

# Distribuering av kunnskap i høyt teknologiske organisasjoner

**Johan Gudheim Hansen**

Master i informatikk

Oppgaven levert: September 2007

Hovedveileder: Knut-Helge Ronæs Rolland, IDI

Biveileder(e): Anders Christensen, IDI  
Juan Manuel Guijarro, CERN



## **SAMMENDRAG**

Kunnskap er en nøkkelkomponent i alt hva vi foretar oss, og blir beskrevet som hoveddelen i den nye kunnskapsøkonomien. Hvordan kunnskap utveksles i organisasjoner har vært et populært tema og blir sett på som avgjørende for en organisasjons evne til å lykkes. IKT baserte systemer som skal støtte utveksling av kunnskap i organisasjoner er ikke alltid like vellykket viser nylige studier. Hvorfor disse feiler har visst seg å ha en rekke sammensatte forklaringer.

Denne oppgaven gjør en studie av hvordan kunnskap blir distribuert i en høyteknologisk organisasjon. En analyse av observasjonene gjøres, og på bakgrunn av denne blir det diskutert hva dette har å si for IKT sin rolle i å støtte distribusjon av kunnskap i organisasjoner. Ved å sammenligne dette med teori understrekes viktigheten av å se på kunnskapsutvekslingen i organisasjonen som en levende prosess.



# FORORD

Denne masteroppgaven er blitt utført og skrevet ved den europeiske organisasjonen for partikkelfysikk (CERN) som en del av det norske tekniske studentprogrammet. Dette er et samarbeid mellom CERN, Norges forskningsråd og utvalgte høyskoler og universitet i Norge, derav Norges teknisk- naturvitenskaplige universitet.

Jeg vil takke min veileder Knut Helge Rolland for interessante diskusjoner og konstruktive tilbakemeldinger under veiledningen av oppgaven. Norges forskningsråd fortjener også en takk for sitt engasjement i forbindelse med CERN som gjør det mulig at norske studenter kan få sjansen til å skrive oppgave i et så spennende miljø.

Jeg vil også takke alle de jeg jobbet sammen med på CERN, og da spesielt min veileder Juan Manuel Guijarro for all hjelp. Takk til Anna Hareide for korrekturlesing.

Johan Gudheim Hansen, Genève 2007



# FIGURLISTE

FIGUR 1: KUNNSKAPSSPIRALEN TIL NONAKA OG TAKEUCHI .....	7
FIGUR 2: LÆRING, KUNNSKAP OG INNOVASJON .....	10
FIGUR 3: KONTEKSTUELLE KOMPONENTER.....	16
FIGUR 4: MEDLEMSLAND TILKNYTTET CERN .....	33
FIGUR 5: LARGE HADRON COLLIDER (LHC) .....	35
FIGUR 6: AVDELINGER PÅ CERN .....	36
FIGUR 7: CERN DATASENTER .....	37
FIGUR 8: QUATTOR ARKITEKTUR (FORENKLET VERSJON) .....	41
FIGUR 9: TABELL OVER NCM KOMPONENTER .....	41
FIGUR 10: ORACLE PROGRAMVAREPRODUKTER BRUKT PÅ CERN.....	44
FIGUR 11: SKJERMBILDE FRA DATABASE INSTALLASJON .....	46
FIGUR 12: SKJERMBILDE AV KADU KADU .....	52
FIGUR 13: OVERSIKT OVER MØTER .....	55
FIGUR 14: BILDE FRA KAFFEPAUSE .....	56
FIGUR 15: SKJERMBILDE AV GOOGLE SØK.....	57





# INNHold

<b>SAMMENDRAG</b> .....	<b>I</b>
<b>FORORD</b> .....	<b>III</b>
<b>FIGURLISTE</b> .....	<b>V</b>
<b>INNHold</b> .....	<b>VII</b>
<b>1 INNLEDNING</b> .....	<b>1</b>
1.1 PROBLEMSTILLING .....	2
1.2 BEGRENSNING AV OPPGAVEN .....	2
1.3 OPPGAVENS STRUKTUR .....	3
<b>2 TEORI</b> .....	<b>5</b>
2.1 KUNNSKAP .....	5
2.1.1 <i>Uformell og formell kunnskap</i> .....	5
2.1.2 <i>Kunnskapsspiralen</i> .....	6
2.1.3 <i>Kritikk</i> .....	7
2.1.4 <i>Kanonisk praksis</i> .....	10
2.1.5 <i>Ikke-kanonisk praksis</i> .....	10
2.2 PRAKSISFELLESKAP .....	11
2.3 KUNNSKAP VS DET Å KUNNE .....	12
2.4 KONTEKST .....	13
2.4.1 <i>Hodebårne (embrained) kontekstuelle komponenter</i> .....	13
2.4.2 <i>Kroppsbårne (embodied) kontekstuelle komponenter</i> .....	14
2.4.3 <i>Kulturbårne (encultured) kontekstuelle komponenter</i> .....	14
2.4.4 <i>Nedfelt (embedded) kontekstuelle komponenter</i> .....	15
2.4.5 <i>Symbolbårne (encoded) kontekstuelle komponenter</i> .....	15
2.5 OPEN SOURCE .....	17
2.6 IKT .....	17
<b>3 METODER</b> .....	<b>21</b>
3.1 FORSKNINGSMETODE .....	21
3.1.1 <i>Kvantitativ forskning</i> .....	21
3.1.2 <i>Kvalitativ forskning</i> .....	21

3.1.3	<i>Fixed design</i> .....	22
3.1.4	<i>Flexible design</i> .....	22
3.2	DATAINNSAMLING:.....	23
3.2.1	<i>Observasjon</i> .....	23
3.2.2	<i>Intervju</i> .....	24
3.2.3	<i>Loggbok/Dagbok</i> .....	26
3.2.4	<i>Dokumentlesing</i> .....	27
3.3	MIN FORUTINNTATTHET .....	27
3.4	MULIGE FEILKILDER OG SVAKHETER I MIN FREMSTILLING.....	28
3.5	KLEIN OG MYERS PRINSIPP .....	28
3.5.1	<i>Det fundamentale prinsipp om den hermeneutiske sirkel</i> .....	28
3.5.2	<i>Kontekstualisering</i> .....	29
3.5.3	<i>Interaksjon mellom forsker og subjekt</i> .....	30
3.5.4	<i>Abstraksjon og generalisering</i> .....	31
3.5.5	<i>Multiple tolkninger</i> .....	31
3.5.6	<i>Mistenksomhet</i> .....	32
<b>4</b>	<b>BAKGRUNN .....</b>	<b>33</b>
4.1	CERN.....	33
4.1.1	<i>Forskning</i> .....	34
4.1.2	<i>Det tekniske studentprogrammet</i> .....	35
4.1.3	<i>Organisasjon</i> .....	36
4.1.4	<i>IT-avdelingen</i> .....	36
4.1.5	<i>ELFMS (Extremely Large Fabric Management Software)</i> .....	37
4.1.6	<i>RPM Package Manager</i> .....	42
4.1.7	<i>Oracle databaseprogramvare</i> .....	43
4.2	MIN ARBEIDSOPPGAVE .....	44
<b>5</b>	<b>RESULTAT .....</b>	<b>45</b>
5.1	ARBEIDET .....	45
5.1.1	<i>Alternativer</i> .....	46
5.2	OBSERVASJONER .....	49
5.2.1	<i>Misforståelser</i> .....	49
5.2.2	<i>Språk og Kulturforskjeller</i> .....	49

