz] Global cell IDs, Tuomas Lukka, 08:59
May 25, 2002
[Gzz] Global cell IDs, B. Fallenstein, 14:33 [jatkuu seuraavalla sivulla]

May 19, 2002
Re: [Gzz] One more new file format ;), Tuomas Lukka, 13:19
Re: [Gzz] One more new file format ;), B. Fallenstein, 12:20
[Gzz] One more new file format ;), Tuomas Lukka, 08:24
May 18, 2002
[Gzz] Status discussions, Tuomas Lukka, 15:26
Re: [Gzz] Jython tests, Tuomas Lukka, 14:54
Re: [Gzz] Jython tests, B. Fallenstein, 12:00
Re: [Gzz] Jython tests, Tuomas Lukka, 10:55
Re: [Gzz] Jython tests, B. Fallenstein, 10:09
[Gzz] Jython tests, Tuomas Lukka, 07:34
May 15, 2002
[Gzz] lava/SimpleSpace, Tuomas Lukka, 08:13

Dokumentti 3: Gzz-dev listan viestien otsikot toukokuussa 2002.

Lukan viesti aiheesta Status Discussions toukokuun 18.päivä sisälsi seuraavan viestin:

From: Tuomas Lukka

Subject: [Gzz] Status discussions

Date: Sat, 18 May 2002 22:26:40 +03

User-age Mutt/1.3.25i

I'm going to be on IRC at random times in the near future, so
let's discuss schedules and plans here.
Most important things in the near future:
1. lava/Leo: get it into use with the fallback client to obtain
a SIMPLE, ROBUST system.
2. PerhePortaali: vegai: I have to deliver the software to Sonera
on tue/wed - could you please give me a good idea of what's going on
with it?
Tuomas

Dokumentti 4: Tuomas Lukan viesti 18.5.2002 Gzz-dev listalla

Dokumentissa 4 on Tuomas Lukan viestin sisältö 18.5.2002. Kehityslistan viestit olivat keskustelunomaisia mielipiteiden vaihtoa teknologisista yksityiskohdista koskien ohjelmoijan työtehtäviä. Viestissä Lukka kertoi sen hetken tärkeimpien työtehtävien sisällöistä. Työtehtävä 2 koski Perheportaalia, jonka lähdekoodin tekemistä olin havainnoimassa kaksi päivää kyseisen viestin lähettämisen jälkeen 20–21.5.2002.

Gzz muutti viestintänsä Sourceforgen kotisivustolta Savannahin kotisivustolle huhtikuussa 2002. Alla olevassa taulukossa on Gzz:n commitviestinnän määrät aikavälillä 9.4.2002–26.9.2002. Gzz-commit lista luotiin ohjelman lähdekoodin siirryttyä Savannahin palvelimelle 9.4.2002. Asko ja Matti aloittivat ryhmässä syyskuussa 2002. Sen alla olevassa taulukossa on puolestaan Gzz-dev listan viestimäärät Savannahissa. Gzz-ryhmä siirtyi Savannahissa listan käyttöön hitaasti. Loppukesästä alettiin valmistella 0.8 versiota ohjelmasta, jolloin viestien määrä listalla nousee.

Taulukko 3: Gzz-commit viestien määrät huhtikuusta 2002–syyskuuhun 2002

Gzz – commit	huhti, 9.4.	touko	kesä	heinä	elo	syys
Antti-Juhani K.	13		1			2
Asko S.						1
Benja F.	45	70		60	213	113
Janne K.	45	76	10	57	111	86
Matti K.						13
Tero M.			1			7
Tuomas L.	225	258	377	308	459	226
Tuukka H.		47	48		6	23
Vesa K.	13	11	28	2	11	10
yhteensä	341	462	465	431	817	459

Taulukko 4: Gzz-dev viestien määrät huhtikuusta 2002–syyskuuhun 2002

Gzz – dev viestit	huhti, 9.4.	touko	kesä	heinä	elo	syys
Antti-Juhani K.				7	19	3
Asko S.						6
Benja F.		4	7	34	91	76
Janne K.				6	14	14
Matti K.						15
Tero M.				5		9
Tuomas L.		8	6	49	106	108
Tuukka H.					13	1
Vesa K.				2	3	3
muut					1	1
yhteensä		12	13	106	247	236

Ryhmä käytti lankapuhelimia ja kännyköitä työnsä apuvälineinä. Nykyisin on mahdollista käyttää myös Internetin kautta ohjelmia[69], jotka toimivat puhelimen tavoin. Työtehtävien sisältöön viittaaminen on hankalaa tietokonevälitteisessä kommunikaatiossa, ja siksi usein siirryttiin Irkki-viestinnästä keskustelemaan asiasta puhelimitse. Siirtyminen tapahtui myös vastakkaisessa järjestyksessä. Irkki-viestien jälkikäteen

[69] Voice over Internet Protocol eli VoIP. Katso tarkemmin http://fi.wikipedia.org/wiki/IP-puhe, tarkistettu 6.5.2006.

analysoiminen osoitti, että useasti oli siirrytty toiseen viestinnän muotoon. Ryhmällä oli monimuotoisia viestintävälineitä, jotka muodostivat erilaisia viestinnän tapoja. Tutkimusryhmän työn havainnointi muodostui monimuotoiseksi erilaisten keskustelu-areenoiden seuraamiseksi. Gzz-ryhmän jäsen esitti havainnoinnin aikana oman käsityksensä heidän työnsä seuraamisesta:

> Joo, varmaan vähän sama jos yrittäisi seurata tutkimusryhmää, kun istuisi kahvipöydässä ja katsoisi kun ihmiset tulee ja menee. Ja yhtäkkiä puheenaihe on joku ja puhuu jostain ihan muusta ja lähtee ja jatkaa sitä jossain tahoillaan. Sitten kun ne tulee takaisin on jo hypätty asiassa. (Tuukan kommentti 15.8.2002)

Lähdekoodin suunnittelu tapahtui useiden apuohjelmien avulla. Lähdekoodi kirjoitettiin tekstieditorilla. Käytetty ohjelmointikieli havainnoinnin aikana (20–21.5.2002) oli kahden ohjelmointikielen sekoitusta. Ohjelmoinnissa käytettiin Javaa ja Pythonia, josta käytettiin nimeä Jython. Suurin osa ryhmän jäsenistä käytti työssään Emacs-tekstieditoria. Lähdekoodin tuottaminen tapahtui seuraavalla tavalla havainnoinnin aikana. Ensinnäkin ohjelmoijalla oli päätteellään auki terminaali ikkunassa yhteys Gzz-ohjelmaan, joka sisälsi ohjelman komennot ja käskyt. Pääteikkunassa ohjelmoija puolestaan valitsi muokattavan tiedoston. Toiseksi ohjelmoijalla oli päätteellään auki kaksi tekstieditoria, jolla ohjelmoija käsitteli valitsemaansa tiedostoa. Toisesta päätteestä ohjelmoija katsoi mahdollisesti tiedostosta mallia, miten lähdekoodia oli tehty ja toisella tekstieditorilla hän kirjoitti uutta koodia. Ryhmän tekninen sisäinen raportointi näistä ohjelmoinnin tuloksista viestitettiin neljänä erilaisena tiedon muotona. Kuvaus tästä on esitetty ryhmän Irkki-keskustelussa 28.8.2002. Pääkehittäjä Lukka kuvasi neljällä toimintatavalla ryhmän Irkki-kanavalla raportoinnin muotoja, jotka ovat julkisesti esillä. Seuraavassa olen numeroinut raportointitavat ja kertonut niistä muutamalla lauseella:

1. Todo (aihealue)-viestejä ryhmän jäsenet lähettävät Commit–listalla. Viestit kuvaavat sosiaalista työnjakoa. Viesteissä asetetaan tehtävät; mitä kenenkin olisi tehtävä tai mitä itse teen. Tehtävänanto jakaa ohjelmointityön sisällön ohjelmoijien kesken.
2. Commit-messagella, ilmoitetaan lähdekoodin tullut muutos sähköpostiviestin muodossa. Commit-messagella ohjelmoija ilmoittaa lähdekoodiin mitä on muuttanut lähdekoodissa ja miksi on tehnyt niin. Muutos aktivoituu viestiä lähetettäessä.
3. Päiväraportti: ryhmän jäsen ilmoittaa pääkehittäjälle tai kuten Gzz-dev listalla (1.7.2002) lähtien, mihin on aikaansa käyttänyt. Viestin otsikossa päivänmäärä, viestissä ohjelmoija ilmoittaa mihin on aikaansa käyttänyt.
4. Read Me tiedostossa ilmoitetaan, mitä ohjelman käyttäjän tulee tietää ohjelmiston tiedostosta. Read Me tiedostot ohjelman sisällä kertoivat käyttäjälle tiedostojen funktionaalisuuden laadusta.

7.2 Kuvaus Gzigzagin työstä ja ohjelmistovirheen korjaamisesta maanantaina 20. toukokuuta 2002

Tässä luvussa esitän ajallisen koordinaation ideaa seuraten millaisista tapahtumista koostui Gzigzag ryhmän työpäivä maanantaina 20. toukokuuta 2002. Keskeinen havainnoinnin kohde ryhmän työssä oli miten ohjelmoijat tekivät uutta koodia, jäljittivät ohjelmistovirhettä ja yrittivät korjata sen. ”Tapahtumia” – tekstijaksoissa esitän ajallisesti projektin ohjelmoijien työtä, ja ”keskustelu” – tekstijaksoissa esitän ryhmän työpaikalla käytyjä keskusteluja. Omaa rooliani osallistuvana havainnoijana esitän jaksoissa ”keskustelu havainnoijan ja ohjelmoijan” välillä.

Ohjelmoijan työpäivän aamu alkoi lukemalla sähköpostiviestit muutoksista, jotka olivat tapahtuneet hänen edellisen työpäivänsä jälkeen. Kehittäjä Tuukka kertoi työpäivän alkamisesta ryhmän toimistolla maanantaiaamuna 20. toukokuuta 2002:

> Eli, mitä tässä nyt aamulla tulee ekaksi tehtyä. Tossa on ollut koko ajan Irkki ollu alhaalla auki ja sitten on sähköpostia tässä luen nytten. Ja sitten mulla on toisella työpöydällä on Debianin päivitys menossa, uusimmat paketit ja ei kai sitten muuta. Niin no täällä on CVS:stä uusin Gzz:n lähdekoodi ja pitää yrittää kattoa mitä on tapahtunu taas viikonlopun aikana. (Tuukan kommentti 20.5.2002)

Työpäivän alkaessa ohjelmistotyöläinen avasi päätteelleen suuren määrän erilaisia terminaali-ikkunoita:

> Hirvee määrä näitä terminaali-ikkunoita yleensä ja sitten on Irkki koko ajan tuolla taustalla. Ja mitäs? Selain on yleensä taustalla, kyllä sitä yleensä aina silloin tällöin tarttee ja sitten silloin kun kirjoittaa koodia niin Emacs. (Tuukan kommentti 20.5.2002)

Kehittäjän Irkki-kanava aktivoitui viestin saapuessa listalle:

> Tämä ohjelma näyttää sillä tavalla että jokainen kanava on yks tommonen välilehti, niin sitten se muuttuu se tekstin väri siitä, jos sinne tulee viesti.(Tuukan kommentti 20.5.2002)

Ohjelmoijien työpäivän aamu kului lukiessa läpi, millaisia muutoksia lähdekoodiin oli tehty commit-viestien muodossa ja millaisia ajatuksia oli vaihdettu kehitystyöstä Gzz-ryhmän listoilla sekä Irkki-kanavalla. Irkki-viestikanava avattiin omaa kehitystyötä varten. Kenttähavainnoinnin aikana Gzz-ryhmällä oli deadline keskiviikoksi saada ohjelman sovellus Soneralle testattavaksi. Maanantai ja tiistai työskenneltiin sovelluksen ohjelmoimiesessa testauskuntoon.

Iltapäivällä alkoi koodaus. Ohjelman sovelluksen testauskuntoon ohjelmointi oli Tuukan, Vesan ja Helsingistä verkon kautta työskentelevän Lukan vastuulla. Gzz-dev viesti-listalla oli viikonloppuna tullut Lukan viesti koskien keskiviikon Perheportaalitestejä. Hän oli kysynyt Vesalta missä vaiheessa työ oli. Tämä viesti on esitetty dokumentissa 4. Ryhmän Irkki-kanavalla oli keskusteltu maanantai-aamupäivänä testeihin liittyvistä asioista.

Iltapäivällä korjattiin ohjelmasta löytynyttä bugia. Tapahtumia ja keskustelu jaksot kuvaavat miten ohjelmoijat toimivat bugia korjatessaan.

Tapahtumia:
Tuukan ja Vesan yhteistyö alkoi keskustelulla klo 13.40.
Vesa ja Tuukka keskustelevat koodauksesta klo 13.40.
Keskustelun jälkeen Vesa oli puhelimessa klo 13.50 Lukan kanssa keskiviikon testauksesta.
Klo 14.10 Vesa ja Tuukka neuvottelivat työnjaosta Vesan päätteen luona ohjelma testauskuntoon valmistamiseksi.

Keskustelu klo 14.10:
Tuukka: Mitä se nyt on, osaaks se sanoo, mitä me tehtäs, onks se siinä vai ?
(se = Tuomas Lukka)
Vesa: Pitäs tehdä Simple Transcent testiavaruus perheportaaliin että olis testiavaruus.
Tuukka: Miks, miks?
Vesa: Ilmeisesti se oli siinä että silloin tiedetään , että meillä on varmasti toimiva avaruus. Silloin kaikki ne bugit mitkä tulee ei voi johtua että avaruudessa on jotain vikaa näis testeissä.
Tuukka: Niin.
Vesa: Jos me tehdään Unit-testejä ja avaruudessa voi olla jotain vikaa, niin se ei kerro ongelmista silloin itse clientissä.
Tuukka: Miten se tehdään ?
Vesa: Niin se ei ookaan niin selvä juttu. Mä veikkaisin että Jythonilla vois tehdä jonkun helpon,itse asiassa mua itteäni kiinnostais just se tehdä.

Hetken päästä paljastui, että testiavaruus ei toiminut:

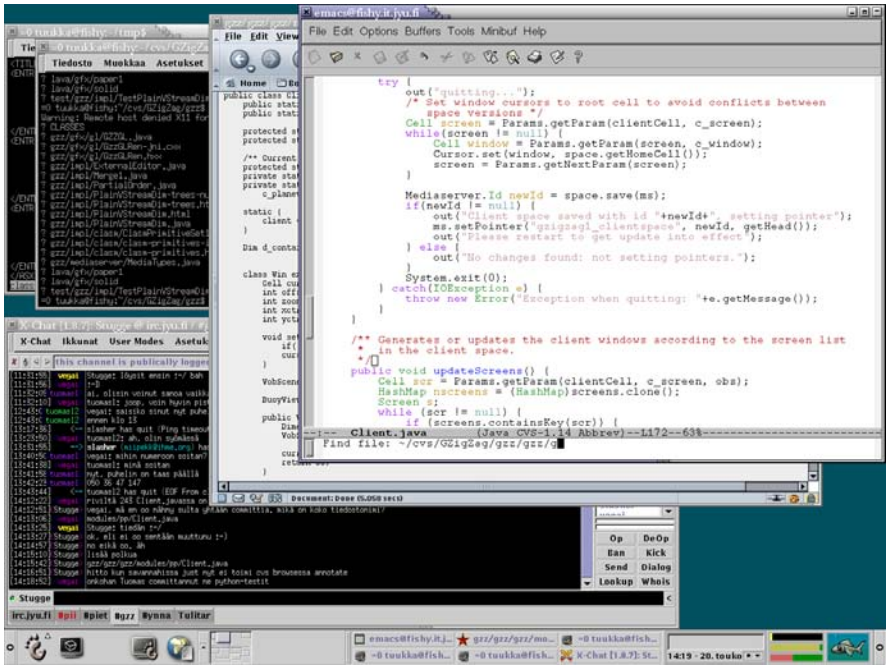
.....

Keskustelu klo 14.12:
Vesa: Se ei halua renderiä alussa. Hei kato joo se ei ota eventtiä, se ei tee yhtään mitään.
Tuukka: Hei jotain tollasta, me voidaan tehdä.
Vesa: Elikä koska täällä ei varmaan tuu mitään eventtiä.
Tuukka: Joo, mitä sä teet just nyt
Vesa: Mä ajattelin kattoo tota simple transinttiä.
Tuukka: Simple transinttiä.
Vesa: No niin, etten mä rupee sitä. Eli jos mä saan sen ajamaan sen clientin niin sitten on vai hyvä.
Tuukka: Joo, kyllä se ajaa sen, mutta tosiaan niitä eventtejä ei tuu niin se ei piirrä mitään ja mitään ei tapahdu.

Vesan ja Tuukan keskustelu kuvaa kasvokkain tapahtuvia keskusteluja ohjelmistotyössä. Kasvokkain käytiin lyhyt neuvottelu päätettäessä ohjelmoinnin osa-alueista ja siitä millainen tehtävä ohjelmoijalla on. Keskustelussa Vesa ja Tuukka saivat työnjaon tehtyä. He palasivat omille päätteilleen ja jatkoivat viestimistä Irkin avulla.

Vesan ja Tuukan päätetyö erosi toimintavoiltaan. Tuukka käytti työssään yhtä päätettä, jossa oli paljon eri terminaali-ikkunoita auki. Vesa puolestaan käytti työssään kahta päätettä. Yhdellä päätteellä hän luki ja kirjoitti pelkästään Irkkiä ja toisella päätteellä hän teki kaiken muun. Vesa kertoi havainnoijalle, mikä oli maanantai-iltapäivänä hänen työtehtävänsä sisältö.

Keskustelu Vesan ja havainnoijan välillä:
Vesa: Lähinnä, joo mä pitäs kirjoittaa testejä siihen Pp-juttuun, nyt mä katon lähinnä esimerkkiä vanhoista testeistä että miten niitä on tehty.
Havainnoija: Siis Tuomakselle keskiviikoks Perheportaaliin ?
Vesa: Joo, tosiaan semmosia testejä mitkä, kun kirjotetaan testit niin ohjelmaa voi vapaasti muuttaa sinänsä, koska ne testit paljastaa jos rikkoo niitä. Yksikkötestausta.
Havainnoija: Onks Tuomas siis, esittääks se jotain, siellä Soneralla just jotain tähän liittyvää?
Vesa: Joo, tai idea on että me viedään Soneralle toi niiden käytettäväks, kyllä se tietysti esittää samalla sitten.



Kuva 1: Tuukan näyttöpääte maanantaina klo 14.19

Kehittäjät keskustelivat tarkemmin siitä millainen testiavaruus tehdään. Kehitystyön suunnasta keskusteltiin puhelimessa Lukan kanssa.

Tapahtumia:
Klo 14.27 Vesa soittaa Lukalle, mutta Lukka ei ehdi keskustella.
Klo 14.33 Vesa soittaa uudestaan lankapuhelimestaan Lukalle, puhuvat keskiviikon testeistä. Vesa tekee samalla merkintöjä paperille. Samaan aikaan Tuukka puhuu omaa kännykkäpuheluaan.
Klo 14.40 Lukka haluaa puhua Tuukan kanssa, Vesa antaa luurin Tuukalle. Vesa poistuu siksi ajaksi. Tuukka puhuu Lukan kanssa viisi minuuttia. Tuukka kommentoi puhelua jälkikäteen : ” Tuomaksella tosi kiire, puhui ratikasta tai junasta oli kovaa hälinää.” Tuukka tekee commitin klo 14.45, eli Tuukka muuttaa Perheportaaliin liittyvää tiedostoa, samalla syntyy viesti Savannahiin ryhmän listalle. Ohjelmasta on löytynyt bugi, joka täytyy jäljittää ja korjata että ohjelma saadaan jälleen toimimaan.