5.2.3	<i>Innføringen av et nytt kunnskapsystem</i>	50
5.2.4	<i>Konkurransen</i>	50
5.2.5	<i>Det polske nettverket</i>	51
5.2.6	<i>Epost</i>	52
5.2.7	<i>TWIKI</i>	53
5.2.8	<i>Møter</i>	54
5.2.9	<i>Kaffepauser</i>	55
5.2.10	<i>Google</i>	56
<b>6</b>	<b>ANALYSE OG DISKUSJON</b>	<b>59</b>
6.1	DISTRIBUSJON AV KUNNSKAP PÅ CERN	59
6.1.1	<i>Praksisfellesskap</i>	59
6.1.2	<i>Seksjonsgruppen</i>	60
6.1.3	<i>Fortellinger</i>	62
6.1.4	<i>Kontekst</i>	63
6.1.5	<i>Epistemiske kulturer</i>	65
6.2	IT SYSTEMERS ROLLE FOR Å STØTTE DISTRIBUSJON AV KUNNSKAP	66
6.2.1	<i>Maktforhold</i>	66
6.2.2	<i>Nytteverdi</i>	67
6.2.3	<i>Kontekst</i>	68
6.2.4	<i>Problemet med å samle all kunnskap i et system</i>	68
<b>7</b>	<b>AVSLUTNING</b>	<b>77</b>
<b>8</b>	<b>KILDER</b>	<b>79</b>



# 1 INNLEDNING

I filmen *Modern Times* med Charlie Chaplin får vi et inntrykk av den "nye tid" med effektiv samlebåndsproduksjon hvor hver enkelt arbeider bidrar med sin lille ting i produksjonen. Mye har endret seg siden den gang. I dag har man gått flere steg videre, med roboter på samlebåndet og kunnskap som den største konkurransefordelen for en organisasjon.

I det post-industrielle samfunnet som vi lever i er kunnskap ikke bare blitt en ressurs for organisasjoner - men en absolutt nødvendighet. Organisasjoners største konkurransefortrinn ligger i hvordan de håndterer kunnskapen og støtter opp om innovasjon i organisasjonen. Organisasjoner blir mer distribuerte, for eksempel på grunn av økonomisk lønnsomhet ved å sette produksjonen ut til andre land, eller at man må søke til andre land for å finne kompetansen man trenger. Også stadig mer avansert teknologi, og store prosjekter krever tverfaglige samarbeid og gjennom dette kommer det opp nye utfordringer i håndteringen av kunnskap. Distribuerte organisasjoner med samarbeid på tvers av heterogene nettverk er med på å forsterke dette.

Tidligere har kunnskapsfeltet vært dominert av synet om å omdanne uformell kunnskap til formell. Det vil si gjøre kunnskapen "forståelig" i form av kodifiserte uttrykk. Problemet med en slik fremgangsmåte er at man samtidig som å kodifisere kunnskapen mister en del av den opprinnelige konteksten rundt og dermed meningen.

Teorien til Nonaka og Takeuchi (Nonaka & Takeuchi, 1998) er en av de mest kjente innen kunnskapsforvaltning. Denne deler opp kunnskap i to hoveddeler, uformell (tacit) og formell kunnskap. Teorien beskriver et rammeverk om hvordan kunnskap blir omdannet i en spiralprosess med fire faser. Nonaka og Takeuchi har også blitt kritisert for denne fremstillingen, blant annet av Walsham (1996), Day (2005) og Tsoukas (2005).

### **1.1 Problemstilling**

LHC (Large Hadron Collider), verdens største partikkelakselerator nærmer seg ferdigstillelse, og foreløpig er det planlagt å starte den opp i midten av 2008. Forskningsprosjektet kommer til å generere enorme mengder data som må lagres, distribueres og analyseres. IT avdelingen på CERN har i hovedsak ansvaret for å skaffe til veie infrastruktur og lagringsplass for disse dataene. Dette gir en utfordring med tanke på alle tjenere, lagringsløsninger, etc som må administreres. Med et antall på over 4000 noder for øyeblikket, og et stadig økende antall vil det kreves enorme mengder manuell arbeidskraft hvis ikke administrering av disse gjøres mest mulig effektiv.

Denne oppgaven vil se på prosessen med å lage en automatisk installasjon av databaseprogramvare, for å minimalisere den manuelle innsatsen som kreves for å sette opp et databasesystem. Selve produktet har ikke fått så mye fokus i oppgaven, da det er prosessen som er det egentlige studiet. Jeg ønsker i denne oppgaven å se på hvordan kunnskap blir distribuert i høyteknologiske organisasjoner som CERN. På bakgrunn av dette ønsker jeg å se på hvilken rolle IKT systemer skal ha i utveksling av kunnskap i organisasjoner.

Problemstillingen blir da todelt, med første del: Hvordan blir kunnskap distribuert i IT avdelingen på CERN?

Og som andre del med bakgrunn av svarene på første problemstilling: Hva har dette å si for IT-systemer sin rolle for å støtte distribusjon av kunnskap?

### **1.2 Begrensning av oppgaven**

I en forskningsorganisasjon med over 2500 fast ansatte, og flere tusen besøkende fra over 80 nasjoner – gir det enorme muligheter for hva som kan studeres.

Da en masteroppgave må gjennomføres innen relativt kort tid, har jeg valgt å fokusere på en del av de problemstillingene jeg kunne ha tatt opp. Av praktiske grunner har jeg tatt utgangspunkt i SIS seksjonen i IT avdelingen, fordi det er der jeg har vært utplassert som teknisk student. Jeg har også tatt med hendelser og informasjon fra de andre seksjonene i

gruppen der det har vært naturlig. Noen ganger har jeg sett på hele organisasjonen som helhet for å få et totalbilde. Det kunne også vært interessant å gjøre en større sammenligning av miljøet i IT avdelingen mot de andre avdelingene som er mer fysikkrettet. Dette har jeg ikke fått tid til, men jeg gir en kort sammenligning mellom mine egne og Karin Knorr Cetina (1999) sine observasjoner av fysikkmiljøet på CERN. Under observasjonstiden var min hovedfokus å se på hvordan kunnskap blir distribuert innad i organisasjonen, selv om det hadde vært veldig spennende å se på den store kunnskapsutvekslingen CERN har med andre fysikkmiljø rundt om i verden. Av flere grunner har jeg lagt mer vekt på observasjoner, og mindre vekt på intervjuer. Dette vil jeg komme tilbake til i metodedelen.

### **1.3 Oppgavens struktur**

- Kapittel 2 Teori: Her vil jeg gi en oversikt over teori som danner et rammeverk for det jeg senere skal diskutere i oppgaven. Først starter jeg med en oversikt over uformell og formell kunnskap, og beskriver noe av arbeidet til Nonaka og Takeuchi og deres kunnskapsspiral (Nonaka & Takeuchi, 1998). Så viser jeg til kritikken av Nonaka og Takeuchi, og teorier om kunnskap som sosial prosess som fører til presentasjon av teorier rundt praksisfellesskap. Deretter blir Thompson og Walsham (2004) sin teori om kunnskap og kontekstuelle komponenter gjennomgått. Praksisfellesskap, og IKT i sammenheng med kunnskapsdistribusjon blir belyst i slutten av kapitlet.
- Kapittel 3 Metode: Dette kapitlet starter med en oversikt over forskningsmetoder, for så å beskrive min egen metode som er fortolkende casestudie. Videre beskriver jeg hvilke kilder jeg har brukt i datainnsamlingen, og begrunnelse for bruk av disse. Til slutt setter jeg et kritisk blikk på min forskning ved å bruke Klein og Myers sine prinsipper for fortolkende feltstudie.
- Kapittel 4 Case: Jeg gir en introduksjon til CERN og IT avdelingen, og en grunnleggende forklaring på teknologien og systemene som jeg har jobbet med under min tid som teknisk student. Dette gjør jeg for at leseren lettere skal kunne forstå de senere samtalene og problemstillingene som presenteres. Mitt arbeidsoppdrag blir så forklart.

- Kapittel 5 Resultat: Her blir resultatene av mine observasjoner og undersøkelser presentert. Resultatene blir forklart sammen med gjennomgangen av arbeidsgruppens fremdrift.
- Kapittel 6 Analyse og diskusjon: På bakgrunn av resultatene og teori blir det her en analyse og diskusjon.
- Kapittel 7 Avslutning: Dette kapitlet oppsummerer mine funn, og kommer med en konklusjon.



## 2 TEORI

### 2.1 Kunnskap

Hvordan man skal definere kunnskap har vært et emne for debatt helt siden antikken og Platon sin tid. Selv om det ikke finnes noen definitiv definisjon på hva kunnskap er, så finnes det en del holdepunkter og teorier.

Platon definerte kunnskap som "justified true beliefs". Ordene viten, lærdom og innsikt blir også knyttet sterkt opp mot kunnskap. Data og informasjon blir også satt i sammenheng med kunnskap.

#### 2.1.1 Uformell og formell kunnskap

I deres publikasjon deler Nonaka og Takeuchi (Nonaka & Takeuchi, 1998) kunnskap opp i formell (explicit) og uformell<sup>1</sup> (tacit) kunnskap. Formell kunnskap er i følge artikkelen en uttrykt kunnskap som er knyttet opp mot kodifisert vitenbasert kompetanse. Den blir sett på som universell og tilgjengeligheten er allmenn. Eksempler på formell kunnskap finnes over alt i samfunnet. Det kan være manualen til din nye sykkel, en oppskrift i en kokebok, en rutetabell over busstider etc.

Nærmest som en motpol til den formelle kunnskapen presenteres den uformelle kunnskapen. Denne karakteriseres som folkelig, og er mye mer personlig enn den formelle. Forfatterne knytter den ofte opp mot utøvelsen av et fag eller yrke.

Bakgrunnen for artikkelen er at Nonaka og Takeuchi ønsket å se på hvorfor japanske organisasjoner var så mye mer suksessfulle i forhold til vestlige land. Dette var beskrevet som en gåte for vestlige organisasjoner, da organisasjonene på utsiden ikke ble sett på som mer effektive. Nonaka og Takeuchi mener at den store forskjellen ligger i hvordan japanske organisasjoner behandler uformell kunnskap. Mens vestlige organisasjoner fokuserer på formell kunnskap som det viktigste, fremhever japanske organisasjoner uformell kunnskap

---

<sup>1</sup> Tacit knowledge blir også kalt skjult kunnskap

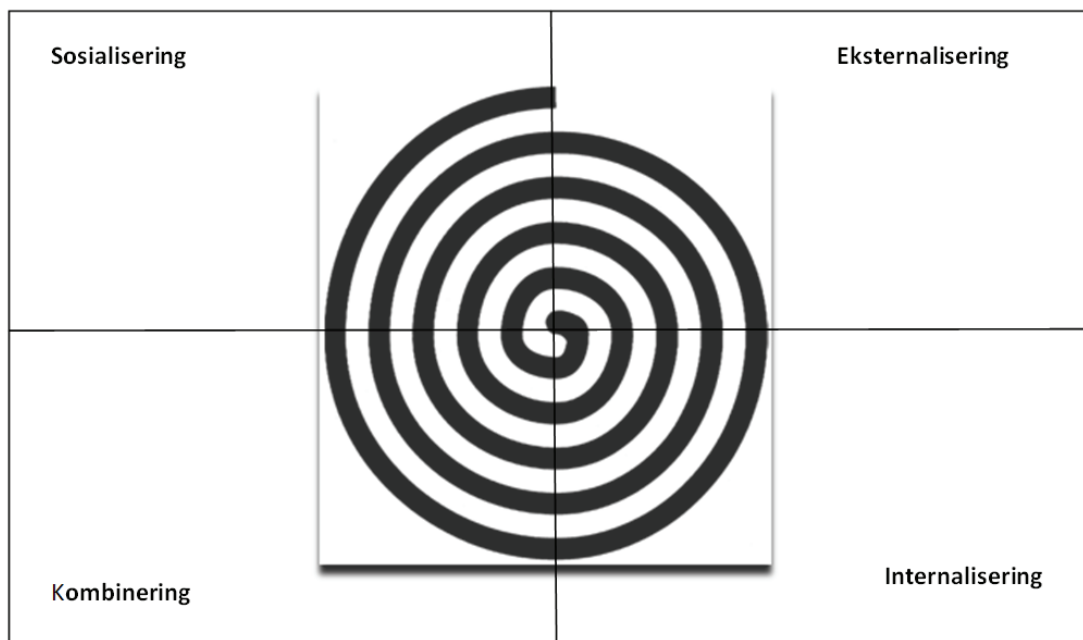
som like viktig. De mener at den formelle kunnskapen i noen tilfeller bare er "toppen av isfjellet" når man ser på den totale kunnskapen i en organisasjon.

### 2.1.2 Kunnskapsspiralen

Nonaka og Takeuchi (Nonaka & Takeuchi, 1998) presenterer et rammeverk for overføring mellom forskjellige typer kunnskap. Det hele beskrives som en kunnskapsspiral med fire forskjellige faser der kunnskap blir overført gjennom; Sosialisering, eksternalisering, kombinerings og internalisering.