Vesa tuli hetken päästä keskustelemaan Tuukan työpisteen luo, koska hänellä Perheportaali ei enää toiminut. Tuukka kommentoi: ”*Poistin yhden rivin, yritän korjata.*” Tuukka kommentoi havainnoijalle koodiongelmaa:

Keskustelu Tuukan ja havainnoijan välillä:
Tuukka: Tutkii sitä ja mä kysyin Tuomakselta silleen että liittyisköhän se kenties siihen, ettei se toimi, ja Vesa sano että ei hän on lisännyt sen rivin sinne että Tuomas on jo korjannut, mutta Tuomas ei oo poistanu sitä riviä sieltä. Niin se on haussa. Että vaikka se dokumentaatio olis niinku suoraan siinä koodin seassa niin se ei siltikään ole ajan tasalla koskaan.
Havainnoija: Miten tommonen ongelma, onks se helpointa ratkaista puhelimessa tai siis ?
Tuukka: Ei sitä voi ratkaista oikein mitenkään. Siis ainut tapa ratkaista se ois että se ois sillon kun se on fiksattu niin se ois kanssa niinku otettu pois siitä.

Hetken päästä Vesa ja Tuukka käyvät lyhyen keskustelun liittyen bugiin, jota Tuukka selvitti havainnoijalle.

Vesan ja Tuukan keskustelu kasvokkain:
Tuukka: Siis sä et oo laittanu sinne semmosta riviä, että get card sys palauttaa aina miinus yks.
Vesa: Oon, mutta sitä ei oo enää siellä se on vanha kommentti. Tuomas korjas sen, mutta se ei poistanu sitä committia.
Tuukka: Joo, okei.
Vesa: Mä tein committia sen jälkeen niin voit poistaa jo sen kommentin.

Tuukka selvitti havainnoijalle CVS:n käyttöä, miten sen avulla hän jäljitti tätä bugia, joka tuotti Vesalle ja Tuukalle päänvaivaa. Samalla Tuukka esitti havainnoijalle mistä bugi löytyisi. (kts. edellä dokumentti 7.1).

Tuukan ja havainnoijan keskustelu:
Tuukka: Joo, yrittää ottaa selvää mitä muutoksia on tapahtunut, Tuomas on korjannu jotain kommentti on "clean up" .[Dokumentti 7.1]
Havainnoija: Pitää ite selvittää ?
Tuukka: Pitää yrittää kattoo.
Havainnoija: Löytyks se mikä oli äskön, miks Vesalla ei toimi ?
Tuukka: Ei siis, ei ehtinyt vielä sitä kattoo.
Havainnoija: Mites se vertailu tapahtuu ?
Tuukka: Ei kun siis just sillä mitä katottiin CVS:stä, että mä voin kattoo.
Havainnoija: Ai sieltä, niin niin joo missä näkyy ne koodin muutokset sieltä.
Tuukka: Siis tästähän ne näkyy, että tää mikä on ollu punasta niin se tarkottaa, että Tuomas on ottanut tän osan pois.
Havainnoija: Joo, joo elikä tosta suoraan.
Tuukka: Ja sitten on lisännyt tämmösen.
Havainnoija: Vihreellä on lisäykset.
Tuukka: Sitten se on muuttunu valkoseks.
Havainnoija: Keltanen on muutos.
Tuukka: Tosta näkee, että se on poistanut tommosen alueen, tuolla on committi merkin alku, mutta se ei koskaan poistanu sitä, toi on committi merkin alku, mutta commitin lopetusmerkkiä ei täältä koskaan löydy, siinä on nyt se vika.
Havainnoija: Joo. Tekeeks toi ohjelma sen automaattisesti, ettei ite tarvi värjätä noita.
Tuukka: Ai mitä tekee ?
Havainnoija: Tekeeks se, automaattisesti siis. Miten joku värjää noi alueet siis ?
Tuukka: Ei kun siis se värjää ne.
Havainnoija: Ei pysty tapahtumaan ihmisen virhettä.
Tuukka: Joo. Siis tässä on esimerkki semmosesta commitista mikä on ollut, tässä on pätkä koodia, joka ei tällä hetkellä toimi niin se on kommentoitu pois.Tää näyttää sen tälleen eri värisenä, se on aika selkee kattoo. Mutta sekin on kommentoitu väärin.

Puoli minuuttia myöhemmin:
Tuukka: Nyt se varmaan taas käänty. Tuomaksella on varmaan ollu tosi kova kiire, kun se on tehnyt tota. Kun se soitti se oli vissiin jossain ratikassa tai junassa.(-) Eli nyt mä oon muuttanu ne ja kääntäny itte sen ja nyt mä haluan ne muutokset Vesalle sinne palvelimelle että se voi päivittää sieltä. Mitä mä laitan tähän viestiin sille?
Tapahtumia :
Kello 15.22 Janne saapuu työpisteelleen.
Kello 15.35 Lukka soittaa Jannelle kännykkään, Janne palaa viiden minuutin päästä takaisin työpisteelleen.
Kello 15.45 Vesa ja Tuukka käyvät lyhyen keskustelun Perheportaaliohjelman toimimisesta ja koodin päivittämisestä.
Kello 15.50 Tero saapuu myös töihin. Kaikki työskentelevät keskittyneesti päätteellään. Työn keskeyttävät vain kahvin hakeminen kahvitilasta.
Kello 16 Janne hakee kahvia ja juo sitä kahvitilassa, Vesa puolestaan hakee myös kahvia, mutta juo sitä päätteen ääressä.
Kello 16.30 Tuukka ja Vesa keskustelevat jälleen lyhyesti Perheportaalista.
Kello 16.40 jälleen Tuukka ja Vesa keskustelevat bugiongelmasta.

Sosiaalinen kommunikaatio työpaikalla oli vähäistä deadlinen kynnyksellä. Vesan ja Tuukan muutamat lyhyet keskustelut rikkoivat hiljaisuuden. Janne puuhaili paperitekstuureita, jotka oli tarkoitus myöhemmin lisätä Perheportaaliin. Tero puolestaan selasi sähköpostia, Irkkiä ja Internettiä. Pääkehittäjä Lukka oli siirtänyt työskennellessään Helsingistä verkon kautta suuren osan koodia eri paikkaan, ja tämä tuotti Vesalle ja Tuukalle ongelmia Jyväskylän Agorassa.

Tuukan ja Vesan keskustelu kasvokkain:
Vesa: Eih.
Tuukka: Hei se on siirtäny kaiken paikasta toiseen. Soita sille ja kysy.
Vesa: Tota missä se saa nyt sen alkunsa, tuleeks se siellä startissa jotenkin ei kun rundissa.
Tuukka: Joo, se ettii sen, tässä
Vesa: Mihin sä pistit?
Tuukka: Find example data. Se vaan cursor piste se.
Vesa: Niin nyt mä tajuan siis se haluu että tää homma tehdään siellä ylemmällä tasolla sen takia se haluu constructorin missä se on nyt, siis se ei pitäis olla ton clientin tehtävä ilmeisesti.
----------.

Tuukan ja Vesan keskustelu kasvokkain hetken päästä uudestaan:
Tuukka: Täällä on poistettu nimittäin ne kaikki, ei olekaan muuten. Tuollainen siis pitäisi olla, mitäköhän siinä pitäisi olla tilalla.
Vesa : (?)
Tuukka: Itse asiassa ne muut on kommentoitu pois. Muistaakseni tai on ne jossain vaiheessa poistettu. Tai siis ne on siirretty siis. Mä laitan ne väliaikaisesti sinne.
[Vesa kävelee pois]
Vesa: Joo

Tapahtumia:
Kello 17.05 Tero poistuu töistä.
Kello 17.10 Vesa ja Tuukka käyvät lyhyen keskustelun kasvotusten koodaamisen kanssa on edelleen on ongelmia.

Tuukan ja Vesan keskustelu kasvokkain:
Vesa: Toi on muuttunut siitä kun.
Tuukka: Mutta miten se on mahdollista ?
Vesa: Ei oo.
Tuukka: Siis ensinnäkin se että mistä on ennen saatu ne arvot niille ulottuvuuksille.
Vesa: Rakenteesta.
Tuukka: Niin mutta missä se koodi on. Ja sitten toinen on se toi että miten sille renderille voidaan antaa se Window-solu jos sellaista Window-solua ei ole.

Sit mä katoin tota, sitä yritin kattoo tota esimerkkiä ei täältä löydy mitään sellasta, koska siis (-) voeb vanishing client.
Tapahtumia:
Kello 17.15 ilmenee tekninen ongelma:ohjelmistotekniikan laitoksen verkko menee epäkuntoon, joten Irkki-työskentely ei enää onnistu. Vesan ja Tuukan on helppo tehdä päätös, että he jatkavat työskentelyä kotoa ja viestivät Irkillä toisilleen.
Kello 17.20 Vesa ja Tuukka lähtevät polkupyörillä Agoran pihalta kohti kotejaan.

Luvun lopuksi käsittelen yhteenvedon omaisesti luvun 7 tapahtumia. Havainnointiluku kuvasi, kuinka ryhmän ohjelmoijille hakkerointi (hacking) on elämänmuoto. Ryhmän sosiaalisesti välittyneet työtavat ovat julkisesti esillä Internetissä ohjelmoijien saatavilla. Keskusteluesimerkit kuvaavat kuinka ohjelmoijien työssä tekniset termit ja neuvottelut muodostuvat omasta työkielestä. Työhön liittyi myös hakkeroinnin ilo, se oli elämäntapa, josta ohjelmoijat saivat myös pienimuotoista palkkaa. Gzz-ryhmän ohjelmistotyön tapoja ohjasivat verkkoviestinnän ja ohjelmoinnin työvälineet. Ohjelmistotyön tärkein työpaikka oli tietoverkko, jossa ryhmän työvälineet ja dokumentit ovat. Kirjalliset viestit eri muodoissa lähetettiin jaettuun tietoverkkoon, jossa ohjelman kehitys tapahtui. Tietoverkossa yhteistoiminta perustui aktiivisuuteen keskustelulistoilla ja Irkki-kanavalla. Yhteistoiminnan dialogisuus loi tehtävät jokaisen itsenäiselle ohjelmointityölle. Gzz-ryhmän kehittäjävirtuaaliyhteisössä yhteistoiminnan sosiaalinen aktuaalisuus muodostui ohjelmoijien yhteistoiminnan dialogisuudesta. Irkissä ja sähköpostissa voi olla kirjautuneena 24 tuntia vuorokaudessa seuraten viestejä tai osallistumalla keskusteluun ohjelmoinnin ohessa. Kehittäjän ”*työpaikalla*” oloa kuvasi hyvin ryhmän jäsenen humoristinen viesti Irkki-kanavalla:

> 2002-09-02 12:06:13 tuomasl: Ahhhh! Olen taas elossa (tai siis netissä) (tai siis sama asia...)

Tämä kehittäjän kommentti kuvasi työn määrittymistä tietoverkon toimintatapoihin. Työpaikalla työskentely määrittyi verkkodialogiin osallistumisen aktiivisuudella sekä omien tehtäviensä suorittamisena niistä raportoimalla. Työn suorittaminen oli riippuvainen verkkoyhteyksistä. Havainnointiesimerkissä laitoksen tietoverkon katkos lopetti työskentelyn siltä päivältä verkossa. Kun julkista ja avoimen toiminnan aluetta projektin työlle ei ollut Internetissä, ei työskentely ollut enää mahdollista.

8. KUINKA VAPAA OHJELMISTOKEHITYS MUUTTUU? : Gzigzagista Fenfireen – tutkimusryhmän projektit 2002 –2004

> *TL: Jotkut on saattanut nähdä jonkun toisen nimen tällä meidän projektilla eli aikaisemmin tätä on kutsuttu nimellä Gzigzag ja Gzz ja näihin vaihtoihin on ollut syynä se, että yhteistyö Ted Nelsonin kanssa loppui ja siinä oli... noh en nyt viitsi mennä tarkemmin siihen, että mitä kaikkea siinä... siinä tapahtui. Kaikilla on oma versionsa tarinasta ja silleen. (Tuomas Lukka esitellessään ryhmän ohjelmistoprototyyppiä It Expo- tapahtumassa 27.8.2003)*

Tuomas Lukka kommentoi edellä olevassa lainauksessa, että yhteistyö Ted Nelsonin kanssa yhteistyö loppui, mutta ei kerro yleisölle tilaisuudessa tarkemmin mitä Gzigzag-ryhmän ja Nelsonin välillä tapahtui. Tässä luvussa esitetään erilaisiin laadullisiin aineistoihin perustuva versio siitä *”mitä kaikkea siinä tapahtui ”* – eli mitä Hyperstruktuuri-tutkimusryhmän projekteissa tapahtui vuoden 2002 alusta vuoteen 2004. Tutkimusprojekti Gzigzag oli osa InBct hanketta Jyväskylän yliopistolla vuosina 2002–2003[70]. Aineistona tässä luvussa on käytetty ohjelmistoprojektin dokumentaatiota, kenttähavainnointia ja ohjelmoijien haastatteluja. Keskeinen teema on mitä arvoja ja merkityksiä ryhmän ohjelmoijat antavat muutoksien yhteydessä kun ohjelmistoprojektin vapaus ja tutkimustyö joudutaan ongelmien edessä määrittelemään uudelleen. Kuinka tärkeätä on ja millaisia merkityksiä ryhmän jäsenille antaa sitoutuminen vapaitten ohjelmistojen arvomaailmaan?

Helmikuussa 2002 ryhmän pääohjelmoija Lukka oli Kauppalehden Optioliitteen kannessa otsikolla Suuria Odotuksia[71]. Artikkeli kertoi päähenkilön tieteellisen uran menestyksekkäästä taustasta ja Zigzagin potentiaalista hypertekstuaalisena

---

[70] Innovation in Business, Communication and technology – tutkimusohjelma sisälsi vuonna 2002 erilaisia osaprojekteja liittyen tietotekniikan ja ihmisen vuorovaikutuksen tutkimiseen. Projektit yhdistivät viestintätieteiden, käyttäytymistieteiden, liiketaloustieteiden ja informaatioteknologian näkökulmat ja lähestymistavat innovaatioiden kehittämiseen. Osaprojekteja oli neljä, joista Gzigzag oli osa projektia: Käyttäjä verkottuneessa toimintaympäristössä (projekti 2). InBct projektin yksi johtohahmoista oli vararehtori, professori Pekka Neittaanmäki, joka oli Tuomas Lukan esimies Jyväskylän yliopistolla.

[71] alaotsikko Jukka Saastamoisen artikkelille (sivut 26-31 Kauppalehden Optio-liitteessä 7.2.2002) on ”Tohtori Tuomas Lukka, 27, johtaa tutkimusryhmää, jonka tehtävänä on selvittää ihmisen ja tietotekniikan välistä suhdetta.”

tiedonhankintamenetelmänä. Artikkelin kirjoittajan mukaan Pekka Neittaanmäki[72] oli kuvannut Lukan hyvin lahjakkaaksi tutkijaksi:

> PN: Lukalla on ilmiömäinen taito omaksua uusia asioita ja yhdistellä niitä suuremmiksi kokonaisuuksiksi.

Artikkelissa kerrottiin Lukasta poikkeuksellisen lahjakkaana tutkijana. Hänen työhuoneessaan ei ollut paperipinoja, vaan kannettava tietokone. Työskennellessään hän kuunteli usein klassista musiikkia. Puolipäiväisesti yliopistolla työskentelevä Lukka ohjasi tutkijaryhmäänsä ja tietojenkäsittelykurssia pääasiassa Internetin kautta. Hän vietti Jyväskylässä vain päivän tai kaksi viikossa. Artikkelissa todettiin, että asiantuntijoiden mukaan Zigzagiin liittyy suuria mahdollisuuksia, mutta tutkimustyö on vasta alkuvaiheessa. Artikkelissa lainattiin Lukan kommenttia Gzigzag-ryhmän kehityksen tilanteesta:

> TL: On maailmassa muitakin hypertekstin ja kokonaisvaltaisten tietoympäristöjen tutkijoita. Mutta me olemme pidemmällä.

8. maaliskuuta 2002 saavuin havainnoimaan ryhmän kokousta aamupäivällä Jyväskylän yliopiston Agoraan. Ryhmän tiloihin saavuttuani kuulin Lukalta heti, että edellisenä päivänä oli tullut sähköposti Nelsonilta. Nelson oli lakimiehensä innoittamana kieltänyt ryhmää käyttämästä Zigzag-tavaramerkin nimeä ohjelmistonsa nimessä. Nelson ei kokenut kontrolloivansa projektia tarpeeksi hyvin, siksi projektin nimen täytyisi muuttua. Lukan mukaan yhteistyö Nelsonin kanssa oli ollut viime aikoina vaikeaa, koska hän ei esimerkiksi vastannut sähköposteihin koskien ohjelmistomäärittelyjä[73]. Ryhmän kokouksessa pohdittiin ensin ohjelmiston teknisiä ajankohtaisia kysymyksiä, joita ohjelmoivat visualisoivat piirustuksien kanssa liitutaululle. Puolen tunnin kuluttua Lukka aloitti keskustelun nimiasiasta:

> TL : Siinä tietysti voisi alkaa tapella, mutta en jaksa. En näe mitään hyötyä mitä siitä saisi. Eli että sanoisin että muutetaan vaan.

Ensimmäinen yleinen ehdotus nimiasiassa oli, että jos nimilyhennys Gzz kelpaisi Nelsonille, se valittaisiin projektin uudeksi nimeksi. Muilla ryhmän ohjelmoijilla oli toisenlaisia nimiehdotuksia, joiden perusteena oli että täytyisi päästä kokonaan eroon Nelsonin ideoihin viittaavasta nimestä. Projektin nimiasiasta ei kokouksessa tehty

---

[72] Pekka Neittaanmäki oli Jyväskylän yliopiston vararehtori ja tietotekniikan laitoksen professori vuonna 2002.

[73] Keskustelu Tuomas Lukan kanssa ennen ryhmän kokousta 8.3.2002 perustuen kenttämuistiinpanoihin. Kenttähavainnoinnin aikana 8.3.2002 nauhoitin vain ryhmän kokouksen.

päätöstä, vaan todettiin että keskustelua jatkettaisiin vielä Internetissä. Noin kuukautta myöhemmin 14. huhtikuuta 2002 ryhmä ilmoitti kotisivullaan ohjelmistonsa uudeksi nimekseen Gzz[74]. Muutamaa viikkoa aikaisemmin Gzz oli rekisteröity Savannah palvelimelle avoimeksi ohjelmistoprojektiksi[75]. Nelson oli halunnut ettei Xanadun yleisellä keskustelulistalla enää keskusteltaisi liian yksityiskohtaisesti Gzigzag-projektin teknologiaan liittyvistä asioista:

> Se meni ihan siihen että meidän piti perustaa ihan oma, oma sähköpostilista sitä kehitystä varten kun ennen. Kun ei ollut mitään muuta Zigzag-kehitystä kun tämä meidän, niin tämä on ollut siellä xanadu.com:n sähköpostilistalla. Niin nyt kun Benja oli lähettänyt jonkun viestin ja se oli osaksi yleistä asiaa ja osaksi spesifistä, niin sitten Ted oli vastannut vihaisesti siihen, että hänellä ei ole aikaa lukea tällaisia juttuja, mitkä liittyy tämmöisiin epäortodoksisiin väsäyksiin. (Tuukan haastattelu 20.5.2002)

15. toukokuuta ryhmä perusti tämän seurauksena keskusteluilleen Gzz-development[76] sähköpostilistan ohjelmistonkehitykseen liittyville asioille. Heinäkuun 1. päivästä lähtien ryhmän jäsenet kirjoittivat kyseiselle listalle päiväraportin omasta työstään muita ohjelmoijia varten. Kesän lopulla elokuun 25. päivänä pääkehittäjä Lukka ideoi viestissään, miten ohjelmistoon ja sen arkkitehtuurin kehitykseen liittyviä asioita käsiteltäisiin:

> Päätin että on aika luoda PegBoard–ehdotukset Gzz:n parantamiseksi, keneltä tahansa (jopa minulta) ei tule hyväksyttyä ilman selittämättä muutoksia Pegissä. Pegien ei tarvitse olla mutkikkaita – vaan selkeitä, tervejärkisiä selityksiä, mitä pitäisi tehdä ja mikä olisi tarpeeksi. Kun Peg on sisällytetty varastoon (CVS:ään), siitä tullaan keskustelemaan ja tarkistamaan, ja lopuksi minä arvioin konsensuksen ja teen päätöksen. (Tuomaksen sähköposti ryhmän listalla 25.8.2002)

Ryhmän ohjelmoijat kommentoivat ideaa Pegistä sähköpostilistalla ja Peg päätettiin ottaa käyttöön. Näin syntyi Pegboard[77] ohjaamaan projektiin liittyvien ideoiden ja ongelmien käsittelyä (tekstin luettavuuden vuoksi käytän tekstissä Pegboardista termiä ehdotustaulu sekä Pegistä termiä ehdotus). Ehdotustaulun tyylistä työmuotoa oli käytetty aikaisemmin Pythonin kehitysprojektissa ja sen alkuperäinen käyttötapa oli peräisin

---

[74] http://www.nongnu.org/gzz/, tarkistettu 17.10.2005.