- Sosialisering beskriver prosessen der uformell kunnskap i form av erfaring eller mentale modeller overføres fra person til person gjennom erfaringsdeling. Et eksempel kan være når en håndverker lærer opp en lærling gjennom praksis. Den kunnskapen som håndverkeren har er noe han har opparbeidet seg gjennom flere år, og dette er svært vanskelig for han å kodifisere dette i en lærebok. Mange av de teknikkene han har lært seg opp gjennom årene utføres uten å tenke noe over hvordan man gjør de.
- Eksternalisering som prosess omdanner uformell kunnskap til formell. Man prøver med andre ord å omdanne den uformelle kunnskapen til et representert format. Et eksempel på denne omdannelsen er å skrive ned, dvs. sette ord på den uformelle kunnskapen man har slik at den blir formell. Da det kan være veldig vanskelig å finne eksakte ord for å forklare kunnskapen, så kan det brukes analogier, metaforer, hypoteser, modeller etc. til å formidle denne.
- Kombinerings systematiserer og kombinerer konsepter inn i et kunnskapssystem. Prosessen blir omtalt som "omdanning" fra uformell til formell kunnskap. For eksempel kan det gjøres ved å kombinere kunnskap gjennom media som dokumenter, møter, telefonsamtaler eller IKT baserte løsninger. Å kombinere kunnskapen kan også skape ny kunnskap.
- I Internalisering inkorporerer man formell kunnskap til uformell. Det er nært knyttet til å få kunnskap via praksis (learning by doing). Erfaringer blir verdifulle gjennom

sosialisering, eksternalisering og kombinerer blir internalisert i form av delte mentale modeller eller teknisk "know how" hos den individuelle sin uformelle kunnskapsbase.



Figur 1: Kunnskapsspiralen til Nonaka og Takeuchi

### 2.1.3 Kritikk

Teorien til Nonaka og Takeuchi ble tatt i bruk av kunnskapsforvaltningsmiljøet<sup>2</sup>, og har hatt relativt stor innvirkning på dette. I den senere tid har teorien møtt en del kritikk av personer som mener at den bygger på feiltolkning av andre teorier (Tsoukas, 2005) (Walsham G. , 1996) (Day, 2005) (Brown & Duguid, 1991).

Haridimos Tsoukas (Tsoukas, 2005) viser til begrepet uformell kunnskap som Michael Polanyi (Polanyi, *Knowing and being: Essays by Michael Polanyi*, 1969) innførte, og hevder at Nonaka og Takeuchi gjør en feiltolkning av dette. I motsetning til Nonaka og Takeuchi så deler ikke Polanyi uformell og formell kunnskap opp i to forskjellige typer kunnskaper, men sier at kunnskap om noe består av både en uformell og en formell del.

<sup>2</sup> Engelsk: Knowledge Management

Hvis man ser på eksempelet med en busstabell som formell kunnskap, så avhenger det at man vet hvordan man skal lese denne, og at det faktisk er en busstabell som viser tidspunkt og steder der bussen stopper. Dette er kunnskap som enten gis sammen med tabellen når man får den, eller kognitiv kunnskap man har fra før.

Til tross for sin kritikk mot Nonaka og Takeuchi, mener Hourimodas at de har bidratt med å sette fokus på og fremheve viktigheten av uformell kunnskap, og hvordan kunnskap utvikles gjennom sosialisering. Men han mener samtidig at ideen om konvertering mellom uformell kunnskap til formell er uholdbar. I stedet er det viktig å se på hvordan kunnskap blir skapt gjennom sosialisering.

*"Tacit knowledge cannot be "captured", "translated", or "converted" but only displayed and manifested, in what we do. New knowledge comes about not when the tacit becomes explicit, but when our skilled performance is punctuated in new ways through social interaction".*

*Haridimos Tsoukas (Tsoukas, 2005)*

Tsoukas viser til at det alltid har vært behov for kunnskap i samfunnet, og særlig innen økonomien. Det som har skjedd i det post industrielle samfunnet er et mye mer fokus på den teoretiske kunnskapen, der det teoretiske står høyere enn det empiriske. Kunnskap blir kodifisert i store kunnskapsdatabaser, slik at man kan dra nytte av den teoretiske kunnskapen. Haridimos hevder at hvis man tar den teoretiske kunnskapen i nærmere øyesyn og ser på hvordan den blir brukt i praksis, vil man se at den ikke er så objektiv, selvstendig og eksplisitt som den ofte blir framstilt som. Den bygger på personlige avgjørelser og uformelle engasjement.

Ny kunnskap blir ikke til gjennom å omdanne uformell kunnskap til formell. Den blir skapt når man reflekterer på hvordan vi faktisk gjør de tingene og relaterer dette til omstendighetene rundt oss på en ny måte.

For å illustrere dette bruker Tsoukas eksempelet til Polanyi med et kart som en representasjon av formell kunnskap. Et Kart representerer et område, og kan sammenlignes med en samling regler for hvordan man skal komme seg fra punkt A til B. Men selv om man

har kartet, må man relatere det til virkeligheten. Kartet kan ikke navigere av seg selv, men er avhengig av at man vet hvor man er, hvor man skal og hvordan man vil gå. Alt dette krever beslutninger fra personen som relaterer kartet til virkeligheten basert på kognitive og følelsemessige midler.

Walsham (Walsham G. , 1996) retter også kritikk mot Nonaka og Takeuchi for at de behandler kunnskap som objekter som enkelt kan overføre til en annen person. Videre tar han for seg formell kunnskap, og hvordan denne er avhengig av den uformelle. I følge Walsham må den formelle kunnskapen "koble" seg til brukerens uformelle kunnskap. For å beskrive hvor kompleks kunnskapsoverføringen er, så bruker han begrepene "sensegiving" og "sensereading". Disse kommer fra Michael Polanyi, og beskriver hvordan en person gir fra seg kunnskap i form av "sensegiving" og hvordan mottakeren tolker dette ved "sensereading". Polanyi bruker en brevveksling mellom to personer som eksempel:

Person A er på besøk i et annet land for første gang. Han skriver et brev til person B og beskriver det han har sett og opplevd. Det som person A opplever tolker han i en "sensereading" prosess, og mens han skriver om dette i brevet til person B bruker han "sensegiving". Person B leser brevet og får et inntrykk av hva person A har opplevd. Når person B leser brevet gjør han en "sensereading". Polanyi peker på at for hver "sensereading"/"sensegiving" blir informasjonen forandret ettersom de to personene gjør forskjellige tolkninger.

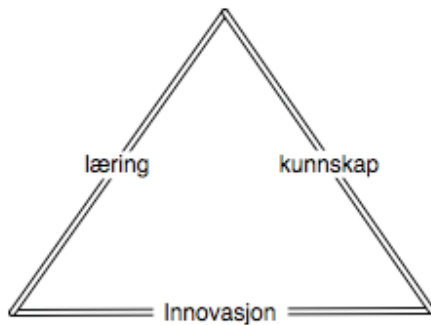
Robert E Day (Day, 2005) gjør det tydelig at Nonaka og Takeuchi har feiltolket Polanyi når det gjelder å dele kunnskap inn i to uavhengige deler, en formell og en uformell del. Han viser til Polanyi (1969, p. 195):

"The ideal of a strictly explicit knowledge is indeed self-contradictory; deprived of their tacit coefficients (personal to the individual), all spoken words, all formulae, all maps and graphs, are strictly meaningless"

Day sier at ironisk nok, så ble boken kreditert av Nonaka for å ha påvirket teorien hans om kunnskapsspiralen (Nonaka & Takeuchi, 1998) for omdanning av uformell kunnskap.

### 2.1.4 Kanonisk praksis

Brown og Duguid (Brown & Duguid, 1991) argumenterer for at det er en viktig sammenheng mellom arbeid (working), læring (learning) og nyskaping (innovating). De mener det er feil å behandle disse som tre separate komponenter, som altfor mange organisasjoner gjør. Forfatterne tar utgangspunkt i Julian E Orr (1996) sine studier av arbeidspraksisen hos serviceteknikere som reparerte Xerox kopimaskiner. Han viste at det var stor forskjell fra den praksisen ledelsen hadde beskrevet, og den praksisen som faktisk ble gjort i organisasjonen. Arbeidet som reparatørene skulle gjøre var beskrevet i manualer, opplæringskurs og jobbeskrivelser definert av ledelsen. Sammenlignet man dette med hvordan reparatørene jobbet, så var det store avvik.



Figur 2: Læring, kunnskap og innovasjon

Brown og Duguid hevder at denne store forskjellen i beskrevet og faktisk praksis, kan gjøre at en organisasjon ikke ser og verdsetter det verdifulle som medlemmene gjør. Det er disse praksisene som bestemmer om organisasjonen er en suksess eller fiasko. Videre går de til angrep på organisasjoner som prøver å bryte ned kompliserte oppgaver i mindre "skreddersydde" steg som kan utføres uten behov for nevneverdig forståelse eller innsikt i det som utføres. Brown og Duguid bruker ordet kanonisk praksis for å beskrive denne, i deres øyne ukultur der alt skal forenkles og operasjonaliseres.

### 2.1.5 Ikke-kanonisk praksis

Julian E Orr (Orr, 1996) beskriver situasjoner som oppstår der manualer og opplæring ikke vil hjelpe, da det ikke er beskrevet noen steder. Dette var kompliserte problemer med mange

avhengigheter, og ikke direkte knyttet opp mot feilkoder som var beskrevet i brukermanualene. I et slikt tilfelle hjalp det ikke med en kanonisk praksis hvis eneste forslag var å bytte ut maskinen. Reparatøren henvendte seg da til de andre reparatørene, og sammen startet de en sosial utveksling av erfaringer i form av å fortelle hverandre historier. Orr kalte dette for "krigshistorier" som hver enkelt reparatør hadde opplevd ute på "slagmarken". Gjennom fortellinger fikk alle deltakerne tilført nye erfaringer, som de igjen sammenlignet med sine egne og så denne kunnskapen i et nytt lys. Til slutt ga alle historiene, testene og erfaringene et grunnlag til å stille en diagnose og å reparere maskinen. Denne måten å jobbe på var i følge Orr svært vellykket.

## 2.2 Praksisfellesskap

Teorien om praksisfellesskap (engelsk: Community of Practice) fikk mye oppmerksomhet da Lave og Wenger publiserte et arbeid om situert læring og deltakelse (Lave og Wenger, 1991). Teorien hadde Etienne Wenger publisert i 1987 i et arbeid (Wenger, 1987) der hun så på lærlingtiden som et læresystem. Hun beskrev læringen i form av et triangulert forhold mellom eksperten, den uerfarne eksperten, og lærlingen. Det var ikke bare eksperten som bidro til at lærlingen utviklet ny kunnskap, men også kontakten med andre lærlinger og eksperter var nødvendige ingredienser.

I sin artikkel så Lave og Wenger på læring som en sosial deltakelse, og identitetsskapende for den enkelte. En person fikk kunnskap gjennom å være en aktiv deltager i sosiale fellesskap, og formet så sin identitet gjennom disse fellesskapene. På bakgrunn av dette ble teorien om praksisfellesskap publisert.

Et praksisfellesskap blir karakterisert som en samling av individuelle (eksperter og andre interesserte) som sammen deler kunnskap over et bestemt praksisområde de har til felles. De utvider og deler sin kunnskap og identitet gjennom å bidra til dette fellesskapet. Etienne Wenger definerer praksisfellesskap på følgende måte:

*"Communities of practice are groups of people who share a passion for something that they know how to do, and who interact regularly in order to learn how to do it"*

*better.”*

*Etienne Wenger (2004).*

Praksisfellesskap finner vi igjen på jobben, hjemme og på skolen. En familie kan for eksempel sees på som et praksisfellesskap. På bakgrunn av dette kan vi si at så å si alle personer er medlem av et eller flere praksisfellesskap.

Lave og Wenger bruker begrepet rettmessig periferisk deltagelse (RPD)<sup>3</sup> som referer til hvordan nykommere blir integrert inn i et praksisfellesskap. Etter hvert som nykommerne tilegner seg kunnskap, vokser deltagelsen deres fra å være perifer til å delta mer aktivt i kjernen av praksisfellesskapet. Etter hvert som det kommer nye medlemmer til vil størrelsen på praksisfellesskapet øke, og det kan bli vanskeligere for alle å delta i samme grad som før.

### **2.3 Kunnskap VS det å kunne**

Mange teoretikere innen kunnskapsfeltet skiller ganske skarpt mellom kunnskap (knowledge), og det å kunne (knowing). Kunnskap er for dem når man ser på den som et objekt, mens det å kunne er å se på selve prosessen som skaper kunnskap og det er dette som er den ekte kunnskapen i deres øyne. Jeg kommer ikke til å skille like mye, men jeg kommer til å sette fokus der kunnskap blir behandlet som objekter.

Stenmark (2002) argumenterer for at kunnskap er det som tidligere er omtalt som uformell kunnskap. Formell kunnskap er i følge Stenmark bare informasjon, og har liten verdi uten sammenhengen.

Ellingsen og Monteiro (2003) skriver om hvordan kunnskap skapes og opprettholdes i en organisasjon ved hjelp av interaksjon og samarbeid. De hevder at kunnskap er ikke noe som må ses på som statisk, men at den genereres gjennom aktivitet. Først når den gjentas, fortelles og holdes levende blir den relevant for andre. Kunnskapen må bli tolket, vurdert ut fra troverdighet og relevans og adapteres til den enkelte situasjon.