[75] http://savannah.nongnu.org/projects/gzz/, tarkistettu 17.10.2005. Savannah palvelimelle rekisteröityi satoja Free Software ohjelmistoprojekteja. Määrittelyissä pystyi ilmoittamaan, onko osa Gnu-projektia vai ei. Gzz rekisteröityi ei-Gnu projektiksi.

[76] arkistot Gzz -dev viesteistä kuukausittain, http://lists.gnu.org/archive/html/gzz-dev/ , tarkistettu 7.11.2005.

[77] Pegboardilla käsitellään systemaattisesti ohjelmiston ideoita ehdotuksina ja keskusteluna. Yksi aihe (peg) on taululla (pegboard). Pääkehittäjän päätösten mukaan keskustelu etenee seitsemän erilaisen statustilan alla. Pegboard on tiedosto, jossa käsitellään mitä tahansa ohjelmistoon liittyvää asiaa demokraattisesti, ja pyritään pääsemään keskustelun jälkeen konsensukseen, jonka pääkehittäjä kokoaa päätökseksi. Pegboard tulee sanoista; index of Proposals for Enchancing Gzz. Gzz pegboard osoitteessa http://himalia.it.jyu.fi/gzzdoc/pegboard/, tarkistettu 6.5.2006.

Internetin RFC-prosessista[78]. Ehdotustaulusysteemi lisäsi projektin ryhmän ohjelmistoideoitten kehittämistä dialogisesti organisoidulla työtavalla. Tuukka kommentoi haastattelussa ehdotus-toimintatavan lisänneen ryhmän työtapojen organisoitumista:

> On niin kuin kasvanut että just on toteutettu se Pegboard systeemi CVS:ään. Missä voidaan tavallaan pitää avoinna olevia kysymyksiä esillä ja sitten että mikä on se ongelma silleen aika tarkasti määritelty. Mikä on ongelma, miten sitä voitaisiin ratkaista, mitä siitä seuraa. Ja sitten lopuksi se mitä yhteisesti on, mihin lopputulokseen on päädytty. (Tuukan kommentti projektin asioiden käsittelytapaan haastattelussa 28.11.2002)

Ehdotusten päättäminen ja ratkaiseminen pysyivät yksin pääkehittäjä Lukan hallinnassa. Hermanni kommentoi ehdotusten hyväksymisprosessin tapahtuvan pääkehittäjän päätösvallan mukaan:

> Ja se on yleensä ihan tämmöisiä niin kuin arkkitehtuurihommia yleensä kaikki, mitä kuuluu ohjelmistotuotantoon…Tämmöinen ehkä puoliformaali menetelmä, jolla sitten niinku yritetään standardoida tiettyjä asioita. Ja sitten sen Pegin tekijä yrittää selvitellä ne. Ja sitten kun ne on selvitetty, niin se yleensä hyväksytään ja ... se aina vaatii... Pegin hyväksyntä aina vaatii Tuomaksen hyväksynnän..... Tuomas sen hyväksyy sitten, jos joku hyväksyy sen. (Hermannin haastattelu 23.5.2003)

Kehitystyötä oli helpottanut syksyllä 2002 Irkki-bottitekniikan[79] luonti. Irkki-automaatin toiminta käytännössä tarkoitti, että ohjelmistopäivityksestä sähköpostilistojen otsikkojen viestirivi näkyi mahdollisimman nopeasti ryhmän Irkki-kanavalla. Tuukka kommentoi Vesan kirjoittaneen Irkki-automaatin :

> Se oli tosi mukavaa, kun Vesa kirjoitti semmoisen Irkki-botin, joka tulee siis Irkki-kanavalle yhdeksi keskustelijaksi ja sitten se aina ilmoittaa siellä kanavalla kun joku committaa CVS:ään jotain. Se oli ihan semmoinen käytännön syykin, kun Savannahin sähköpostilistat[80] oli niin hitaat, että kesti puoli tuntiakin että muut sai edes sähköpostina viestin siitä että joku oli committanut jotain. (Tuukan haastattelu 28.11.2002)

Syyskuussa 2002 Nelson[81] antoi yllättäen kommentteja ryhmän sähköpostilistalla uusimman kehitysversion testaamisesta. Viestissään hän kehui ajatuskarttanäkymiä[82] ja ilmoitti bugista. Hän ehdotti samalla korjauksia näppäimistökomentoihin. Viestin alku oli luonteltaan ystävällisen kehuva:

---

[78] Henkilökohtainen sähköposti Benja Fallensteinin ja Juha Lemisen välillä 18.3.2004 Pegien taustasta. RFC:t ovat sisällöltään ja asemaltaan hyvin kirjava joukko asiakirjoja, jotka kuvaavat Internetin erilaisia käytäntöjä, lähinnä teknisten menettelyjen järjestelmiä eli protokollia. Ensimmäinen RFC luotiin Kalifornian yliopistossa vuonna 1969.

[79] Bot on lyhenne termistä robot, automaattinen. Siksi käytän botista termiä Automaattinen.

[80] Savannahin palvelin toimi Yhdysvalloissa.

[81] Ted Nelsonin viesti ja Benjan vastaus viestiin. Verkossa http://lists.gnu.org/archive/html/gzz-dev/2002-09/msg00223.html, tarkistettu 3.4.2006.

[82] Ryhmän ohjelmoijista Asko oli kehittänyt ajatuskarttamaisia näkymiä, joita Nelson intoutui kehumaan. Katso Mind Mapin historiasta välineenä, http://fi.wikipedia.org/wiki/Miellekartta, tarkistettu 6.5.2006.

> Hei, Tuomas ja muut. Käytin useita tunteja versiota 0.63 Windowsissa, ja täytyy sanoa, että se on upean näköinen ja erittäin vakaa.

Kommentit olivat kannustavia ja yhteistyö jatkui. Nelsonin ystävälliset kommentit paransivat Gzigzagin ja Nelsonin välistä luottamusta. Lokakuussa ryhmä esitti ohjelmiston kehityksestään demon InBct-projektin johtoryhmälle, joten motivaatio saada 0.8 versio valmiiksi seuraavaan kevääseen mennessä oli suuri.

28. marraskuuta 2002 ryhmän ohjelmoijia haastateltiin ryhmän tiloissa. Ryhmään oli liittynyt syksyllä 2002 kolme uutta jäsentä, jotka kertoivat liittymisestään projektin jäseniksi. Kaikki olivat tutustuneet Lukkaan eri tavalla. Ensin heidän oli täytynyt opetella perusasiat ohjelmoinnista. Matemaattisin algoritmeihin ja ohjelmistokieliin liittyvä perusosaaminen oli vaativaa itseopiskelua. Uusilla jäsenillä oli erilaisia tehtäviä projektissa: yksi ohjelmoija keskittyi Perheportaalin kehittämiseen, toinen kehittämään uudenlaisia graafisia näkymiä ohjelmistoon ja kolmas pro gradu- tutkielmaa kirjoittamaan sekä kehittämään ohjelmiston tieteelliseen tutkimukseen liittyviä teemoja. Havainnointini yhteydessä ohjelmoijat esittelivät projektin ensimmäisen visiopaperin, joka tiivisti projektin ideat. Visio kertoi projektin vapaiden ohjelmistojen arvoihin sitoutumisesta. Visiossa kerrottiin ryhmän vapaasta toimintamallista seuraavaa:

> Gzz koostuu kahdesta osasta tutkimuksesta ja ohjelmointitekniikasta…tavoitteena valmistaa työkaluja tietotyöläisille…Vapaa ohjelmisto (Free Software). Lisenssimme ovat pääasiassa LGPL, arvokkaimmissa innovaatioissamme GPL…jos koskaan hankimme patentteja, myönnämme niille maailmanlaajuisen peruuttamaton patentin GPL-ohjelmistoihin…Meitä rahoittavat yritykset, jotka joko omistavat tai lisensoivat nämä patentit, meillä on yhteisymmärrys, että ne myöntävät GPL oikeudet, eikä tämä heikennä heidän liiketoimintaetujaan.

Visiossa mainittiin myös että rahoituksen saamiseksi ryhmän projektille olisi tärkeää pitää näyttäviä demoja silloin tällöin. Ryhmä listasi kehitystavoitteisiinsa valmistaa käyttökelpoinen prototyyppi jatkokehitystä varten. Myös käytettävyystestaukset, uudet julkaisut ja yhteistyö yritysten kanssa innovaatioiden soveltamiseksi olivat keskeisiä asioita ryhmän toiminnalle tulevaisuudessa.

8.1 Nelson: ”En koskaan olettanut olevani Open Source projektissa.”

Vuoden 2003 alusta pääkehittäjä Lukka työskenteli kokopäiväisesti projektissa. Ryhmä jatkoi kehitysprojektina InBct-hankkeessa[83] ja ryhmän saksalainen jäsen Benja saapui Suomeen opiskelemaan Jyväskylän yliopistolle ja työskentelemään projektissa.

[83] Ryhmän posteri InBct-hankkeessa vuodelta 2003 liitteessä 3.

Helmikuun 9. päivänä 2003 yhteistyö Nelsonin ja Gzz välillä kärjistyi. Nelson esitti vaatimuksia ja luottamus yhteistoimintaan oli vaarassa. Sähköpostiviestissään Nelson[84] kertoi, ettei hänellä ollut alunperin tarkoitus panostaa Gzz-projektiin kuin vuodeksi. Hän omaksui keskeisen kantansa Gzz-tietorakenteen vapaaseen kehittämisen. Hän sanoi, ettei hän halua antaa omistamaansa Zigzag-tiedostorakenteen Yhdysvalloissa rekisteröityä patenttia Open Source lisenssien mukaiseen käyttöön. Nelson oli hakenut vuonna 2001 Yhdysvalloissa rekisteröidyn patentin Zigzag-tiedostorakenteelle. Hän tarjosi kuitenkin mahdollisuutta neuvotella kanssaan vuoden mittaisesta vapaasta lisenssistä. Nelson ehdotti, että ryhmän tulisi maksaa vapaan maailmanlaajuisen lisenssin saamiseksi jonkinlainen rahallinen korvaus. Hän toteaa ehdotuksensa olevan vastoin vapaata ohjelmistokehitystä:

> …ajattelen, että se rikkoisi GPL mallin .

Lisenssiasiasta oli keskusteltu noin kuukausi aikaisemmin Nelsonin ja ryhmän välisissä henkilökohtaisissa sähköpostiviesteissä. Ryhmän jäsen Benja oli ehdottanut Nelsonille viestissään 16. tammikuuta seuraavasti:

> Mahdollinen ratkaisu olisi lisätä lisenssiin rajoitus, ettei sitä saisi käyttää tai jaella Yhdysvalloissa. GPL / LGPL lisenssit mahdollistavat tämän.

Nelson vastasi, että hän on käyttänyt Xanadu-projektiin 40 vuotta elämästään, eikä halua luopua ideoistaan täysin ilmaiseksi. Hän koki nyt Gzz-projektin pääkilpailijakseen, koska hän oli itse siirtynyt muihin kehitysprojekteihin:

> Oma harkitsemattomasti annettu osuuteni Gzz-projektiin on tehnyt teistä pääkilpailijani, vaarantaen kaiken jonka vuoksi olen työskennellyt…Olisi helpotus, jos lopettaisitte ZZ-rakenteen[85] kehittämisen. Luulen että teidän on vaikea lopettaa, mutta se olisi suuri helpotus minulle.

Nelsonin mukaan hänen ja Lukan välillä sovitussa yhteistyössä oli ollut huomattava väärinkäsitys. Hän totesi olevansa pettynyt yhteistyöhön ja sen tulokseen:

> Luulin olevani tavallani vastuussa projektista, ja tarvitsin välttämättömästi luotettavaa systeemiä muistiinpanoilleni…en koskaan olettanut olevani ”Open Source projektissa”, vaan vastavuoroisessa järjestelyssä Tuomaksen kanssa... Toivoin saavani nopeasti pienen systeemin ja tuen sen käyttöön, mutta tuntuu etten saanut mitään.

---

[84] Nelsonin ”harkittu” vastaus ryhmän jäsenen Benjamin Fallensteinin 16.tammikuuta 2003 sähköpostiviestiin. Url : http://lists.nongnu.org/archive/html/gzz-dev/2003-02/msg00042.html , tarkistettu 9.11.2005.

[85] Yhdysvalloissa rekisteröity patentti ZigZag-tiedostorakenteelle : United States Patent numero 6,262,736 , rekisteröity heinäkuun 17, vuonna 2001 : Interactive connection, viewing, and maneuvering system for complex data.

Benja oli esittänyt tammikuun viestissään Nelsonille ehdotuksen, että nykyinen prototyyppi riittäisi korvaukseksi maailmanlaajuisen lisenssin myöntöön :

> BF: Toivon, että tämän nykyisen version käyttöönsaaminen olisi tarpeeksi korvausta sinulle antaaksesi meille maailmanlaajuisen lisenssin.

Nelson totesi arvostavansa tasoa, jolla Gzz toteutti hänen ideoitaan. Nelson kertoi työskentelevänsä nyt toisen implementaation parissa ja seuraavansa Gzz-projektin kehitystä vain kaukaa. Nelson toivoi sovitteluratkaisun kiistaan löytyvän ja tapaavansa mielellään ryhmän jäseniä hypertekstikonferenssissa Nottinghamissa elokuussa 2003:

> TN: Kunnioitan suuresti teidän työtänne ja pidän teistä kaikista. Ja toivon että molempia osapuolia tyydyttävä ratkaisu löytyisi.

10.helmikuuta 2003 pääkehittäjä Lukka totesi sähköpostiviestissään[86] patentti-ongelman olevan todellinen, ja siksi oli toimittava välittömästi sen ratkaisemiseksi. Lukka esitti ryhmän jäsenille kaksi vaihtoehtoista ratkaisutapaa Gzz-tiedostorakenteen kehittämiseen jatkossa. Vaihtoehdot olivat kieltää Gzz:n jakelu Yhdysvalloissa tai ryhmä vaihtaisi kokonaan toisenlaiseen tietorakenteeseen ohjelmistonkehitystyössään.

8.2 ”Ei enää yhteistyötä Nelsonin kanssa”

13. helmikuuta ryhmä päätti vaihtaa uuden tietorakenteen Zigzag-tiedostorakenteen tilalle. Nelsonin kanssa olisi mahdollisesti päästy sopimukseen pitkien neuvottelun jälkeen, mutta ryhmä halusi välittömästi irtisanoutua yhteistyöstä. Ryhmän jäsenet olivat keskenään solidaarisesti sitoutuneet vapaaseen toimintamalliin ja lisäksi ryhmän jäsenet kokivat ettei Nelson ollut luotettava yhteistyökumppani. Ryhmä halusi sitoutua ilman ehtoja vapaan ohjelmiston kehittämiseen. Projektia rahoittavat yritykset hyväksyivät projektin käyttämät vapaat ohjelmistolisenssit, joten ryhmän oli helppo päättää jatkamisestaan vapaana projektina. Tuukka kommentoi ryhmän ja Nelsonin välistä kiistaan ratkaisua. Tilanteen kärjistyessä ryhmä halusi purkaa kaikki siteensä Nelsoniin:

> Sitä just puhuttiin että sen olisi voinut vielä muutama viikko sitten neuvotella että jos Ted olisi vaan antanut Gzz:lle lisenssin, niin kuin vapaan lisenssin…Sähköpostia oli vaihdettu ja just niitä ehtoja, jos vaan päästäisiin yhteisymmärrykseen niin voitaisiin jatkaa. Sitten Benja varmaan ensimmäisenä alkoi katsomaan mitä vaihtoehtoja sitten olisi ja sitten jotenkin tuohon RDF:ään alettiin päätyä. Niin me ollaan periaatteessa pistetty välit poikki. Silleen että me ei tehdä sen rakenteen kanssa enää mitään. Ja aikaisemmin jo muilta osin. Tedillä on jotain muita toteutuksia työn alla.
> (Tuukan haastattelu 3.3.2003)

[86] http://lists.nongnu.org/archive/html/gzz-dev/2003-02/msg00043.html , tarkistettu 9.11.2005.

Projekti päätettiin nimetä uudestaan ja ryhmän eri ohjelmistoideat oli tarkoitus jakaa sen alle alaprojekteihin.

Ryhmän Irkki-kanavalla keskusteltiin 13. helmikuuta 2003, että yhteistyö Nelsonin kanssa oli tullut päätökseen[87]. Lukka ja Antti kertoivat, että ongelmat olivat olleet henkisesti raskaita. Pääkehittäjä Lukalla oli helpottunut olo, kun yhteistyö Nelsonin kanssa loppui:

tuomasl: tuntien tedin taustan minä olen ainakin henkisesti valmistautunut siihen jo kauan
tuomasl: ja zz-rakenne on enää häviävän pieni osa gzz:aa
ibid: nyt jo varmaan uskaltaa puhua asioista :-) mulla oli silloin ns. paha hiki eli sellainen ahdistava väsymys
tuomasl: minulla kanssa; siitä minun masennus lähti
tuomasl: mut viime vuosi paransi tilannetta paljon
ibid: jep
tuomasl: ja nyt on jo hyvä olo
ibid: samoin ;-)
tuomasl: oikeastaan mitä enemmän ted on eronnut tästä projektista, sitä parempi olo minulle on tullut...
tuomasl: koska olin luvannut tedille kontrollin ja siksi en itse uskaltanut tehdä monia päätöksiä
tuomasl: mutta ted ei kuitenkaan ollut tarpeeksi mukana tosissaan päättääkseen mitään
tuomasl: se minua tässä nappaa että se nyt jaksaa väittää menettäneensä kaksi vuotta; yhteensä se teki niitten aikana meidän kanssa hommia korkeintaan 2kk
an-toni: taisi viitata odottamiseen. mutta epäreilua kyllä
tuomasl: ei odottamiseenkaan; sillä oli samaan aikaan vaikka mitä projekteja päällä

Lukka kirjoitti ehdotuksen[88], joka käsitteli Gzz:n lohkoamista ja uudelleen nimeämistä. 20. helmikuuta pääkehittäjä Lukka kirjoitti viestin, miten ehdotus oli ryhmän keskustelujen perusteella ratkaistu. Lukka perusteli syyt ja kertoi miten projektia jatketaan vapaana projektina. Lukan mukaan Zigzag-tiedostorakenteen käyttäminen jatkossa olisi patentoidun teknologian käyttämistä, joka ei olisi avoimesti lisensoitavissa. Patentoidun teknologian käyttö oli vastoin ryhmän avointen ohjelmistojen filosofiaa. Lukan mukaan idea xanalogisesta hypertekstistä ei ollut patentoitu. Se oli ainoastaan liikesalaisuuden suojassa ja koska teknologia oli yhteistyön aikana avoimesti paljastettu, sen käyttöä ei voisi rajoittaa. Lukan mukaan patentoitavuuteen liittyvien uhkien vuoksi ryhmän ohjelmistoinnovaatioiden ideat jaettiin jatkossa alaprojekteihin. Jokainen alaprojekti sai oman vastaavansa ja projektien jako tehtiin kehitettävien erilaisten innovaatioiden jakautuessa alaprojekteihin.