---

<sup>3</sup> Av det engelske uttrykket LPP (Legitimate Peripheral Partition)



## 2.4 Kontekst

Vi har alle hørt begrepet tatt ut av kontekst, der noen referer til at informasjon eller kunnskap har blitt tatt ut av den sammenhengen det hører til, og brukt slik at det får en annen betydning, eller ingen betydning i det hele tatt.

Thompson og Walsham, heretter kalt T&W tar for seg kontekst innen kunnskapsforvaltning i sin artikkel (Thompson & Walsham, 2004), der de kommer med forslag til organisasjoner for å støtte kunnskap i organisasjonen. I sitt studie bruker de Blackler (Blackler, 1995) sin typologi for forskjellige former kunnskap. Denne typologien deler kunnskapen inn i hva slags medium som er den aktive "bærer" av kunnskapen. Blackler bruker her fem bilder for dette; hodebåren (embrained), kroppsbåren (embodied), kulturbåren (encultured) og nedfelt (embedded) kunnskap. Disse har Blackler hentet fra Collins (1993), men har i tillegg skilt deler av Collins sin symbolbårne (encoded) kunnskap ut i en nedfelt (embedded) kunnskap. T&W redefinerer Blackler sin typologi og bruker den til å dele inn i kontekstuelle komponenter. Til sammen vil disse komponentene gi innsikt i relasjoner rundt kunnskap som prosess, og "avsløre" ingredienser til konteksten. Jeg vil her gå gjennom hver enkelt av disse inndelingene, og komme med eksempler på de forskjellige formene.

T&W advarer i samme artikkel om at kontekst er ingen stabil prosess, men endrer seg fortløpende. Derfor er det viktig å vedlikeholde konteksten i en organisasjon, og samtidig jo mer kontekstuell kunnskap som organisasjonen er kjent med, jo rikere er denne. De viser til at man bør prøve å få oversikt over relasjonssammenhenger for å en helhet.

### 2.4.1 Hodebårne (embrained) kontekstuelle komponenter

Blackler beskriver hodebåren kunnskap som kunnskap avhengig av konseptuelle ferdigheter eller kognitive evner (1995, p. 1023). Han beskriver hjernens struktur og evne til å utvikle komplekse regler og forstå komplekse årsaksforhold. Når dette så blir redefinert til en unik konfigurasjon av kontekst for prosessen å kunne, så passer denne delen godt til å fremheve at måten det å kunne består av er relasjonsmønstre av forskjellige kontekstdeler i hver person sitt sinn. Derfor vil anlegg og begavelse til den enkelte person kunne virke inn på relasjonsmønsteret av kontekst som blir skapt. Dette latente mentale potensialet kan bli sett

på som en kontekstuell komponent, siden det er en ingrediens til det å kunne, men alltid nevnt i forbindelse med de andre kontekstuelle formene.

### **2.4.2 Kroppsbårne (embodied) kontekstuelle komponenter**

Denne kategorien defineres av Blackler som handlingsorientert, og at den trolig er bare delvis eksplisitt uttrykt. Han tar utgangspunkt i Zuboff (1988), som mener at slik kunnskap er avhengig av personers fysiske tilstedeværelse, sansende og følelsebasert informasjon, fysiske spor og fysiske person til person diskusjoner (face-to-face discussions). Denne typen kunnskap oppnås gjennom handling, og har rotfeste i spesifikke kontekster.

Komponentene kan best oppfattes som fysiologiske filter og sensorimotoriske rutiner der hver person opprettholder fysisk kontakt med omverden. Selv om det finnes fysiologiske begrensninger, og at disse blir behandlet forskjellig av hjernen (jmf. hodebårne kontekstuelle komponenter), blir slike subjektive faktorer først aktivert i dialog med omverdenen. Fysiologiske kroppsbårne prosesser er derfor alltid knyttet opp mot relasjoner, på en slik måte at de blir til gjennom vår deltagelse i omverden, og er også farget av våre tidligere aktiviteter.

### **2.4.3 Kulturbårne (encultured) kontekstuelle komponenter**

T&W mener denne er den vanskeligste komponenten rent analytisk. Den blir definert som prosessen for å oppnå felles forståelse. Blackler viser til Orr (1996) sine studier av kopimaskinteknikere og Nonaka og Takeuchi (1998) sine kunnskapsorganisasjoner som eksempler på slik kunnskap. For å redefinere denne kategorien i en kontekstuell måte er det nødvendig å legge vekt på et annet aspekt: de historiske prosessene med sosialisering og kulturtilegnelse for å danne seg en felles forståelse. Samtidig må man sette fokus på at det er kunnskapsrike enkeltpersoner som oppnår en slik forståelse, og ikke grupper.

En kulturbåren forståelse oppnås gjennom en repetert sosial prosess, og er derfor relasjonsbasert sosialt fenomen. Selve forståelsen består av den samlede forventningen fra personene i den sosiale prosessen.

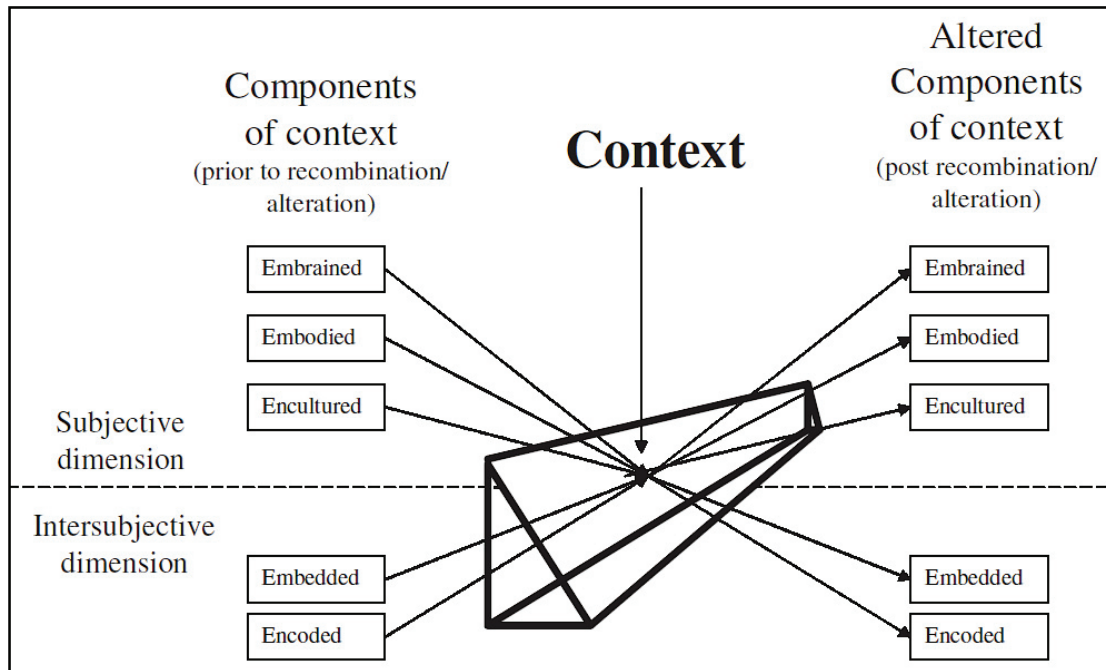
#### **2.4.4 Nedfelt (embedded) kontekstuelle komponenter**