Lukka kertoi muutoksesta seuraavaa:

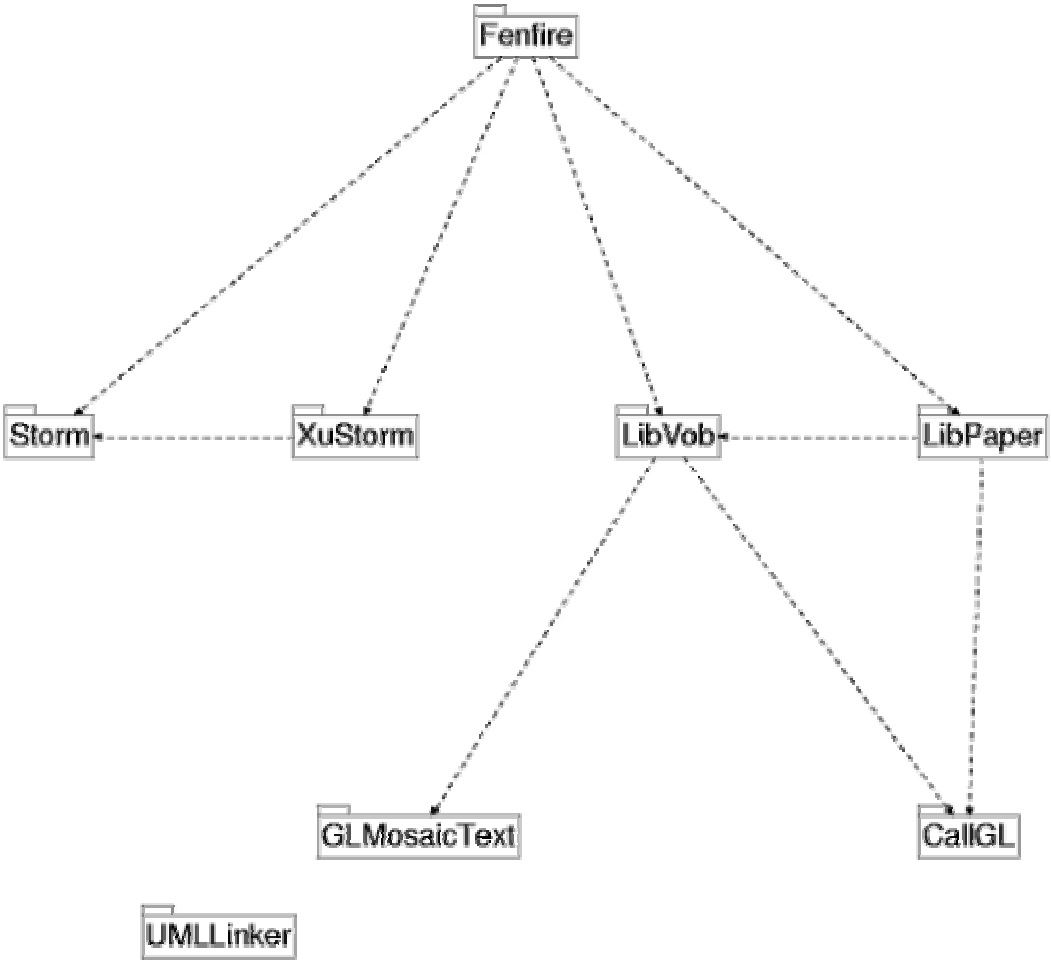
---

[87] Irc- kutsumanimet keskustelussa ovat Tuomasl on Tuomas, Ibid on Antti ja An-toni on Toni.

[88] Kyseinen Pegboard on verkossa, http://himalia.it.jyu.fi/ffdoc/fenfire/pegboard/pegboard.gen.uml.html#pegboards, tarkistettu 7.11.2005. Kyseisen Pegin ratkaisun voi lukea Pegboardilta tai suoraan Urlista : http://himalia.it.jyu.fi/ffdoc/fenfire/pegboard/projectsplit--tjl/peg.gen.html, tarkistettu 7.11.2005.

> Gzz projekti päättyy, vaikkei se ole akronyymi ja muotisana (BuZZword). Gzz on tuonut zz- rakenteen liian helposti mieleen. Koodi josta Gzz on koostunut jaetaan uusiin projekteihin kerroksien tapaan, jotkut projektit ovat toisista vähän riippuvaisia, kuitenkin ollen mahdollisimman itsenäisiä.

Ryhmän keskusteluissa uudeksi vapaaksi tietorakenteeksi ryhmän ohjelmisto-innovaatioille valikoitui RDF (Resource Description Framework), joka on World Wide Web konsortiumin avoin standardi[89]. Kevään aikana nimettiin kokonaisprojekti Fenfire[90], jonka alle nimettiin lopulta 10 alaprojektia ohjelmistoinnovaatioille. Ryhmälle alkoi uusi työ kirjoittaa ohjelmistokoodia osittain uusiksi tietorakenteen vaihtamisen vuoksi.



Kuva 2: Fenfire-projektin suunnitelma jakaa innovaatiot pääprojektin alle omiksi alaprojekteiksi

Kenttähavainnoinnin aikana 3.3.2003 keskustelin vapaamuotoisesti ryhmän viiden ohjelmoijan kanssa ryhmän tilanteesta. Ohjelmoijat olivat kehittäneet lisää uusia ideoita,

---

[89] World Wide Web konsortiumi kehittää Internetiin liittyviä standardeja, yksi konsortiumin perustajista on Internetin keksijä Tim Berners-Lee.

[90] Ryhmä nimesi projektille uuden Cvs-tietovaraston 22.2.2003. http://savannah.nongnu.org/projects/fenfire/ , tarkistettu 7.11.2005. Ryhmä myös perustaa projektilleen uuden kotisivun , http://fenfire.org/ , tarkistettu 7.11.2005.

rakentaa mm. legopalikoista uusia ohjaimia tietokoneelle. Arvioinnissa olleiden käsikirjoitusten palaute käyttäjätestien puutteesta oli huomioitu aloittamalla käyttäjätestejä prototyyppiohjelmistolla. Tietokoneen käytettävyyteen liittyvien asioiden kehittelyyn oli sidottu yhteistyösuhteita muiden Jyväskylän yliopiston tietokoneen käytettävyyttä tutkivien osapuolten kanssa[91].

27. maaliskuuta 2003 havainnoin ryhmän esiintymistä Jyväskylän yliopiston Agorassa InBct-tutkimusohjelman seminaarissa. Seminaarin aluksi pääkehittäjä Lukka kertoi haastavasta uudesta tilanteesta ryhmän ja Nelsonin yhteistyön kariuduttua. Hän kertoi oman versionsa yhteistyön loppumisesta ja siitä johtuvista muutoksista projektissa:

> Eli ensimmäinen iso muutos on se, että Ted Nelson päätti lopettaa yhteistyön meidän kanssa ja päätti samalla myös, että ei suvaitse meidän käyttää hänen patentoimaansa rakennetta vapaassa softassa. Mistä johtuu että meillä on mennyt tästä keväästä aika iso aika siihen että on mietitty, mitä me ruvetaan käyttämään sen tilalla. Meillä on kuitenkin, oltiin siinä mielessä onnekkaita että löydettiin erittäin hyvä muu rakenne siihen tilalle eli RDF. RDF, joka on siis ton WWW konsortion kehittämä kieli struktuurien kuvauksille.

Tilaisuudessa oli paikalla opiskelijoita, tutkijoita ja rahoittajien edustajia. Ryhmän ohjelmoijat esittelivät demoina ryhmän ohjelmistoteknologiaa ja sen teknologisia osa-innovaatioita.

Lukka viestissään[92] 31. maaliskuuta 2003 ilmoitti ryhmän saaneen negatiivista palautetta projektin Pr:stä ja organisaatiosta. Tämä johti siihen, että ryhmän Gzz-dev listalla käytiin keskustelua, että mitä asialle voitaisiin tehdä. Tämän seurauksena ryhmä päätti perustaa tietovarastoonsa tiedoston journaleista, joka tarkoitti että jokainen kehittäjä omassa journalissaan ilmoittaisi tekemästään työstä ja mitä tekisi seuraavien viikkojen aikana. Lisäksi ryhmä päätti laittaa kotisivustoilleen visuaalisia animaatioita teknologisten innovaatioiden toimivuudesta ja ryhmä päätti kirjoittaa ”milestones” tiedoston kotisivustolleen ryhmän historiallisesta kehityskaaresta. Jatkossa oli tarkoitus panostaa projektin markkinointiin ulospäin.

---

[91] Yhteistyötahoina mm. kognitiotieteen professori Pertti Saariluoma ja tutkija Marketta Niemelä. Pertti Saariluoma julkaisi vuonna 2004 kirjan Käyttäjäpsykologia ; ihmisen ja koneen vuorovaikutuksen uusi ajattelutapa.

[92] http://lists.gnu.org/archive/html/gzz-dev/2003-03/msg00282.html, tarkistettu 20.6.2006.

8.3 Rahoitushakemukseen ja tieteellisen julkaisutoimintaan panostamista

23. toukokuuta 2003 ryhmän ohjelmoijat kertoivat haastattelujen ja kenttähavainnoinnin yhteydessä projektin luonteen muutoksesta. Tuukka kertoi etukäteen sähköpostiviestissä, että ryhmä teki nyt pääasiassa tutkimusta eikä työskennellyt ohjelmistokoodin kimpussa.

> Nyt onkin sillä lailla epätavallista, että teemme artikkeleita ja seminaariesitelmiä. Tai kevään ajan se on tainnut olla enemmän sääntö kuin poikkeus. (Tuukan sähköposti)

Ryhmä teki toukokuun lopussa rahoitushakemuksen Suomen Akatemialle tutkiakseen ja kehittääkseen ohjelmistoinnovaatioitaan ”Theoretical issues on Fenfire”[93]. Dokumentti kertoi ryhmän tutkimusideoista ja niiden teknologisista yksityiskohdista ja yhteistyöverkostosta. Yhteistyöosapuoliin kuuluvat Jyväskylän yliopiston yhteistyötahojen lisäksi Comas-tutkijakoulu, Tekes, Sonera ja Nokia. Tutkimusympäristönä ja työvälineinä mainittiin tietokoneet, 3D-kortit tietokoneissa ja vapaat ohjelmistokehitystyökalut.

Haastatteluissa nousi esille kolme teemaa, jotka olivat ryhmän työlle keskeisiä: sitoutuminen avoimen ohjelmistonkehityksen työmuotoihin uuden projektin, Fenfiren, alla, tutkimustyöhön panostaminen sekä käytettävyystestausten aloittaminen. Avoimuuden tärkeys perustavana arvona tuli esille monen ohjelmoijan haastattelussa. Julkisuus ja tiedon avoimuus oli merkittävässä asemassa ohjelmoijien kommenteissa. Avoimuus ja julkisuus oli tärkeää ryhmän tieteellisen tutkimuksen arvoille:

> Kyllä se minusta on sillä tavalla niin kuin hyvä asia, että innovaatiota ja hyvää lähdekoodia voidaan jakaa. Ja se on just niin kuin semmoinen reilun pelin henki.
> (Hermannin haastattelu 23.5.2003)

> Tuntuu kivalta, että me…tehdään avointa ja julkista tutkimusta. Et vaikka välillä puhutaan muka jostain salaisesta jutusta, niin silti ne ovat saatavilla. Periaatteessa kenen tahansa saatavilla CVS:ssä…kaikki on ollut julkista. Ja sitten mitä se tarkoittaa tämän jälkeen on, että kaikkea mitä täällä on tehnyt niin saa käyttää myös tämän jälkeen ja saa puhua tämän ulkopuolella niistä. (Askon haastattelu 23.5.2003)

Patentitkaan eivät rajoittaisi ryhmän ideoiden sovelluskäyttöä:

> Että jos jollekin asialle haetaan patenttia, niin se ei saa rajoittaa sen... sen idean tai... tai ajatuksen käyttöä siinä vapaassa koodissa. Eli kenen tahansa pitää saada käyttää sitä vapaata koodia, jossa käytetään sitä patentoitua tekniikkaa tai muuta. (Jannen haastattelu 23.5.2003)

[93] Ryhmältä saatu dokumentti tutkimussuunnitelmasta haastattelujen yhteydessä 23.5.2003.

Ryhmä oli keskittynyt keväällä tutkimustyöhön ja julkaisujen tekemiseen. Vuosien tutkimustyön tuloksista kirjoitettiin nyt tieteellisiä julkaisuja:

> Mutta nyt tässä on kovat yritykset saada valmiita sovelluksiakin tehtyä. Tai sitten…eikä silleen tämmöisessä yliopistotutkimuksessa ole edes välttämättä saada valmiita tuotteita, mutta silti jää vähän kivempi mieli, jos jotain konkreettistakin eikä vaan aina uusia ideoita, joita sitten ei voi hyödyntää.... Yritetään tehdä tieteellistä työtä enemmän mitä ennen. Tosiaan artikkeleihin ollaan panostettu tosi paljon, että sitten tavallaan on tuntunut vähän, että ei olla saatu niin sen takiakaan tehtyä mitään silleen valmista nyt koska ollaan vaan kirjoitettu artikkeleita niistä tai... jota on tutkittu tai tehty vuosia... tai täällä on tehty vuosia. (Askon haastattelu 23.5.2003).

Tieteellisille artikkeleille käytettävyydestä ohjelmiston testaaminen koehenkilöillä oli tärkeätä tieteellisen evidenssin osoittamiseksi:

> Nykyisin ihan joka paikassa tutkijat käyttävät käytettävyystestauksia. Eli tutkimus on kokeellista tutkimusta. Ja sitä ollaan nyt niinkö kovasti käynnistelemässä. Ollaan solmittu suhteita muihin yliopistolla oleviin.(Askon haastattelu 23.5.2003)

Käyttöliittymätekniikoita ryhmä pyrki tutkimaan myös teoreettisesti ja aiheesta oli tehty Suomen akatemialle rahoitushakemus:

> Kaikkiin projekti osiin jossakin mielessä perustutkimus. Esimerkiksi ne uudet käyttöliittymätekniikat, niin vaikka ne lähtee aikalailla käytännöllisistä lähtökohdista. Niin sieltä pohjalta löytyy niitä jotain teoreettisia kysymyksiä, joilla sitten tutkitaan myös eteenpäin ihan perustutkimuksen lähtökohdasta. Ja sitten sitä kautta voidaan myös... aiheutuu vastauksia johonkin kysymykseen, josta sitten taas on hyötyä projektille niissä käytännön asioissa.(Jannen haastattelu 23.5.2003)

Ryhmän väitöskirjan tekijä, Janne, näki perustutkimuksen ja soveltavan tutkimuksen rajan ryhmän työssä liittyvän ohjelmistokoodin tehtävään kehitystyöhön. Käyttöliittymätekniikoissa oli havaittu ryhmän tutkimuksissa vahva sidos kognitiiviseen psykologiaan ja havaitsemisen psykologiseen mallintamiseen :

> No voisi oikeastaan sanoa, että soveltavaa on kaikki se koodin kirjoittaminen ja perustutkimusta sitten hyvin paljon se julkaiseminen…Mihin tämä perustuu tämä kaikki? Että voisiko sieltä löytyä pohjalta jotain yleisempää asiaa. Niin kuin esimerkiksi, että jossakin vaiheessa huomattiin, että itse asiassa kaikki nuo käyttöliittymäideat perustu vahvasti siihen havaitsemisen tai näkemisen mallintamiseen.
> (Jannen haastattelu 23.5.2003)

8.4 Pääkehittäjä Lukka etsii yhteistyökumppaneita Free Software / Open Source yhteistyöhön

Vuoden 2003 Inbct-projektin esittelyposterissa ryhmä kertoi tekevänsä tutkimusta vapaana ohjelmistoprojektina. Englanninkielisen version esittelytekstissä ryhmä nimesi

tutkimustyönsä Open Source projektiksi. Myöhemmin[94] kysyn sähköpostitse pääkehittäjä Lukalta, miksi ryhmä käytti sekä Free Software ja Open Source termiä eri yhteyksissä ohjelmistoprojektistaan. Lukka vastasi ryhmän käyttävän molempia termejä yleisön ja kontekstin mukaan:

> JL: Mietinkin oletteko ryhmässänne keskustelleet mikä kuvaa projektianne parhaiten, jotta käyttäisitte yhteneväistä kuvausta projektista erilaisissa yhteyksissä?
>
> TL: Kumpikin, eri yhteyksissä. Asia riippuu mm. yleisöstä, siitä mikä sana on todennäköisemmin tuttu jne.

Elokuussa ryhmä vihdoin sai ensimmäisen sovelluksen toimimaan. 18. elokuuta viestissään[95] ryhmän saatua ensimmäinen sovellus testikäyttöön Lukka vertasi saavutusta teknologisesti Vannevar Bushin Memexiin:

> Ajattelen että kukaan tietokoneitten historiassa ei ole ollut lähempänä Memexiä kuin me olemme nyt. On fantastista nähdä, että emme ole työskennelleet turhaan.

27. elokuuta pääkehittäjä Lukka esitti ohjelmistollaan demon yleisölle suuressa ohjelmistoalan tapahtumassa. Lukan visuaalinen demo tilaisuudessa pyrki osoittamaan käyttöarvolupauksen siitä mahdollisesti kiinnostuville yrityksille ja uusille yhteistyö-tahoille. Lukka pyrki markkinoimaan tilaisuudessa ohjelmistoa toimivan visuaalisen demon avulla. Esityksen aikana paljastui pieni ohjelmistovirhe, mutta Lukka pystyi esittelemään ohjelmiston toimivuutta suurilukuiselle yleisölle. Hän kertoi ensin lyhyesti riitaantumisesta Nelsonin kanssa[96] ja uudesta haastavasta tilanteesta projektille:

> Ja perustavoitteet ja projektin niin kuin pääsisältö on säilyneet samana ihan alusta asti, eli vuodesta 1999. Tämä Fenfire tarkoittaa aarnitulta. Eli siis tavallaan jossain suolla tai metsässä palava kummallinen valo, jota ei koskaan saa kiinni. Ja mä nyt vähän myöhemmin selitän, että miksi se on ihan järkevää... niin kuin nimi tälle idealle.

Lukan mukaan ryhmän uusi nimi kuvaa vertauskuvallisesti, että pikkuhiljaa saadaan aikaan muutoksia nykyisiin tietokoneen käyttöliittymiin, ja siksi projektin loppupäivää ei ollut tiedossa:

> Mainitsin siitä, että Fenfire on se kaukana vilkkuva tuli, jota voi yrittää tavoittaa, mutta jota koskaan ei ihan saavuta, mutta.... No idea onkin se että meillä... nyt me saatetaan joskus tavoittaa se tai sitten siihen voi... voi mennä todella pitkä aika. Mutta idea on se, että siinä matkalla me kuitenkin saadaan konkreettisia asioita aikaan.

Lukka kertoi innoissaan, että ensimmäinen sovellus, FenPdf, oli saatu ryhmän sisäiseen testikäyttöön:

---

[94] Henkilökohtaiset sähköpostiviestit pääkehittäjä Tuomas Lukan kanssa 27.10.2003.
[95] http://lists.gnu.org/archive/html/fenfire-dev/2003-08/msg00034.html, tarkistettu 10.11.2005.
[96] Haastattelulainaus luvun alussa It Expo – tapahtuman luennolta Helsingin kaapelitehtaalta 27.8.2003. Lukan luennon otsikko oli ” Fenfire eli miten tietokoneiden pitäisi mielestäni toimia”.