Den fjerde kunnskapskategorien Blackler viser til er embedded kunnskap og defineres som "analyzable in systems terms, in the relationships between, for example, technologies, roles, formal procedures, and emergent routines" (1995,p. 1024). Dette viser til synlige organisatoriske sammensetninger, slik som foretningsprosesser som formes rundt intersubjektivt erfart kontekstuelle deler. Kategorien blir sett på som organisatoriske innbakte ingredienser til konteksten å kunne, og skiller mellom former kulturbårne deler (som finnes i hodet til personer) og relasjonsmønstre av eksplisitte organisasjonelle deler (teknologi, rutiner, prosedyrer, hierarki etc).

#### **2.4.5 Symbolbårne (encoded) kontekstuelle komponenter**

Dette er den siste og mest eksplisitte behandlingen av kunnskap, og blir av Blackler definert slik: "information conveyed by signs and symbols" (1995, p.1025). Da disse objektene blir sendt mellom personer i intersubjektiv kommunikasjon, så blir de ofte uriktig tatt for å være det faktiske målet for kommunikasjon og av det følger ordet kunnskap som betegner en delt form av verdier. Det er derimot hevdet at sett fra et aktivitetsstandpunkt, så er det ikke noe som er kunnskap, men heller en kontinuerlig prosess som dreier seg om det å kunne. Dette kommer fra den kontinuerlige "justeringen" av kontekstuelle komponenter som er aktivert av relasjoner.

T&W sier at på bakgrunn av disse komponentene så ser man at det å kunne ikke er sammensett av symbolsk interaksjon, men består av et unikt og situert relasjonsmønster av hodebårne, kroppsbårne, kulturbårne, nedfelte og symbolbårne deler av kontekst. Alle disse delene sammen gir en forklaring på hvorfor to personer aldri kan ha eksakt samme mening, da mening er utledet med relasjon til konteksten, og denne er igjen utledet fra en unik og midlertidig sammensetning av både delte og ikke-delte kontekstuelle deler som er generert i hodet til fysiologiske og biografiske unike enkeltindivider. På bakgrunn av dette kan aldri relasjonsmønstre være det samme. Symbolbårne kontekstuelle ingredienser er derfor meningsløse hvis de ikke blir satt i sammenheng med andre typer kontekst.



**Figur 3: Kontekstuelle komponenter**

Figuren ovenfor viser T&W sine fem kontekstuelle komponenter (basert på Blackler), og viser forholdet mellom dem. Konteksten blir sett på som et prisme der historiske erfaringskomponenter (delte og ikke delte) smelter sammen til en unik sammensetning til en spesiell erfaringspraksis. Etter hvert så kan en slik unik sammensmeltning ha historiske konsekvenser, da alle disse komponentene kan bli påvirket av interaksjon, for eksempel endring av minner, holdninger, kompetanse, rutiner eller skriftlig materiale. Disse endrede komponentene blir så kombinert med andre ingredienser som blir tilført neste "prismebilde". Figuren viser også tydelig at konteksten finnes i menneskets hjerne, der subjektiv og intersubjektive dimensjoner møtes.

## 2.5 Open Source

En studie (Gutwin et al, 2004) av hvordan kommunikasjonen foregår i store Open Source-programvareprosjekter gav interessant informasjon om hvordan group awareness ble ivaretatt gjennom prosjektene. Group awareness handler om at man har kunnskap om personene man jobber med, hva de jobber med og hvilke planer de har. Prosjektene som ble undersøkt var NetBSD, Apache httpd og Subversion. Resultatene var overraskende da det viste seg at deltakerne i prosjektet kunne jobbe uten å dele opp koden, noe som krever god oversikt over hvem som jobber med hva. Felles åpne epostlister var det som ble mest brukt for å diskutere og koordinere programmeringen. CVS-loggene ble også noe brukt for å se hvilke endringer som hadde blitt gjort. Instant messaging ble også noe brukt til kommunikasjon.

Toumi (2001) presenterer en kombinasjon av aktørnettverkteori og teori rundt community-of-practice for å forklare open source prosjekter. Arbeidet sees på som en økologi med sosio-teknisk utvikling, og ikke et prosjekt som implementerer en predefinert plan.

Et annet studie (A, Fielding, & Herbsleb, 2002) sammenlignet utviklingsarbeidet mellom to Open Source prosjekter, og hvordan de greide å organisere prosjektene.

Selv om Open Source prosjekter hevdes å være like effektive, eller til og med mer effektive enn tradisjonell programvareutvikling, trenger de noen koordineringsmekanismer når de kommer over en viss størrelse (se Mockus et al.).

## 2.6 IKT

Organisasjoner investerer store summer i teknologi, og da spesielt IKT for å bedre deling av kunnskap i organisasjonen, men er det virkelig så enkelt? Walsham (2001) ser på hva som gjør det vanskelig å implementere IKT-baserte kunnskapssystemer i organisasjoner, og hva man kan lære med tanke på videre arbeid med IKT som støtte til kunnskapsdistribusjon.

Et sitat fra Walsham (2001, p.600) viser at selv om systemet er tilgjengelig, er det ingen garanti for at personer vil bruke det:

*A manager of a multi-national company recently described his company's intranet to me as 'a large warehouse that nobody visits'. It seems that the 'explicit knowledge' contained on the intranet had not succeeded in connecting to the users' tacit world.*

Walsham understreker at en kunnskapsdatabase eller andre IKT baserte systemer bare er verktøy, og ikke løsningen alene for å dele kunnskap i organisasjonene. De må brukes til å støtte opp allerede eksisterende, eller nye sosiale nettverk der kunnskap utveksles. Generering og distribusjon av kunnskap kan ikke løses ved hjelp av teknologi alene. Utfordringen ligger i design av IKT baserte systemer, slik at disse støtter opp om det fellesskapet som er der fra før. En av utfordringene Walsham peker på er når fellesskapet har forskjellig språk

En annen utfordring som blir skissert er forholdet mellom kunnskap, makt og organisatorisk politikk. Walsham viser til at et IKT basert system som fra utsiden ser ut til å støtte uformell og formell kunnskapsutveksling, kan feile av andre moment enn det som går på selve systemet. Selv om systemet er veldig bra, kan det ligge andre organisatoriske grunner bak for at personer ikke vil ta det i bruk. Et eksempel Walsham bruker er en organisasjon som hadde resultatbasert belønningssystem. De ansatte så ingen fordel ved å dele sin kunnskap med andre, da det kunne gå direkte utover deres lønn.