> Joka on tällä hetkellä meidän ryhmän sisäisessä... sisäisesti tuotantokäytössä. Se on ollut meille tosi iso askel.

Lukka kertoi esitelmänsä lopussa yleisölle, että kiinnostuneet olivat tervetulleita mukaan vapaaseen projektiin testaamaan ensimmäistä koekäytössä olevaa sovellusta:

> Eli ottakaa yhteyttä. Ja tosiaan näin lopuksi sanon että me ollaan tultu tässä todella mielenkiintoiseen vaiheeseen tässä projektissa, saatu se omaan käyttöömme. Mä olen itse erittäin innostunut tästä asiasta, että se on nyt niin kuin lopulta niin monen vuoden jälkeen, kun on oikeasti käytettävä systeemi itsellä. Ja tämä projekti tosiaan jatkuu.

Luennon jälkeen yleisö esitti kysymyksiä, joista ensimmäinen koski projektin rahoitusta. Lukka vastasi projektin jatkuvan, vaikka sen rahoitus joskus loppuisikin:

> No se menee aina sillain, että tämänhetkisellä rahoituksella on loppupäivämäärä, mutta Fenfirella itsellään ei. Että vaikka rahoitus loppuisi, niin mä todennäköisesti jatkaisin sitä vapaa-ajallani sitten.

Toinen kysymys koski miten vapaassa projektissa suhtaudutaan patentointiin. Lukka vastasi, että mahdollisimman suuri osa heidän työstään on avointa:

> Tossa on monta eri aspektia. Että ensimmäinen on se, että me haluamme että asioita on vapaasti käytettävissä. Joistakin asioista me olemme tehneet keksintöilmoituksia rahoittajille. Mut me on pyritty tekemään sellasia diilejä heidän kanssaan, että GPL-lisenssin alla nämä olis sitten vapaasti käytettävissä edelleen. Eli se tarkottais että kaupallinen softa ei pystyisi välttämättä silloin hyödyntämään sitä, mutta niinkuin vapaa softa kyllä. Mutta mikään mitä tossa näytettiin ei ole patentoitu. Ja jos joku haluaa niinkuin tässä nyt, tän kans jotakin tehdä niin meidän kanssa voi ihan keskustella. Me todella selkeästi autetaan niin kuin että mitkä osat on vapaasti käytettäviä ja mitkä ei. Me pyritään siihen, että vapaasti käytettävissä on niin kuin mahdollisimman paljon. Etenkin vapaitten ohjelmien yhteydessä. Sitten kaupalliseen softaan imettävät asiat saattaa olla vähän pienempi osa meidän tuotoksista.

Päivää myöhemmin Lukka esitteli ryhmän tilannetta InBct-hankkeen rahoittajatahoille Jyväskylässä[97]. Lukka kertoi että ensimmäisen innovaation koekäyttö oli aloitettu ja että tulos Nelsonin kanssa koettujen ongelmien jälkeen oli ryhmälle positiivinen. Nokia oli pyytänyt käyttöliittymien testaamista pienillä näytöillä. Lopuksi Lukka kertoi että lisenssiongelma viivytti ja haittasi valitettavasti yritysyhteistyötä, mutta enemmän yritysyhteistyötä kaivattaisiin jatkossa. Yritysten kanssa haluttaisiin yhdessä selvittää olisiko joistain ideoista syytä tehdä keksintöilmoituksia. Nämä olivat keskeisiä perusteluja sille, että Fenfire jatkaisi seuraavana vuonna InBct-hankkeessa. Syksyn aikana ryhmä teki jatkosuunnitelmaa jatkaakseen osana InBct-tutkimusprojektia vuonna 2004.

---

[97] Luettelen tässä esityksen pääkohdat Lukan esityskalvojen perusteella.

8.5 Uusi patenttiongelma-projekti vaarassa?

Syksyllä projekti jatkui normaaleissa merkeissä. 18. marraskuuta pääkehittäjä Lukka ilmoitti ryhmän jäsenille[98] viestissään yllättäen uudesta suuresta patenttiongelmasta. Lukka toivoi, että ratkaisu patenttiongelmaan löydettäisiin ryhmän jäsenten kesken yhteistyössä:

> Anteeksi, että olen ollut pitkään vaitonainen, olen ollut alamaissa tästä johtuen…
> On yhdysvaltalainen patentti numero 6,031,537 , joka on myös anottu Eurooppaan (ei myönnetty vielä, nähtävästikään), joka kattaa kaikki käyttäjän käyttöliittymät, joita olemme työstäneet (fokus + konteksti graafiset navigaatiot). Valitettavasti, siellä on huomattava olemassa oleva tekniikka (prior art). Jotkut patentin väitteistä ovat naurettavan laveita. Olen keskustellut patenttilakimiehen kanssa tästä ja on ehkä mahdollista välttää seuraukset ohjelmistollemme. Meidän tulisi valmistaa tiivistelmä olemassa olevasta tekniikasta ja mitä väitteitä se koskettaa…tämän keskustelun jälkeen (kun olen tavannut patenttilakimiehen ja keskustellut hänen kanssaan) toivon, että on minulla selvempi kuva tilanteesta. Julkisuus asiasta tässä tilanteessa, kun meillä ei ole kaikkia faktoja selvillä, ei olisi hyvästä… Mutta tee kaikkesi estääksesi ohjelmistopatentit Euroopassa…Se on silti mahdollista.

Tieto patenttiongelmasta tulisi pitää vain ryhmän sisällä ennen kuin tiedettäisiin millaiseen ratkaisuun ongelman suhteen päädyttäisiin. Siksi vuoden lopulla tästä patenttiongelmasta ei puhuttu julkisesti. Viestilistoilla keskusteltiin vain normaaleista teknisistä ongelmista. Fenfire siirsi toimintansa myös pois CVS:stä, toisen versionhallintaohjelman Darcsin alle. Lukka kirjoitti tutkimussuunnitelmaa uutta rahanhakua varten. Tutkimussuunnitelman nimi oli ”*Paikkariippumaton Web*” ja siinä haettiin rahoitusta vuosille 2004–2006. Rahoitusta hakemukselle ei kuitenkaan tullut.

Vuoden 2004 alussa projektin sähköpostilistoilla ei ollut enää pääkehittäjän viestejä. Ratkaisua patenttiongelmaan ei ollut löytynyt. Projekti ei jatkanut enää vuonna 2004 InBct-ohjelmassa. Maaliskuussa 2004 kävin tapaamassa ryhmän ohjelmoijia. Ryhmässä työskenteli enää pääasiassa kolme henkilöä: Benja, Matti ja Janne. Keskustelin saksalaisen Benjan kanssa, joka oli valittu ryhmän uudeksi pääkehittäjäksi. Hän kertoi, että ryhmä oli saanut muutaman kuukauden projektirahoituksen Soneralta. Rahoituksella työllistettiin Benjan lisäksi toinen ohjelmoija Matti, lisäksi tutkija Janne viimeisteli väitöskirjaansa vuoden loppuun tutkijakoulun rahoituksella. Ryhmän pääkehittäjä Lukka oli henkilökohtaisista syistä lopettanut projektissa vuoden vaihteessa ja siirtynyt työskentelemään suomalaiseen kolmiulotteista tietokonegrafiikkaa kehittävään

[98] Luottamuksella ryhmän ohjelmoijalta saatu alkuperäinen sähköpostiviesti, jonka Lukka lähetti Fenfire-ryhmän jäsenille 18.11.2003.

yritykseen. Julkisesti ryhmän tilanteesta Internetissä[99] kertoi huhtikuussa 2004 ryhmän uusi pääkehittäjä Benja (11.4.2004):

> Fenfire on aina ollut Free Software projekti, mutta nykyään meidän käyttämä käyttöliittymä koodi (Loom, varsinaisesti) ei ole julkinen ja eikä ole ollut sitä viime syksystä johtuen patenttiongelmista. Enkä tiedä milloin tähän tulee ratkaisu. Itse asiassa, on olemassa älyttömän laaja patentti, joka kuten me kehittäjät uskomme että, on laaja olemassa oleva teknologia. Mutta meidän tarvitsee tietää mitä tapahtuu jos joku rupeaa oikeustoimiin raivoissaan ja päättää muuta ja miten me voisimme olla persoonallisesti vastuussa…

Ryhmän tutkimuksellisia ansioita arvostettiin Jyväskylän yliopistolla. Ryhmän kolme jäsentä palkittiin Schildtin rahaston stipendillä toukokuussa 2004. Tiedote kertoo urauurtavasta tutkimustyöstä otsikolla ”Toimiva työympäristö vapaana ohjelmisto-projektina”:

> Erikoisstipendi Tuomas Lukan tutkimusryhmälle. Schildtin rahasto jakoi 7000 euron arvoisen stipendin nuorten tutkijoiden ryhmälle Benja, Janne ja Matti.
> Benja (s.1982), Janne (s.1977) ja Matti (s.1981) jatkavat Agora Centerissä Tuomas Lukan johdolla aloitettua projektia, joka tekee uraauurtavaa työtä uusien tiedonhallintamenetelmien rakentamiseksi. Ryhmä rakentaa toimivaa työympäristöä, jossa toisiinsa liittyvät järjestelmät saadaan liitettyä toisiinsa sujuvasti. Se etsii hyperrakenteen ja joustavien käyttöliittymien yhdistelmää ratkaisuksi informaatio-tulvaan. Tavoitteena on, että järjestelmässä erilaiset tiedot ovat yhdisteltävissä ja versionhallintakin tukee tilapäistäkin ryhmätyöskentelyä. Kyseessä on vapaa ohjelmisto-projekti, jonka tuotoksena kasvaa koko ajan vapaa Fenfire-prototyyppi.
> (Suomen Kulttuurirahaston Keski-Suomen rahaston tiedote 28.4.2004)

Joulukuussa 2004 ryhmän tutkija Janne väitteli kiitettävällä arvosanalla tieteellisestä laskennasta. Monitieteellinen tutkimus oli tehty Fenfire-projektissa. Tammikuussa 2005 ansiokas tutkimus palkittiin Jyväskylän teknologiakeskuksen Vuoden 2004 Agora-väitöskirjapalkinnolla. Varatoimitusjohtaja Jukka Akselin Jyväskylän Teknologiakeskus Oy:stä totesi palkinnon perusteluissa seuraavasti:

> Janne Kujalan väitöskirja tavoittelee vahvojen tieteellisten ansioiden lisäksi erinomaisesti Jyväskylän aseman vahvistamista mallintamisessa. Jyväskylän Teknologiakeskus haluaa palkita tieteiden välisellä alueella valmistuneen ansioituneen väitöskirjatyön osoituksena tärkeästä tutkimustyöstä, jota Jyväskylän yliopistossa tehdään. Tiivis yhteistyö yliopiston kanssa alueen elinkeinoelämän edistämisessä on meille erittäin tärkeää.

Näiden perusteiden saattelemana Jyväskylän Teknologiakeskus myönsi Jannelle 2000 euron suuruisen tutkimusapurahan hyvin alkaneen ansiokkaan tieteellisen toiminnan tukemiseksi.

---

[99] Benjan tilannekatsaus löytyy Danny Ayersin blogista. http://dannyayers.com/2004/04/12/fenfire/, tarkistettu 3.4.2006.

Nelsonin ja ryhmän riitaantumista ei juuri kukaan ryhmän jäsenistä kommentoinut julkisesti. Ryhmän tutkimustyön ”testamentiksi” muodostui Lukan ja Fallensteinin käsikirjoitus *”Hyperstructure: Computers built around things that you care about”* . Artikkeli kertoi ryhmän ideoitten sovellettavuudesta myös Semanttisen Webin kaltaiseen teknologiaan[100]. Syyskuussa 2004 ryhmässä vuosina 2000–2002 toiminut Antti kertoi oman versionsa ryhmän tarinasta Internet sivustolla[101]. Antti oli työskennellyt projektin jälkeen ohjelmistotekniikan laitoksella ja seurannut läheltä ryhmän vaiheita. Hän kertoi lyhyesti oman versionsa ryhmän tarinasta:

> Se on surullinen tarina, se on raivostuttava tarina.
> Tapasin Ted Nelsonin ensimmäistä kertaa helmikuussa 2000. Tuomas Lukka oli kutsunut hänet vierailemaan yliopistollamme ja menin kuuntelemaan hänen luentoaan. Se oli korkealentoista, enkä ymmärtänyt kaikkea…Hän esitti varhaisen ZigZag-demon genealogia tietokannasta, ja yhden asian jonka muistan oli hänen epäluottamuksensa ” *nörtit tuottavat lahjoja*[102]” – hänen mielestään tekniikkaihmiset tekivät ohjelmiston täysin väärin (hän ei puhunut sisäisestä organisaatiosta vaan käyttäjän kokemuksesta). Seuraavana kesänä liityin Tuomas Lukan tutkimusprojektiin tutkimusavustajana. Tarkoitus oli valmistaa Zigzag Javalla…Alussa se oli mukavaa, ja meillä oli paljon hauskaa Tedin kanssa, joka oli projektin ohjaaja (kuvittele elokuvaohjaajaa)… Gzigzag-projekti ei materialisoinut Tedin unelmasysteemiä niin nopeasti kuin hän olisi halunnut, niinpä hän veti tukensa pois. Ensin hän ei osallistunut niin paljon enää keskusteluihin kuin ennen oli osallistunut. Jonkin aikaa myöhemmin hän kielsi tavaramerkkinsä käytön (hän oli aikaisemmin antanut luvan käyttää sitä), ja muutimme nimen Gzz. Saman aikaan olin itse loppuun palamassa, osittain Tedin epäluottamuksen takia ja suurin syy oli projektin sisäiset ongelmat. Jäin pois projektista vuoden 2002 lopulla…Tämän jälkeen Ted toi patentin esille. Projekti oli pakotettu luopumaan kaikesta Zigzagiin liittyvästä, se muodostui uudelleen pelastakseen innovaatiot jotka oli liitetty Zigzagiin ja uudeksi kuoreksi tuli RDF…Riittää kun sanon että Tuomas myös loppuun paloi. Ihastelen Ted Nelsonia hänen nerokkuudestaan, mutta halveksin hänen toimiaan tässä tapauksessa. En luota hänen enää koskaan projektikumppanina, vaikka myönnänkin Zigzagin cooliuden.

Luvun lopuksi käsittelen yhteenvedon omaisesti luvun 8 tapahtumia. Ryhmä perusti uuden Fenfire-projektin, kun epäluottamus Nelsoniin purkautui kiistana, johon ei löytynyt ratkaisua. Ryhmä halusi pysyä avoimena ja vapaana taatakseen itselleen rekursiivisen yhteisöllisyyden toimintahorisontit. Syksyllä 2003 ryhmä sai yhden ohjelmistoinnovaatioittensa ensimmäisen testiversion koekäyttöön, mutta pian sen jälkeen ilmeni patenttiongelma, johon ei löytynyt ratkaisua. Pääkehittäjä Lukka teki omat johtopäätöksensa ja lopetti työskentelynsä projektissa. Projekti jatkoi kuitenkin vielä vuonna 2004 ja sai erinäisiä tunnustuksia tutkimustyölleen Jyväskylän yliopiston toimesta.

---

[100] Artikkeli saatava verkosta http://fenfire.org/manuscripts/2004/hyperstructure/, tarkistettu 13.6.2006.
[101] http://lambda-the-ultimate.org/node/233#comment-1713, tarkistettu 3.4.2006.
[102] Geeks bearing gifts oli Nelsonin käyttämä lause.

Projektin vuonna 2003 kokema patenttiongelma koitui projektin tuhoksi. Kysymyksessä oli ns. prior art-patenttiongelma[103] suhteessa jo olemassa olevaan teknologiaan. Ongelmaan ei löydetty nopeaa ratkaisua ja sen vuoksi projektin rahoitus seuraavalle vuodelle huononi oleellisesti. Vuoden 2004 alusta Fenfire ei jatkanut Jyväskylän yliopiston Innovation in Business, Communication and Technology-hankkeessa ja ryhmän pääkehittäjä lopetti työskentelyn ryhmässä. Ryhmän kehittämää ohjelmistoa ei saatu projektin puitteissa tuotteistettua. Ohjelmiston testiversioita ei juurikaan käytetty ryhmän ulkopuolella muuten kuin ohjelmistotyön kursseilla ja vapaaseen ohjelmistoprojektiin ei liittynyt juurikaan ulkopuolisia kehittäjiä. Ryhmän työn tulokset olivat tieteellistä perustutkimusta ja tieteellistä yhteistyötä harjoitettiin usean tahon kanssa. Muita ryhmän tieteellisiä ansioita olivat muutama tieteellinen artikkeli kansainvälisissä julkaisuissa, muutamia konferenssiesityksiä sekä yksi opinnäytetyö. Ryhmässä työskennelleet ohjelmoijat saivat lisäksi arvokasta työkokemusta ohjelmointialalta. Monille työstä oli paljon apua opiskeluun ja lisäksi ohjelmoijat saivat palkkaa työstä. Ryhmä kehitti ohjelmistotyössään monta sosiaalisesti innovativiista tapaa toimia tehokkaammin verkon välityksellä.

Valmista ohjelmistotuotetta ei onnistuttu kehittämään, mutta tuotteen kehittämisen epäonnistumiseen ei ole yhtä selkeää syytä. Projektinhallinta, liian vähäinen yhteistyö rahoittavien yritysten kanssa sekä uusien jäsenten houkuttelemisessa vapaaseen ohjelmistoprojektiin epäonnistuttiin. Hypoteettinen kysymys on, että olisiko tutkimusprojekti toiminut edes niin montaa vuotta ilman sitä rahoittaneita tahoja. Yliopiston ja yritysten taloudellinen tuki monen vuoden ajan mahdollisti tämän kunnianhimoisen projektin toteuttamisen. Merkittävä osuus epäonnistumisessa oli luottamuksen purkautuminen merkittävimmän yhteistyökumppanin, Ted Nelsonin kanssa. Nelson osoittautui epäluotettavaksi yhteistyökumppaniksi ja yhteistyö hänen kanssaan loppui hänen pyydettyään lisenssimaksua. Avoimen standardin valinta kehitystyön alustaksi ei kuitenkaan auttanut projektia, vaan projekti ajautui uuteen patenttiongelmaan. Vuonna 2003 meni myös rahoittajien usko projektiin ja mihin patenttiongelma oli osasyynä.

[103] Prior art tarkoittaa patenttiongelmaa suhteessa olemassa olevan teknologiaan. Ongelmat liittyvät esimerkiksi siihen miten teknologioita voidaan vertailla keskenään, että patenttiväite pystyttäisiin toteamaan ja osoittamaan lainsäädännöllisesti.

9. POHDINTAA

Vapaa ohjelmistoprojekti pyrki pitämään toimintansa mahdollisimman avoimena rekursiivisen yhteisöllisyyden alueella. Toisaalta se oli projektille riski, koska patentti-ongelmat uhkasivat projektin vapautta. Patenttiongelmien ilmetessä ei kuitenkaan lähdetty selvittämään patenttiväitteiden kestävyyttä, vaan patentoitavuusväitteiden muodostamat uhkakuvat aiheuttivat toiminnan muutoksen. Projekti reagoi uhkakuviin kehittämällä tavan jatkaa mahdollisimman vapaana julkisen rekursiivisen alueena. Novelty ja prior art kysymykset eivät tulleet selvitetyksi patenttiuhkien yhteydessä. Rekursiivisen yhteisöllisyyden alueella on useita, erilaisia uhkia vapaalle toiminnalle regulatiivisessa kehyksessä. Sopimukset ja erilaiset toimijoiden väitteet tiedon omistamisesta ja sen vapaasta kehittämisestä ovat kamppailuja tämän rekursiivisen toiminnan luonteesta. Innovaatioprosessia on usein kuvattu rekursiiviseksi ja oppimis-keskeiseksi toimintatavaksi. Rekursiivinen innovaatiomalli on siirtynyt myös vapaiden ohjelmistoprojektien toimintamalliksi ja teknologiayritysten kanssa kilpailevaksi malliksi. Internet on täynnä erilaisia foorumeja vuorovaikutteisille rekursiivisille prosesseille.

Ohjelmistoyhteisöt muodostavat oman yhteisöllisen tapansa toimia. Open Sourcen ideaa on kritisoitu viime aikoina ja toisaalta on väitetty useimpien projektien epäonnistuvan (Fogel 2005, 9)[104]. Walker puolestaan väittää ettei ole Open Source yhteisöä[105]. Keskeisin informaatioaikakauden ongelma liittyy kuitenkin tiedon vapaaseen jakamiseen ja tiedon omistamiseen. Vapaat ohjelmistoprojektit yliopistoympäristössä ovat hankalia toteuttaa ohjelmistopatentointiin liittyvien uhkakuvien vuoksi. Yliopistoilla ei ole usein resursseja toimia kiistojen ratkaisemiseksi, esim. ohjelmistojen lisenssimaksuihin tai patenttiasioihin liittyvissä oikeudellisissa toimenpiteissä. Tapani Tarvaisen mukaan[106] yliopistolla ei ole minkäänlaista ohjeistusta vapaan ohjelmistotutkimusprojektin toiminnasta. Lukan tutkimusryhmällä ei ollut juurikaan resursseja taistella vapaan

---

[104] Karl Fogel: Producing Open Source Software, http://www.producingoss.com/producingoss.pdf, tarkistettu 20.4.2006.
[105] John Mark Walker: There is no open source community http://www.onlamp.com/pub/a/onlamp/2006/01/12/no_oss_community.html, tarkistettu 20.4.2006.
[106] Sähköposti Tapani Tarvaisen ja Juha Lemisen välillä 20.12.2005. Tapani Tarvainen on Effin vara-puheenjohtaja ja Jyväskylän yliopiston yli-insinööri. Effi ry on yhdistys, jonka tarkoituksena on toimia sananvapauden ja yksityisyyden suojaamiseksi ja edistämiseksi niin Internetissä kuin yhteis-kunnassa laajemminkin.

ohjelmistotoiminnan puolesta. Toisaalta yhteistyökumppaneina olleet teknologiayritykset taistelevat vasta toimivien ohjelmistoinnovaatioiden puolesta, ja siksi alkuvaiheissa olevien innovaativisten ohjelmistoideoiden kehittely on suojaamatonta.

9.1 Kohti rekursiivisesti yhteisöllistä toimintatapaa?

Jos ymmärrämme vapaata lähdekoodia sisältävän teknologisen innovaation tai avoimen lähdekoodin ohjelmiston myös sosiaaliseksi innovaatioksi, niin silloin sen voidaan nähdä omaavan mahdollisuus muuttaa vallitsevia käytäntöjä. Innovaatiotoimintaan kiinnittyy hegemonistisia kamppailuja, jotka ilmenevät erilaisina valtaresursseina, joiden avulla kehitetään tai estetään sellaisia innovaatioita, joihin liittyy yhteiskunnan institutionaalisiin rakenteisiin kohdistuvia muutoksia[107]. Mikrotasolla toimijoiden välisessä dynaamisessa yhteistyössä kehittämiseen pyrkivä toiminta ilmenee luottamusta osoittavien toimintatapojen välityksellä.

Esitän tutkimukseni lopuksi tulkinnan, miten rekursiivisen yhteisöllisyyden aluetta voidaan kuvata yhteiskunnan sosiaalista rakennetta vasten. Aloitan Christopher Keltyn minulle osoittamasta kritiikistä, joka koski ”Recursive Publics” käsitteen suomentamisesta termillä rekursiivinen yhteisö. Hän kommentoi seuraavasti[108]:

> Yhteisö on kömpelö korvike, koska se epäpolitisoi ”julkisen”--- yhteisö voi olla olemassa ilman, että sillä on mitään pakollista poliittista roolia. Julkinen (ja julkinen alue) ovat konstituoitu ensisijaisesti (ehkä kenties ainoastaan) asioina, jotka ovat riippumattomia organisoidusta hallinnosta. Habermasin alkuperäisen tulkinnan hengessä, julkisen alue antaa ohjeita hallinnolle, tai vastustaa hallinnon valtaa valistuksen aikakaudella tai valistuksen vallanpitäjiä. Nykyisin, joka tapauksessa (johon haluan kiinnittää huomiota) julkinen voi perustua itsessään tarkastamaan hallinnon entiteettejä kuten yrityksiä, kansainvälisiä organisaatioita tai jopa jossain määrin ”yhteisöjä” --- kuten yhtä lailla tarkastaa konventionaalista hallinnointia valtio/kansakunta tyyliin.

Oman tutkimukseni valossa vastaan Keltyn kritiikkiin teoreettisella yleistyksellä kuvaamalla vapaitten ohjelmistojen toiminnan kenttää hypoteettisella mallilla, jolla pyrin kehittämään Keltyn teoriaa. Diskursiivisen materiaalinen luonne tarkoittaa, että ohjelmisto hyödykkeenä ei ole pelkästään materiaalinen objekti, vaan että objektilla on *informaationaalisesti erilaisia materiaalisia ulottuvuuksia.* Tämä merkitsee sitä, että toiminnan kentässä tuotetaan *informaationaalisia valtasuhteita eri tavoin eri toimijoiden*

---

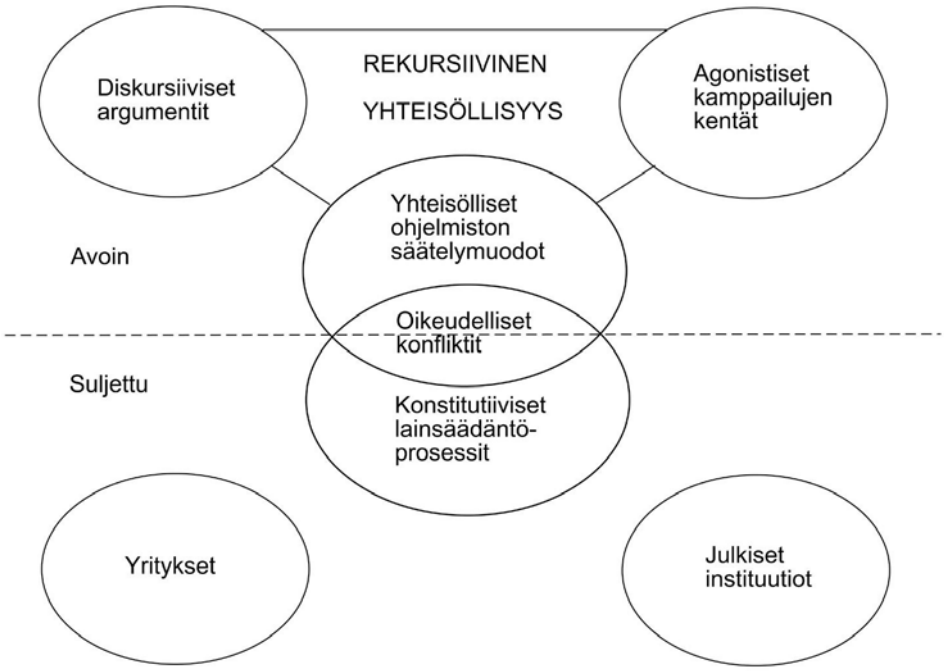
[107] Katso Heiskala 2006.
[108] Christopher Keltyn sähköposti Juha Lemiselle 31.5.2006.

välillä. Kuvaan tätä valtasuhteiden kenttää kaaviolla 6. Kaavio perustuu ajatukseen, että rekursiivisen yhteisöllisyyden agonistista kenttää voidaan kuvata diskursiivisten argumenttien, agonististen kamppailujen ja yhteisöllisten ohjelmistosäätelymuotojen välisenä toiminnan kenttänä. Tämä kenttä muodostaa rekursiivisen yhteisöllisyyden avoimien ja vapaiden toimintamallien puolesta ja sen sisällä käydään agonistisia kamppailuja, jotka ovat informaation omistamiseen ja jakamiseen materiaalisesti kytkeytyneitä. Diskursiiviset argumentit ovat yhtä lailla kielellisesti tuotettuja sekä käytännöllisiä tekoja. Kentän ulkopuolella toimivat liiketoiminta- ja julkiset instituutiot sekä konstitutiiviset lainsäädäntöorganisaatiot.

Jos diskursiivista argumenteista koostuvassa keskustelussa ei päästä rekursiivisen yhteisöllisen kentällä konsensukseen, toimijat vetoavat usein argumentteihin jotka on tuotettu rekursiivisen yhteisöllisyyden ulkopuolella. Tällaisia ovat esimerkiksi tavaramerkit, patentit tai liiketoimintaan liittyvillä argumenteilla esitetyt uhkat tai oikeudellinen haastaminen. Tästä esimerkkinä käyn lyhyesti läpi hyperstruktuuriryhmän toimintaan liittyneet kiistat: (1) Nelson vaati ryhmää vaihtamaan projektinsa nimeä, ja koska ryhmä ei halunnut riitautua Nelsonin kanssa se vaihtoi projektinsa nimen Gzz:ksi. Nelson toimi lakimiehensä välityksellä, joten uhka tuli rekursiivisen yhteisöllisyyden ulkopuolelta. (2) Keväällä 2003 Nelson väitti, ettei olettanut olevansa OS-projektissa ja vetosi persoonalliseen argumenttiin, ettei halua antaa omaa elämäntyötään vapaasti kehitettäväksi. Tämän lisäksi Nelson käytti argumenttina patenttiaan Zigzag-tiedostorakenteelle ja pyysi ryhmältä lisenssimaksua. Tähän ryhmä ei suostunut vaan jatkoi kehitystään vapaana, valitsemalla tietomalliksi avoimen standardin RDF:n, vaihtaen samalla nimensä Fenfireksi. (3) Viimeinen patenttiongelma ilmeni syksyllä 2003, jolloin paljastui Harlan Hughin[109] rekisteröimä BrainTeknologian yhdysvaltalainen patentti, jolla oli samankaltaisuuksia Fenfiren kehittämän teknologian kanssa. Neuvotteluissa ei edetty ja patenttiasian ratkaisu jäi auki. Kaikki Hyperstruktuuriryhmän kokemat uhkat ohjelmistoprojektin vapaalle toimintavalle olivat lainsäädännöllisiä uhkia: koska niiden todellista luonnetta ei pystytty selvittämään neuvottelujen tai oikeudellisten prosessien kautta, aiheuttivat nämä ”performatiiviset” uhat sen, että projekti varmisti pysyvänsä vapaana rekursiivisesti yhteisöllisenä projektina tekemällä

[109] Harlan Hughin esittely Internetissä http://www.thebrain.com/company/bios/hhugh.html, tarkistettu 6.6.2006.

muutoksia, joilla se vältti uhkat. Kuitenkaan viimeisestä patenttiongelmasta ei löydetty ulospääsyä ja siksi projektin toiminta on ollut siitä lähtien hyvin pienimuotoista.



Kaavio 6: Ohjelmistotoiminnan julkisuuden kaksi tasoa sekä avoimen-suljetun välimaasto

Vastauksena Keltylle totean, että rekursiivinen yhteisöllisyys kattaa myös siinä mielessä julkisen alueen, että jos ongelmat tuodaan oikeusprosessin osallistuvat ne silloin myös toimintaan ja säätämiseen, joka liittyy säädäntömuotoihin ja liiketoimintatapoihin. Siksi toiminta, johon liittyy informaationaalista materiaalia joko sisällöllisesti tai toiminnan kohteena, on aina jollain tavalla poliittista ja määrittelee julkisella alueella yhteisesti toimimista. Siksi valinta yhteisön käsittestä on poliittinen. Rekursiivinen yhteisöllisyys kuvaakin sitä, että tällä alueella toimii monia erilaisia rekursiivisia yhteisöjä monilla eri foorumeilla ja toiminnan tasoilla.

Kaavio 6 osoittaa sen, kuinka ohjelmistoihin liittyvä toiminta tapahtuu kahdessa julkisen toiminnan ulottuvuudessa, ja jos kehitystyö tuodaan näitä julkisia yhdistävälle alueelle, sitä kautta ohjelmistojen lainsäädäntöä voidaan kehittää yhteisöllisesti. Vastavuoroisuuden normit sekä viimeistelemätön konsensus (rough consensus) syntyy vain rekursiivisen yhteisöllisyyden alueella. Kutsun rekursiiviseksi yhteisöllisyydeksi aluetta, jossa erilaiset yhteisöt neuvottelevat toimintavoistaan argumenttien, vapaiden ja

avointen liikkeiden sekä ohjelmistolisenssien kautta. Suljetun tasolla pyritään säätämään tai liiketoiminnallisesti hallitsemaan avointa lähdekoodia ja avoimia innovaatioita. Se kuitenkin onnistuu vain jos ohjelmiston tuottamiseen liittyvät sosiaaliset siteet tuodaan rekursiivisen yhteisöllisyyden piiriin. Lukan projekti pyrki pitämään tuotoksensa tässä avoimessa piirissä, jonka ulkopuolelta eri toimijat muodostivat projektille uhkia. Rekursiivisen toiminnan alueella ohjelmistotoimijoiden keskinäinen aktiivinen luottamus ja osaamisen osoittaminen luovat perustan yhteistoiminnan rekursiiviselle jatkuvuudelle sekä ohjelmistotyökalujen ja lainsäädännön kehittämiselle.

Tutkimukseni lopuksi totean, että on tärkeää myös pohtia, miten globaali ja paikallinen toiminta yhdistyvät vapaitten ohjelmistojen tuottamisessa? Internet muodostaa globaalin verkoston toimijoille. Yritykset ja instituutiot voivat kuitenkin osallistua avoimeen innovaation kehitykseen rekursiivisen yhteisöllisyyden alueella. Alapuolella kuviossa kuvatut ”yritykset” ja ”instituutiot” kuitenkin toimivat aina paikallisesta toiminta-kulttuurista käsin, jossa heidän toimintansa nojaa joko paikallisen lainsäädäntöön tai liiketoimintamalliin. Tämä osoittaa kuvion moderatiivisuuden, koska kuvio ja toimijoiden asema siinä on sidottu kulttuurisidonnaisuuteen. Kuvion keskipisteessä on kuvattu, kuinka avoimen/suljetun kulttuurin toimintamallien riitojen kärjistyessä syntyy oikeudellisia konflikteja. Tämä alue on kuitenkin demokraattinen kehto myös ohjelmistokehityksen uudenlaiselle toimintakulttuurille ja lainsäädännön kehittämiselle.

Vertaileva tutkimusote, jossa tutkittaisiin monenlaisia vapaita ohjelmistoprojekteja voisi antaa syvyyttä hahmottelemalleni teoreettiselle mallille. Kun nämä tutkittavat projektit tulisivat erilaisista toimintakulttuureista olisi mahdollista selvittää, millaiset projektit pystyisivät parhaiten toimimaan rekursiivisen yhteisöllisyyden piirissä.

LÄHTEET

Barry, Andrew. 2001.Political Machines. London:Athlone Press New York.

Berry, David M. 2004. The Contestation of Code. Critical discourse studies Volume 1 (1), 65-89.

Berry, David M & Moss, Giles. 2006. Libre Commons = Libre Culture + Radical Democracy. Verkossa käsikirjoitus http://www.noemalab.org/sections/ideas/ideas_articles/pdf/berry_moss_libre_commons.pdf, luettu 29.4.2006.

Bergquist, Magnus & Ljungberg, Jan. 2001. The Power of Gifts. Organizing Social Relationships in Open Source Communities. Information Systems Journal. Special Issue on Open Source. No 11. London: Blackwell Science. 305-320.

Bergquist, Magnus. 2003. Open Source Software Development as Gift Culture: Work and Identity Formation in an Internet Community. Teoksessa Garsten, Christina & Wolff, Helena (toim.): New Technologies at Work. People, Screens and Social Virtuality. Oxford: Berg Publishers.

Bezroukov, Nikolai. 1999. Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism). First Monday 4(10)

Bourdieu, Pierre. 1990. The Logic of Practice. Cambridge: Polity.

Bridgman, Todd & Willmott, Hugh. 2006. Institutions and Technology. The Journal of Applied Behavioral Science. Volume 42 (1), 110-126.

Castells, Manuel. 2004. Informationalism, Networks, and the Network Society: A Theoretical Blueprint. Teoksessa Castells, M (toim.): The Network Society: A Cross-Cultural Perspective. London: Edward Elgar Pub. 3-45.

Castells, Manuel & Himanen, Pekka. 2004. Institutional Models of the Network Society : Silicon Valley and Finland. Teoksessa Castells, M (toim.): The Network Society: A Cross-Cultural Perspective.London: Edward Elgar Pub. 49-83.

Cheal, David. 1988. Gift Economy.Routledge: New York.

Coleman, Gabriella. 2005. The Social Construction of Freedom in Free and Open Source Software: Hackers, Ethics and Liberal Tradition. Academic Dissertation. The University of Chicago.

Creswell, John W. 1998. Qualitative Inquiry and Research Design. Sage: Thousand Oaks.

Day, Ronald E. 2004. Poststructuralism and Information Studies. Teoksessa Cronin B. (toim.): Annual Review of Information Science and Technology. Volume 39. New Jersey: Information today Inc. 575-609.

Denzin, Norman. 2004. Online Environments and Interpretive Social Research. Teoksessa Mark Johns, Shing-Ling Sarina Chen & G.Jon Hall (toim). Online Social Research. Peter Lang Publishing: New York. 1-12.

Dewey, John. 2006. Julkinen toiminta ja sen ongelmat, suom. Mika Renvall. Vastapaino: Tampere.

Douglas, Mary. 1999. Pyyteettömiä lahjoja ei ole. Teoksessa Mauss. M, 1999. Lahja. Helsinki:Tutkijaliitto ry. 7-23.

Durkheim, Emile. 1980. Uskontoelämän alkeismuodot. Helsinki: Tammi

Fogel, Karl. 2005. Producing Open Source Software. Verkossa http://producingoss.com/producingoss.pdf

Gallivan, Michael J. 2001. Striking a Balance Between Trust and Control in a Virtual Organization: a Content Analysis of Open Source Software Case Studies. Information Systems Journal. Special Issue on Open Source. No 11. London: Blackwell Science. 277-304.

Gaonwar, Dilip Parameshwar & Povinelli, Elizabeth A. (2002). Technologies of Public Forms. Public Culture 15(3), 385-297.

Geertz, Clifford. 1973. Interpretation of Cultures. New York:Basic Books, cop.

Giddens, Anthony. 1996. Riski, luottamus ja refleksiivisyys. Teoksessa Beck, U., Giddens, A. & Lash, S. : Nykyajan jäljillä. Jyväskylä: Gummerus. 249-267.

Giddens, Anthony. 1990.The Consequences of Modernity. Cambridge: Polity Press.

Godelier, Maurice. 1999. The Enigma of the Gift. Cambridge: Polity Press.

Graeber, David. 2001. Toward an Anthropological Theory of Value. New York: Palgrave.

Grönfors, Martti. 1982. Kvalitatiiviset kenttätyömenetelmät. Porvoo: Wsoy.

Habermas, Jurgen. 2004. Julkisuuden rakennemuutos. suom. Veikko Pietilä. Tampere: Vastapaino.

Hagstrom, Warren O. 1982. Gift Giving as Organizing Principle in Science. Teoksessa Barnes, Barry & Edge, David (toim.): Science in Context. Milton Keynes: The Open University Press. 21-34.

Hautamäki, Antti. 2003. Kyllä Amerikka opettaa:hyvinvointivaltio muutosten edessä. Helsinki: Edita.

Heiskala, Risto. 2006. Sosiaaliset innovaatiot ja hegemonisten mallien muutokset: kuinka tulkita Suomen 1990-luvun murrosta? Teoksessa R.Heiskala, E.Luhtakallio (toim.): Uusi jako. Helsinki: Gaudeamus. 202-217.

Herring, Susan C. 2002. Computer-Mediated Communication on the Internet. Teoksessa Cronin B. (toim). Annual Review of Information Science and Technology. volume 36. New Jersey: Information today Inc. 109-168.

Holopainen, Reijo. 2001. Sonera ja SysOpen rahoittavat radikaalia Zigzag käyttö-järjestelmää. Verkossa http://www.digitoday.fi/showPage.php?page_id=9&news_id=5382. Luettu 3.4.2006.

Holvas, Jakke. 2000. Radikaali vaihto; Marcel Maussin Lahja johdatuksena ei-taloudelliseen vastavuoroisuuteen. Tiede ja edistys 2000 (2), 106-127.

Howarth, David. 2005. Applying Discourse Theory. Teoksessa Howarth,D., Torfing,J (toim.): Discourse Theory in European politics. Palgrave Macmillan: Hampshire, 318-349.

Hyysalo, Sampsa. 2006. Innovaation käsite. Teoksessa Ali-Yrkkö, J., Honkanen, P., Hyysalo, S. & Uotinen, J (toim.): Globalisaatio, innovaatio ja kansalaisuus – ProACTiivisia näkökulmia innovaatiopolitiikkaan. ProACT-tutkimusohjelman loppuraportti. Helsinki: Kauppa- ja teollisuusministeriö & Tekes.

Ilmonen, Kaj. 2005. Luottamuksen operationalisoinnista. Teoksessa P.Jokivuori (toim.): Sosiaalisen pääoman kentät. Jyväskylä: Minerva Oy. 45-68.

Ilmonen, Kaj. 2004. The problem of disinterestedness and reciprocity in gift economy. Verkossa http://www.sifo.no/files/Kai_Ilmonen.pdf. Luettu 27.2.2006.

Ilmonen, Kaj. 2000. Epilogi. Teoksessa Ilmonen, Kaj (toim). Sosiaalinen pääoma ja luottamus. Jyväskylä: Sophi. 170-179.

Ilmonen, Kaj & Jokinen, Kimmo. 2002. Luottamus modernissa maailmassa. Jyväskylä: Sophi.

Ingo, Henrik. 2005. Avoin elämä:Näin toimii Open Source. Espoo: OtaMedia.

Isosaari, Kyösti. 2003. Tietotekniikan perusteita etsimässä, Kauppalehti 24.3.2003.

Kelty,Christopher. 2005. Geeks, Social Imaginaries, and Recursive Publics. Cultural Anthropology. Volume 20 (2), 185-214.

Kelty, Christopher. 2003. Qualitative Research in the Age of the Algorithm: New Challenges in Cultural Anthropology. Verkossa http://cygnus.rlg.org/en/page.php?Page_ID=2201. Luettu 9.5.2006.

Knorr-Cetina, Katrin & Bruegger, Urs. 2002. Global Microstructures. American Journal of Sociology. Volume 107 (4), 905-950.

Korkki, Raisa. 2000. Tekoäly ja Xanadu lennättävät korkealle. Verkossa http://lipas.uwasa.fi/yolehti/06-00/23.html. Luettu 3.5.2006.

Koskinen, I., Alasuutari, P., Peltonen, M. 2005. Laadulliset menetelmät kauppa-tieteissä. Gummerus: Jyväskylä.

Kuula, Arja. 2006. Tutkimusetiikka. Tampere: Vastapaino.

Laclau, Ernesto & Mouffe, Chantal. 1985. Hegemony and Socialist Strategy. London: Verso.

Layder, Derek. 1998. Sociological Practice, London: Sage.

Latham,R. & Sassen,S. 2005 Digital Formations: Construction an Object of Study. Teoksessa Latham, R & Sassen (toim.): Digital Formations. New Jersey: Princeton University Press. 1-35.

Latour, Bruno. 2005. Reassembling the Social. Oxford: Oxford University Press.

Latour, Bruno. 1996. Ces Réseaux Que la Raison Ignore - Laboratoires, Bibliothèques, Collections. Verkossa
http://www.ensmp.fr/~latour/articles/article/064.html
Luettu 19.4.2006.

Latour, Bruno. 1987. Science in Action. Open University Press: Milton Keynes.

Latour, Bruno & Woolgar, Steve. 1979. Laboratory Life. New Jersey: Princeton University Press.

Lee, Benjamin & LiPuma, Edward. 2002. Cultures of Circulation: The Imaginations of Modernity. Public Culture 14 (1), 191-213.

Lin, Yuwei. 2005. The Future of Sociology of FLOSS. Verkossa: http://firstmonday.org/issues/special10_10/ Luettu 12.12.2005.

Lukka, Tuomas & Ervasti, Katariina. 2001. GZIGZAG – A Platform for Cybertext Experiments. Teoksessa M. Eskelinen, R. Koskimaa (toim.): Cybertext Yearbook 2000. Saarijärvi: Gummerus. 141-151.

Mackenzie, Adrian. 2006. Cutting the Code. New York: Peter Lang Publishing.

Mackenzie, Adrian. 2005. Performativity of the Code. Theory, Culture & Society 22 (1), 71-92.

Mackenzie, Adrian. 2001. Open Source Software: When is a Tool? What is a Commodity. Science As Culture. Volume 10 (4), 541-552.

Manning, Philip. 1992. Erwing Goffman and Modern Sociology. California: Stanford University Press.

Mauss, Marcel. 1999. Lahja. Helsinki: Tutkijaliitto ry.

Miettinen, Reijo. 2006. Hajautettu luominen ja tiedon omistusoikeudet. Tieteessä Tapahtuu 2006 (4), 5-14.

Miettinen, R., Lehenkari, J., Hasu, M. & Hyvönen J. 1999. Osaaminen ja uuden luominen innovaatioverkoissa. Helsinki: Taloustieto.

McNeill, J.R. & McNeill, William H. 2005. Verkottunut ihmiskunta. Tampere: Vastapaino.

Millar, C., Choi, C., Russell, E. & Kim, J-B. 2005. Open Source Communities: an Integrally Informed Approach. Journal of Organizational Change Management vol. 18 (3), 259-268.

Moody, Glyn. 2001. Kapinakoodi. Jyväskylä: Gummerus.

Mouffe, Chantal. 2005. On the Political. Abingdon: Routledge.

Mouffe, Chantal. 2000. The Democratic Paradox. London: Verso.

Nelson, Ted. 1987. Literary Machines: Edition 87.1. San Antonio:[Ted Nelson].

Oinas-Kukkonen, Henry. 2002. World Wide Webin kummisedän näkemyksiä ja kokemuksia uuden median maailmasta. Tekniikan Waiheita 2002 (2), 6-26.

Payne, Geoff & Williams, Malcolm. 2005. Generalization in Qualitative Research. Sociology 39 (2), 295-314.

Perens, Bruce. 1999. The Open Source Definition, Teoksessa, C. DiBona, S.Ockman ja M.Stone (toim.): Open Sources; Voices from the Open Source Revolution. Sebastopol: O`Reilly. 171-188.

Raymond, Eric S. 1999. The Cathedral and the Bazaar. Beijing: O`Reilly.

Saastamoinen, Jukka. 2002. Aika fiksu ihmiseksi, Kauppalehti Optio-liite 7.2.2002.

Schatzki, Theodore.R. 2001. Practice Mind-ed Orders. Teoksessa Schatzki, T. Knorr-Cetina K. & Von Savigny, E. (toim.): The Practice Turn in Contemporary Theory. London: Routledge.

Sennett, Richard. 2006. The Culture of the New Capitalism. New Haven: Yale University Press.

Siapera, Eugenia. 2006. Multiculturalism Online. European Journal of Cultural Studies. Volume 9 (1), 5-24.

Siitonen, M., Sivunen, A., Feldt,T., Neittaanmäki, M., Metsäpelto, R-L., Pulkkinen,L. 2002. Mitä hyötyä tieto- ja viestintätekniikasta on perheelle? Perheportaali-projekti 1.vaiheen esitutkimus: Jyväskylän yliopisto.

Silvonen, Jussi. 2003. Linux ja vapaat ohjelmat- haaste informaatiokapitalismille? Teoksessa Heiskanen,J & Mäntylä,J (toim.): Marx IT- informaatiokapitalismin kriittistä tarkastelua. Helsinki: Hakapaino Oy. 117-129.

Smith, Anna Marie. 1998. Laclau and Mouffe, London: Routledge.

Smith, Gregory. 1989. A Simmelian Reading of Goffman. Academic Dissertation. University of Salford.

Stallman, Richard. 1999. The GNU Operating System and the Free Software Movement. Teoksessa C. DiBona, S.Ockman ja M.Stone (toim.): Open Sources: Voices from the Open Source Revolution. Sebastopol: O`Reilly. 53-70.

Stallman, Richard. 2002. Why ”Free Software” is Better Than ”Open Source”. Teoksessa Stallman, Richard: Free Software, Free Society: Selected Essays of Richard M.Stallman. Boston: Gnu Press. 55-60.

Suchman, Lucy. 1997. Centers of Coordination: A Case and Some Themes. Teoksessa Resnick, L. B., Säljö, R., Pontecorvo, C., & Burge, B. (toim.): Discourse, Tools, and Reasoning: Essays on Situated Cognition. Berlin: Springer-Verlag. 41-62.

Taylor, Charles. 1992. The Politics of Recognition. Teoksessa Taylor & Gutmann (toim.). Multiculturalism and the “Politics of Recognition”. New Jersey: Princeton University Press. 25-73.

Thompson, J.B. 1995. The Media and Modernity. Cambridge: Polity Press.

Tikka, Juha-Pekka. 2003. Microsoft varoittaa yhä Linuxista, Iltasanomat 19.9.2003.

Turunen, Annamari. 2006. Innovations as Communication Processes; a Legal Architecture for Governing Ideas for Business. Academic Dissertation. University of Lapland. Rovaniemi: University of Lapland Press.

Vaden, Tere & Vainio, Niklas. 2005. Software Strategies for Sustainable Development. Teoksessa Olli Hietanen (toim.): University Partnership for International Development. Finnish Development Knowledge. Finland Futures Research Centre:Turku School of Economics and Business Administration. 222-234.

Vainio, Niklas & Vaden, Tere. 2006. Valoa basaarista-Internetin vapaan ja avoimen koodin kollektiivinen kehitystyö. Teoksessa Parviainen, Jaana (toim.): Kollektiivinen asiantuntijuus. Tampere: Tampereen yliopistopaino Oy. 341-365.

Van den Boomen,M & Schäfer,M.T. 2005. Will the Revolution be Open-Sourced: How Open Source Travels through society? Teoksessa Wynants & Cornelis (toim.) How open is the future? Verkossa http://crosstalks.vub.ac.be/publications/Howopenisthefuture/howopenfuture_CROSSTALKSBOOK1.pdf. 31-67. Luettu 10.11.2005.

Walker, John Mark. 2006. There is no Open Source Community. Verkossa http://www.onlamp.com/pub/a/onlamp/2006/01/12/no_oss_community.html. Luettu 20.4.2006.

Weber, Steven. 2004. The Success of Open Source. Cambridge: Harvard University Press.

Weber, Steven. 2005. The Political Economy of Open Source Software and Why it Matters. Teoksessa Latham, R & Sassen (toim.): Digital Formations. Princeton University Press: New Jersey. 178-211.

Williams, Malcolm. 2002. Generalization in Interpretive Research. Teoksessa Tim May (toim.): Qualitative Research in Action. London: Sage. 125-143.

Williams, Malcolm. 2000. Interpretivism and Generalization. Sociology 34 (2): 209-224.

Wittgenstein, Ludwig. 1981. Filosofisia tutkimuksia. Suom. Heikki Nyman. Juva: Wsoy.

Vähämäki, Jussi. 2006. Fetisismi. Teoksessa M.Jakonen, J.Peltokoski ja J.Vähämäki (toim.): Uuden työn sanakirja. Helsinki: Tutkijaliitto. 281-289.

Zeitlyn, David. 2003. Gift Economies in the Development of Open Source Software: Anthropological Reflections. Research Policy 32 (7), 1287-1291.

LIITE 1:

Hypertekstuaalisten keksintöjen lyhyt historia – Memex, Ted Nelson ja Tuomas Lukka

Ted Nelsonin ja Tuomas Lukan[1] tärkeimpiä vaikuttajia hypertekstuaalisten keksintöjen suunnittelulle olivat Vannevar Bush ja Douglas Engelbart. Toisen maailmansodan jälkeisenä aikana kyberneetikot ja teknovisionäärit kehittivät ajatuksia ihmisen yksilölliselle tiedonhallinnalle (Mattelart 2003, 50-71). Nämä ajatukset ilmenivät ideoina teknologisista apuvälineistä ihmisen aisteille, erityisesti muistille. Vannevar Bush toimi Yhdysvalloissa presidentti Rooseveltin tieteellisenä avustajana. Bushin innovatiivinen ajattelu huipentui ajatukseen Memex[2] laitteesta. Bush esitteli idean Memexistä ensimmäisen kerran artikkelissaan vuonna 1945:

> Kuvittele tulevaisuuden henkilökohtaiseen käyttöön laite, joka on eräänlainen henkilökohtainen arkisto ja kirjasto. Laite tarvitsee nimen, ja keksiäkseni jotain sattumanvaraisesti, ”*memex*” saa kelvata. Memex on laite, johon henkilö tallentaa kaikki kirjansa, levynsä ja kommunikaationsa ja joka on mekanisoitu niin, että sitä voi käyttää erittäin nopeasti ja joustavasti. Se on laajennettu henkilökohtainen lisäke käyttäjänsä muistiin. Laite koostuu työpöydästä, ja vaikka sitä luultavasti voi täyttää välimatkan päästä, se on ensisijaisesti se huonekalu jonka ääressä käyttäjä työskentelee. Yläpuolella on kalteria, läpikuultavia näyttöjä, joille materiaali voidaan heijastaa helposti luettavaksi. Lisäksi laitteeseen kuuluu näppäimistö sekä nappuloita ja vipuja. Muuten se näyttää tavalliselta työpöydältä.[3]
> (Vannevar Bushin artikkelista Memexin esittelyn alusta, suomentanut Juha Leminen)

Memexiä ei ikinä onnistuttu kehittämään toimivaksi teknologiaksi, mutta se oli ensimmäinen suunnitelma hypermediajärjestelmästä. Memex perustui aikakautensa teknologiaan, joita olivat mikrofilmaus, valokennot ja optinen koodaus. Bushin idean perustana oli, että ihminen voi liittää tekstit, piirrokset, valokuvat ja muistiot haluamallaan tavalla asiayhteyksiin perustuen. Ne järjestettäisiin yhtenäiseksi dokumentaatioksi Memex-laitteen avulla. Bushin idean merkittävyys oli kehittää teknologinen laite ihmiselle henkilökohtaiseksi työasemaksi, kyseessä oli suunnitelma tietokoneen esimuodosta. 1960-luvulla Douglas Engelbartin onnistui kehittää ensimmäinen toimiva teknologinen keksintö Bushin ajatusten perusteella. Engelbartin tutkimuksessa[4] (1962) esiteltiin ihmisen ja tietokoneen välistä vuorovaikutusta, jonka

---

[1] Lukka mainitsee Bushin ja Engelbartin merkityksen (Lukka ja Ervasti 2001, 143)
[2] Memex on lyhenne sanoista Memory Extender.
[3] http://www.theatlantic.com/doc/194507/bush/4, tarkistettu 3.5.2006.
[4] http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html, tarkistettu 3.5.2006

avulla voitaisiin laajentaa ja tukea ihmisen luovaa toimintaa. Engelbart kehitti hiiren vuosina 1963–1964 Stanfordin tutkimusinstituutissa. Ensimmäinen hiiri oli puinen laatikko, jossa oli pyörä liikkeiden säätämiseen. Laitteen kehittyessä se nimettiin joustavien kaksiulotteisten liikkeiden ja johtohäntänsä perusteella[5]. Tietokoneen hiiri muodosti myöhemmin perustan Apple Macintoshin ja Microsoft Windowsin käyttöjärjestelmien tiedonhallinnan tekniikoille.

Vuonna 1960, toisena vuotenaan opiskellessaan sosiologiaa, Ted Nelson osallistui kurssille ”*Computers for the Social Sciences*”. Nelson näki kurssilla ensimmäistä kertaa IBM 7090 tietokoneen. IBM 7090 oli ainoa tietokone, joka oli silloin Harvardissa käytettävissä. Kun Nelson näki tietokoneen näytön grafiikan, hänen hyperteksti-ideansa saivat perustan. (Rheingold 2000, 300). Nelson (2002, 51) ymmärsi, että tietokoneiden graafisen näytön kehittämisessä olisi ihmiskunnan tulevaisuus. Nelson laajensi ideansa koskemaan työskentelyä visuaalisilla näytöillä ja niiden kehittyviin mahdollisuuksiin lukea, kirjoittaa ja julkaista ikuisesti laajeneviin elektronisiin kokoelmiin. Vuonna 1965 hän julkaisi artikkelin[6], jossa hän käytti ensimmäisen kerran termejä hypermedia ja hyperteksti.

Nelson perusti Xanadu-projektin, jonka tavoitteena oli kehittää virtuaalinen maailmankirjasto, jossa teksteillä olisi erilaisia yhteyksiä toisiinsa[7]. Projekti sai nimensä Coleridgen runosta Kubla Khan. Coleridgen runossa yhdistettiin assosioiden erilaisia mielikuvia. Nelson nimitti Xanadun kirjallisen muistin maagiseksi paikaksi, johon olisi tallennettu maailman kaikki tieto[8]. Nelson (1991) esitti vuonna 1972, kuinka Vannevar Bushin esittämät ideat olisivat siirrettävissä tietokoneeseen hypertekstijärjestelmäksi dokumenttien hallintatavaksi. Xanadun perusideassa yhdistetään kaksi dokumenttien käyttömuotoa. Dokumenttien viittaustapoja on kahta komplementaarista muotoa. Ensimmäinen viittamisen tapa on säilyvät linkit, jotka eivät katkaise dokumenttien suhteita. Toinen tapa on transkluusiot, joissa dokumentit

---

[5] Kuva Engelbartin hiirestä sivustolla http://sloan.stanford.edu/MouseSite/, tarkistettu 3.5.2006.
[6] Nelson, Theodor Holm: A File Structure for the Complex, The Changing and the Indeterminate, ACM 20th National Conference, sivut 84-100, 1965.
[7] Vuonna 2002 Nelson kertoi väitöskirjassaan, jonka hän teki Keion yliopistossa Japanissa, Xanadun liittyvien projektien historian ja omien hypertekstiajatustensa taustan. Tutkimuksessani käsittelen Nelsonin ideoita näkökulmasta miten ne liittyvät Tuomas Lukan avoimiin ohjelmistoprojekteihin.
[8] Belinda Barnet (2005): The Magical Place of Literary Memory™: Xanadu, http://www.latrobe.edu.au/screeningthepast/firstrelease/fr_18/BBfr18a.html, tarkistettu 3.5.2006.

sisällyttävät mahdollisuuden käyttää uudestaan tekstistä leikattua osaa haluamallaan tavalla yli dokumenttien rajojen. Näiden kahden dokumentaalisen tyylin käyttäminen symmetrisesti visuaalisessa paralleelisessa dokumenttiavaruudessa muodosti Xanadun toiminnan perusajatukset (Nelson 2002, 59, 90, 126). Myöhemmin Nelson nimesi tämän dokumenttiavaruuden Zigzagiksi. Zigzag oli alun perin suunniteltu Xanadun tietorakenteeksi. Zigzagissa kaikki tieto on samassa tietojärjestelmäavaruudessa, jossa tietoa yhdistellään kaksisuuntaisin viittauksiin ilman linkkien katkeamista. Nelsonin näkemyksessä Zigzag avaruudessa ei ollut nykytietokoneitten kansioita, tiedostoja tai sovelluksia, vaan mallissa tieto esitetään soluissa, jotka ovat kytkeytyneitä toisiinsa. Zigzag on siis tietokannan ja käyttöliittymän yhdistelmä (Lukka ja Ervasti 2000).

Vuonna 1994 Tim Berners-Leen kehittämä World Wide Web teki läpimurtonsa tietokoneen käyttäjien keskuudessa. Mosaic-selaimen avulla tuhannet ihmiset ympäri maailmaa käyttivät sitä päivittäiseen tiedonhakuun. Ted Nelsonille pettymys oli suuri. Xanadu-projekti ei ollut toteuttanut toimivaa ohjelmistoa ja nyt WWW tuli yleiseen käyttöön särkien Nelsonin unelman. WWW ei kuitenkaan toteuttanut Nelsonin ideoita dokumenttien suhteista. Vuonna 1992 Autodesk oli lopettanut Xanadu-projektin rahallisen tukemisen, jonka jälkeen projekti oli menettänyt toimintaedellytyksensä. Vuonna 1992 WWW oli jo käytössä, mutta ei ollut tehnyt läpimurtoaan. (Berners-Lee 2000, 64-66. Nelson 2002, 107-111.) WWW:n läpimurron jälkeen Xanadu-projektin tavoitteet muuttuivat. Nelson muotoili Xanadun tavoitteet yhteensovitettaviksi WWW:n teknologiaan. Väitöskirjassaan Nelson (2002) esitti kehittäneensä 34 hyperteksteihin liittyvää suunnitelmaa. Näitä ideoita oli toteutettu 19 Xanadun sisäisessä projektissa, joista 12 oli toteutettu prototyypin tasolle. Zigzag-rakenteeseen liittyviä projekteja olivat toteuttaneet Andrew Pamin projektit (zz 86 ja zz 97) ja muutamat muut pienimuotoiset projektit[9]. Avoimen lähdekoodin projektina ensimmäinen Zigzag toteutus oli Tuomas Lukan aloittama Gzigzag. Vuonna 1999 O`Reillyn konferenssissa Nelson oli vapauttanut aikaisempien projektiensa koodia Open Source lisenssien alle vapaasti kehitettäväksi. Nuori saksalainen kehittäjä Benja herätti keskustelun Zigzagin vapaudesta Xanadun keskustelulistalla 2. marraskuuta vuonna 1999:

[9] Nelson (2004) luettelee ZigZag-projektien historian taulukossa 1, http://jodi.ecs.soton.ac.uk/Articles/v05/i01/Nelson/, tarkistettu 9.5.2006.

> Arvoisa Zigzag kehittäjätiimi. Minulla on teille kysymys koskien uskomusta. Kun luin ensimmäisen kerran ZZ:sta, minua kiehtoi ajatus sen mahdollisuuksista ohjelmistona. Arviointikuukauteni aikana, tein kokeiluja sillä ja se ei ottanut kauanSaavuttaa rajoja kokeiluille. Mutta olen edelleen innostunut ja voisin helposti kuvitella sen olevan paradigma tietokoneen käytölle tulevaisuudessa. Mutta arviointikuukauteni jälkeen, lopetin käyttämästä ZZ:ia. En rekisteröitymismaksun takia. Jopa kaltaiseni köyhälle opiskelijalle se on maksullinen. Ei, minun ongelmani oli moraalinen. Uskomukset koskien Open Sourcea / Free Softwarea ovat eroavia. Uskomukset koskien omistusoikeuksia ovat eroavia. Jos uskot, että on oikein pitää ohjelmisto suljettuna, hyväksyn tämän helposti. Mutta täytyykö systeemin, jonka uskomme olevan perustavanlaatuinen tietokoneen rakenteelle kuten Zigzag olla suljettu. Kuten asiat ovat, jos Zigzag tulee menestymään, XOC tulee olemaan uusi Microsoft. Ja vaikka emme hyväksy käyttäjän oikeuksia systeemiin, niin kun ohjelmisto heidän systeemeilleen on ei-monopolistinen (tai vapaa), meidän pitää hyväksyä jossain vaiheessa, että joku aikoo ohjelmoida vapaata ZZ. Ja suljetun ja vapaan ZZ:n käyttäjät, niin heille ei tule olemaan helppoa aikaa vaihtaa tietoa.
> Vaihtoehto jonka toivoisin teidän ottavan on se, mitä Eric Raymond kuvaa uudessa tekstissään "*vapauta ohjelmisto, myy brändiä*". Vapauta ZZ ja anna jokaisen jakaa kopioita ja ohjelmoida uusia sovelluksia protokollista, mutta säilytä oikeudet Zigzag nimelle ja symbolille. Anna jokaisen kutsua heidän jäljitelmää " *ei-viralliseksi Zigzag jäljitelmäksi*", mutta kun joku haluaa Zigzag symbolin, heidän täytyy maksaa.

Tämän jälkeen Benja kertoi pyrkivänsä kirjoittamaan vapaan version Zigzagista ja pyytää Nelsonilta vastausta. Viestiin tuli vain yksi vastaus, eikä Nelson tai kukaan hänen puolestaan vastannut kysymykseen. Gary Wolf[10] on kirjoittanut artikkelin "Xanadun kirous" (1995) , jossa Nelson esitellään kiisteltynä nerona, jonka projektit eivät koskaan valmistuneet toimivan teknologisen innovaation tasolle. Nelson kuitenkin kiisti vastinessaan[11] suurimman osan Wolfin väitteistä ja on jatkanut toimimista erilaisissa projekteissa vielä 2000-luvullakin.

Haastattelussaan vuonna 2002 Nelson (Oinas-Kukkonen 2002, 16-26) kertoi yrittäjän voimasta ja ihannoi yrityksiä toteuttaa sekä rahoittaa informaatioteknologisia mullistavia keksintöjä. Nelson työskenteli 1990-luvulta 2000-luvun alkupuolelle pääosin japanilaisissa yliopistoissa vierailevana professorina. 2000-luvulla hän on työs-kennellyt erilaisissa yliopistohankkeissa vierailevana professorina Euroopassa ja Yhdysvalloissa.

Tuomas Lukka väitteli kiitettävällä arvosanalla tohtoriksi fysikaalisesta kemiasta Helsingin yliopistosta vain 21-vuotiaana vuonna 1995. Tohtoriksi valmistuttuaan hän palasi takaisin aiheeseen, joka häneltä oli jäänyt ratkaisematta. Häntä kiinnosti erityisesti miten tietokone saadaan helpommin käytettäväksi. Lukka lähti jatko-

---

[10] http://www.wired.com/wired/archive/3.06/xanadu.html, tarkistettu 28.3.2006
[11] http://xanadu.com.au/ararat , tarkistettu 28.3.2006

opiskelijaksi arvostettuun Society of Fellowsiin Harvardin yliopistoon. Hän opiskeli Harvardissa neuraalilaskentaa ja tekoälyteoriaa vuosina 1996–1999. Tutkimusaiheen perustana hänellä oli tietokoneiden käytettävyysongelma käytännöllisen ongelman näkökulmasta. Käytännöllisen ongelman taustalla oli ajatus, voidaanko tietokone opettaa pelaamaan kiinalaista Go–peliä. Go-peli on vanha laudalla pelattava strategiapeli, jossa visuaalinen hahmottaminen oli tärkeämpää kuin matemaattinen hahmottaminen. Pelissä rakennetaan samanaikaisesti montaa erilaista strategiaa toista pelaajaa vastaan eri puolella pelilautaa. Lukka kertoi kiinnostumisestaan tutkia tietokoneen käytettävyyttä:

> Mähän olin jo kouluaikana kiinnostunut kahdesta (-) asiasta. Toinen oli miten kvanttimekaniikka toimii, ja toinen oli että miten tietokoneet saadaan ajattelemaan. Mä tein sen väitöskirjan, jotta oppisin ymmärtämään miten kvanttimekaniikka toimii. Sen jälkeen palasin tähän tekoälypuolelle, eli että miten tietokone saadaan ajattelemaan. (Tuomas Lukan haastattelu 28.5.2001)

Toinen tärkeä asia Lukalle oli puhua vapaiden ohjelmistojen kehitysmallin puolesta[12]. Kauppalehden Optio-liitteen kannessa hän esiintyi helmikuussa 2002. Artikkeli kertoi lupaavasta Zigzag-hankkeesta ja Lukan lahjakkuudesta tutkijana[13]. Vuonna 2003 Kauppalehdessä[14] hän puhui projektistaan myös tekoälyn perustutkimuksena, jonka avulla kehitettäisiin käyttäjää hyödyttäviä sovelluksia:

> Kyse on tekoälyn perustutkimuksesta…Jos koneet saadaan toimimaan lisääntyneen ymmärryksen myötä riittävän hyvin, siitä seuraa valtavasti mahdollisuuksia, esimerkiksi teollisuussovelluksien alalla.

Lukka puhui vahvasti vapaiden ohjelmistojen puolesta Microsoftin monopolia vastaan. Artikkelin mukaan hän oli niin tiukka periaatteissaan ettei suostunut avaamaan sähköpostin liitetiedostoja, joiden tiedostomuotoja ei oltu vapaasti dokumentoitu. Hän kertoi oman näkemyksensä Free Software termin oikeasta ymmärrystavasta:

> Se käännetään ilmaissoftaksi, vaikka oikea käännös kuuluisi vapaa softa. Kyse on siitä, että jokaisella on oikeus tehdä ohjelmistoilla mitä haluaa – esimerkiksi kehittää niitä. Ja myös tarjota näin syntyviä uusia palveluja – vaikka maksusta.

Lukan mukaan Microsoftin monopoli vaikeuttaa tuotekehitystä sekä uusien innovaatioiden toteuttamista:

---

[12] Lukka, Tuomas (2002): Sota tietoyhteiskunnan tulevaisuudesta, Tekniikan Akateemiset 1/2002
[13] Saastamoinen, Jukka (2002): Aika fiksu ihmiseksi, Kauppalehti Optio-liite 7.2.2002
[14] Isosaari, Kyösti (2003): Tietotekniikan perusteita etsimässä, Kauppalehti 24.3.2003

> Siinä missä Free tukee markkinataloutta uusien kilpailumahdollisuuksien myötä, monopolistiohjelmistot edustavat sosialistista järjestelmää, joissa keskuskomitea päättää millainen ohjelma tehdään…Sitä paitsi järjestelmä hidastaa kehitystä. Monesti isot ohjelmistotalot ovat ostaneet uusia ideoita ja keksintöjä vain saadakseen ne pois markkinoilta. Sen jälkeen niistä ei ole kuulunut mitään.

LIITTEEN 1 LÄHTEET

Berners-Lee, Tim. 2000. Weaving the Web. New York: HarperBusiness.

Bush, Vannevar. 1945. As We May Think. Verkossa: http://www.theatlantic.com/doc/194507/bush/4. Luettu 6.2.2006.

Engelbart, Douglas. 1962. Augmenting Human Intellect. A Conceptual framework. Verkossa: http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html. Luettu 6.2.2006.

Lukka, Tuomas & Ervasti, Katariina. 2001. GZIGZAG – A Platform for Cybertext Experiments. Teoksessa M. Eskelinen, R. Koskimaa (toim.). Cybertext Yearbook 2000. Saarijärvi:Gummerus. 141-151.

Oinas-Kukkonen, Henry. 2002. World Wide Webin kummisedän näkemyksiä ja kokemuksia uuden median maailmasta. Tekniikan Waiheita 2002 (2), 6-26.

Mattelart, Armand. 2003. Informaatioyhteiskunnan historia. Jyväskylä: Gummerus.

Nelson, Ted. 1991. As We Will Think. Teoksessa Nyce & Kahn (toim.)From Memex to Hypertext: Vannevar Bush and the Mind's Machine. San Diego: Academic Press Professional.245-260.

Nelson, Theodor Holm. 2002. Philosophy of Hypertext. 10 January 2002. Submitted for the Ph.D. degree at Keio University.

Nelson, Theodor Holm. 2004. A Cosmology for a Different Computer Universe: Data Model, Mechanisms, Virtual Machine and Visualization Infrastructure. Verkossa http://jodi.ecs.soton.ac.uk/Articles/v05/i01/Nelson/. Luettu 29.4.2006.

Rheingold, Howard. 2000. Tools for Thought. Cambridge: Mit Press.

Wolf, Gary. 1995. The Curse of Xanadu. Verkossa http://www.wired.com/wired/archive/3.06/xanadu.html. Luettu 6.1.2006.

LIITE 2. Aineistoluettelo: ohjelmoijien haastattelut

1.Haastattelu: Tuomas Lukka
sisältö: Tuomas Lukan haastattelu 28.5.2001, haastattelija Jussi Silvonen.

2. Haastattelu: Tuukka Hastrup ja Vesa Kaihlavirta
sisältö: haastattelu 14.12.2001, haastattelija Juha L.

3. Haastattelu: Antti-Juhani Kaijanaho.
sisältö: haastattelu 14.12.2001, haastattelijat Jussi Silvonen ja Juha Leminen.

4. Haastattelu: Janne Kujala
sisältö: haastattelu 14.12.2001, haastattelijat Jussi Silvonen ja Juha Leminen.

5. Haastattelu: Tero Mäyränen.
sisältö: haastattelu 14.12.2001, haastattelijat Jussi Silvonen ja Juha Leminen.

6. Haastattelu: Kimmo Wideroos
sisältö: haastattelu 20.5.2003, haastattelija Juha Leminen.

7. Haastattelu: Tuukka Hastrup
sisältö: haastattelu 28.11.2002, haastattelija Juha Leminen.

8. Haastattelu: Hermanni Hyytiälä
sisältö: haastattelu 28.11.2002 Jyväskylässä, haastattelija Juha Leminen.

9. Haastattelu: Asko Soukka
sisältö: haastattelu 28.11.2002, haastattelija Juha Leminen

10. Haastattelu: Matti Katila
sisältö: haastattelu 28.11.2002, haastattelija Juha Leminen

11. Haastattelu: Janne Kujala
sisältö: haastattelu 28.11.2002, haastattelija Juha Leminen

12. Haastattelu: Tuukka Hastrup
sisältö: haastattelu 3.3.2003 Jyväskylässä, haastattelija Juha Leminen

13. Haastattelu: Hermanni Hyytiälä
sisältö : haastattelu 22.5.2003, haastattelija Juha Leminen

14. Haastattelu: Janne Kujala
sisältö: haastattelu 22.5.2003, haastattelija Juha Leminen

15. Haastattelu: Asko Soukka
sisältö: haastattelu 22.5.2003, haastattelija Juha Leminen

Kenttähavainnoinnit Agorassa, ryhmän kokoukset ja julkiset esiintymiset

1. Richard Stallmanin ja Gzigzag ryhmän tapaaminen Agorassa.
sisältö: Tuomas Lukan vetämän Gzz-ryhmän ja Richard Stallmanin, 75 minuuttia kehityskeskustelu 14.7.2001 Jyväskylässä. Lukka esittää Gzigzag ohjelmistodemoa Stallmanille.
Osallistujat (litteraatiossa): Tuomas Lukka, Richard Stallman, Tuukka Hastrup, Antti-Juhani Kaijanaho, Rauli Ruohonen. Havainnoijina Jussi Silvonen ja Juha Leminen.

2. Ryhmän kokous Agorassa 8.3.2002
sisältö: noin 40 minuuttia kestävä ryhmän kokous
osallistujat: Tuomas Lukka, Tuukka Hastrup, Janne Kujala, Vesa Kaihlavirta ja Tero Mäyränen.

3. Kenttähavainnointi Agorassa 20.5 – 21.5.2002 – 4 videonauhaa

4. Kenttähavainnointi Agorassa 15.8.2002 – 2 videonauhaa

5. Kenttähavainnointi Agorassa 28.11.2002

6. Kenttähavainnointi Agorassa 3.3.2003

7. Kenttähavainnointi Agorassa22.5.2003 – 1 videonauha

8. Fenfiren esiintyminen InBct -seminaarissa 27.3.2003 – 2 ääninauhaa

9. Lukan esiintyminen It Expo -tapahtumassa

10. Fenfire ryhmän kokous 8.3.2004

LIITE 3: Fenfire-ryhmän esittelyjuliste vuodelta 2003

AGORA

UNIVERSITY OF JYVÄSKYLÄ

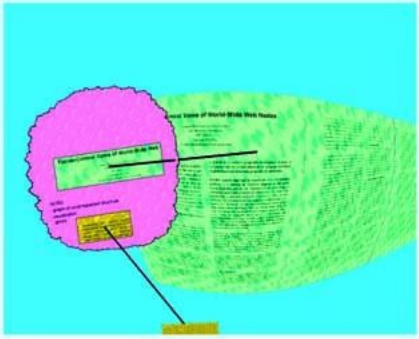
InBCT 2.1: Information Management and Sharing.

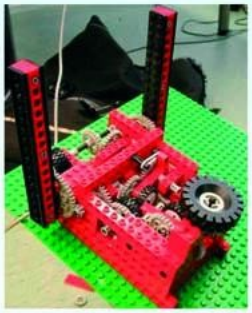
Hyperstructure and flexible user interfaces as a solution to the information flood

Team Leader: Dr. Tuomas Lukka, Agora Center

Goals

- Interactive hyperstructure: the user can connect any two items which will then be near each other in hyperspace, easily visible and accessible from one to the other.
- Novel user interfaces taking advantages of modern graphics cards
- New user interface devices
- Ad hoc teamwork: non-centralized P2P version control of structured data. Two or more persons can form a small team to change an object and subsequently submit those changes for consideration by a larger group. The software will manage all versioning.





Software

Our work forms the **Fenfire** project. Fenfire itself is the "molten core" where research is pursued, and mature submodules are productized into their own projects. Most significant "spin-offs":

Storm – a distributed storage framework

Libvob – a graphical user interface framework allowing rapid development of free-form user interfaces not constrained to boxes

LibPaper – a library for generating and rendering unique background textures (as shown behind this text).

Alph – an implementation of Xanalogical media on top of Storm

Loom – an RDF browser

Reproducibility; commodity components

We want our software and research results to be usable and reproducible at a minimal budget. Because of this, we use commodity parts (LEGO® bricks and computer mice) for creating prototypes of our novel controllers. No electrical engineering expertise needed, either!

Recent publications

Re/preprints available on request

- Fillets: Cues for Connections in Focus+Context Views of Graph-like Diagrams (IV'02)
- Freenet-like GUIDs for implementing xanalogical hypertext (HT'02)
- Rendering recognizably unique textures (IV'03)
- Storm: Using P2P to make the desktop part of the Web (HT'03)
- Fast, Cheap and in Control (poster about controllers built from LEGO® bricks and mice, IV'03)

DEPARTMENT OF PSYCHOLOGY • UNIVERSITY OF JYVÄSKYLÄ, FINLAND

LIITE 4: Hyperstruktuuri-ryhmän projektien jäsenet vuosina 2000–2004

	Tehtävä	Milloin ?
Tuomas Lukka	pääkehittäjä	2000 – 2003
Benja Fallenstein	ohjelmoija	2000 – 2004
Janne V.Kujala	tutkija	2000 – 2004
A – J Kaijanaho	ohjelmoija	2000 – 2001
Tuukka Hastrup	ohjelmoija	2000 - 2004
Rauli Ruohonen	ohjelmoija	2000 – 2001
Kimmo Wideroos	ohjelmoija	2001
Tero Mäyränen	ohjelmoija	2000 – 2002
Katariina Ervasti	hallinnollinen työ	2000 – 2001
Asko Soukka	ohjelmoija	2001 – 2004
Matti Katila	ohjelmoija	2002 – 2004
Hermanni Hyytiälä	tutkija	2002 – 2003
Vesa Kaihlavirta	ohjelmoija	2001 – kevät 2003
Otso Kivekäs	ohjelmoija	2000
Vesa Parkkinen	ohjelmoija	2000
Toni Alatalo	ohjelmoija	2001 - 2004
Marc Schiereck	ohjelmoija (Saksa)	2002?
Anton Feldmann	ohjelmoija (Saksa)	2002?