Det ble også beskrevet et eksempel av et system for erfaringsdeling som fungerte veldig bra, men som de ansatte sluttet å bruke da sjefen deres fikk tilgang til systemet. Et system som

Konklusjonen til Walsham består av en rekke spørsmål som må stilles når en organisasjon skal opprette et kunnskapssystem.

Brown (Brown J. , 1998) foreslår nye måter å bruke IKT til for å støtte nye og eksisterende praksisfellesskap. Han peker på at IKT tradisjonelt har blitt brukt til å forsterke de gamle arbeids og læringsparadigmene som finnes i organisasjonene. Web teknologier bør brukes for å identifisere og anerkjenne praksisfellesskap, slik at utviklingen blir enda sterkere.

CSCW (Computer Suported Cooperative Work) er et relativt nytt felt, men det har en rekke interessante innfallsvinkler til det å formidle kunnskap i organisasjoner. Begrepet CSCW dukket i følge Grudin (1994) opp i 1984, og han mener at det var resultatet av et initiativ fra teknologer for å lære fra andre yrkesgrupper hvordan de samarbeidde i grupper. CSCW er betegnelsen på selve forskningsfeltet, mens Groupware er en samlebetegnelse på implementasjoner av teorien.

En rekke studier rundt innføring av CSCW systemer, og har prøvd å definere hva som ligger til grunn for at disse lykkes eller mislykkes. Orlikowski (1992) beskriver en rekke problemer rundt innføringen av et "groupware" produkt i en organisasjon. Innføringen skjedde i form av en top-down implementasjon, der de ansatte ble pålagt å bruke dette produktet. Ciborra (1996) tar blant annet for seg en innføring av et "groupware" produkt kalt MedNet i et farmasøytifirma. Heller ikke dette systemet ble noen suksess i perioden det ble studert. Forfatteren begrunner dette med en rekke startproblemer som mangel på kompetanse og teknologi, samt tekniske problemer.

Bowers (1994) gjorde en annen studie, og så på innføringen av et CSCW system i en organisasjon der personene allerede hadde noe kjennskap til CSCW fra før. Han ønsket på samme måte som Walsham å vise til problemer forbundet med CSCW systemer. Bowers trekker frem fem ulike aspekter rundt det å innføre et datastøttet samarbeid: tvetydighet, heterogenitet, forholdet mellom teknologi og organisasjon, maktforholdet mellom teknologi og organisasjon og til slutt produktivitetsparadokset ved innføring av ny teknologi (administrativ overhead). Dette viser at selv for organisasjoner som er godt kjent med hvordan IKT system for å støtte kunnskapsdeling fungerer, så kan det ikke garantere for at systemet blir en suksess for det. Det kan være mange momenter som hindrer dette.





## 3 METODER

I dette kapitlet vil jeg gå gjennom forskningsmetoden som jeg har brukt under min studie. Hvilke datainnsamlinger som ble brukt, og hvordan de ble gjennomført vil også bli belyst. Til slutt tar jeg en vurdering av egen forskning, og ser på den med et kritisk blikk. Dette gjøres hovedsakelig ved hjelp av Klein og Myers sine syv prinsipper for vurdering av fortolkende feltstudier.

### 3.1 Forskningsmetode

Innen informasjonssystem (IS) forskning finnes det to hovedretninger, gjerne omtalt som motpoler til hverandre; kvantitativ og kvalitativ forskning.

#### 3.1.1 Kvantitativ forskning

Kvantitativ forskning er den tradisjonelt vitenskaplige retningen, og var lenge dominerende innen IS forskning. Forskningen baserer seg på helt klare vitenskaplig objektive fakta, og man prøver å forklare sine funn ved hjelp av matematiske formler, modeller eller andre positive forklaringer som forklarer lovmessigheter. I resultatene blir det ofte fremstilt en rekke tall i form av tabeller og grafer.

Kvantitativ forskning dreier seg mye om det som kan måles, og settes en tydelig størrelse på.

#### 3.1.2 Kvalitativ forskning

I motsetning til kvantitativ forskning, har ikke kvalitativ samme fokus på at alt skal måles. Her prøver man å tolke og forstå fenomener i sammenheng med omgivelsene. Forskningen går ofte ut på å observere oppførsel, og gi en forklaring på denne. Man prøver ikke å forklare virkeligheten med matematiske modeller, eller en annen positivistisk sammenheng. Mye av IS forskningen kunne heller ikke vært forklart ved hjelp av kvantitativ forskning.

### 3.1.3 Fixed design

Fixed design har et ganske strengt syn på hvordan forskningen skal foregå med tanke på planlegging og gjennomføring. Den krever at man på forhånd spesifiserer en del om hva som det skal forskes på, og hvordan forskningen skal utføres. Dette bør spesifiseres før en går i gang med hoveddelen av forskningen. Ved bruk av fixed design i real world forskning finnes det i følge Colin Robson (Robson, 2002) en rekke krav for planlegging og utførelse. Han viser til Anastas og MacDonald (Anastas & MacDonald, 1994) som blant annet har krever at forskeren utvikler et konseptuelt rammeverk eller teori, slik at man vet hva man skal se etter. I tillegg må man gjennomføre omfattende pilotarbeid for å se hva som er gjennomførbart av "forskningsplanen" man har satt opp. De vil også at forskeren skal ha total kontroll over alle momenter i forsøket, og viser til spesifikasjoner om hva som skal gjøres for å utføre forskningen etter en profesjonell standard i form av en rekke instruksjoner og prosedyrer.

Kontrollerte eksperimenter og spørreundersøkelser er eksempler på fixed design. Selv om disse fortsatt blir mye brukt og gir svar på mange forsknings spørsmål, har de også møtt en viss kritikk. Problemet med å sette opp en teori eller konseptuelt rammeverk over det en skal studere, er at man i samfunnsforskning ikke kan vite alt om det feltet man skal til å studere. En annen ulempe med fixed design er at man ikke hele tiden vet hva man vil studere, og da ikke kan definere dette. Etter hver har det også blitt en del oppslutning til en annen tilnærming innen samfunnsforskning kalt kvalitativ design.

### 3.1.4 Flexible design

Anastas og MacDonald bruker benevnelsen flexible design istedenfor kvalitativ. Kvalitativ refererer til alle de kvalitative resultatene man får ved å bruke denne tilnærmingen, mens flexible viser til at forskningen ikke blir så predefinert, og man har det dermed mer fleksibelt med tanke på forskningsprosessen forandrer seg. Dessuten kan flexible design ta i bruk kvantitative metoder i tillegg til de kvalitative – og da blir det litt misvisende å navngi tilnærmingen kvalitativ design. Når det gjelder flexible design, så er den vanskeligere å definere enn fixed design, blant annet fordi det er relativt nylig at design forhold til denne

tilnærmingen har blitt tatt opp. Flexible design benytter seg av kvalitative forskningsmetoder, og har siden tidlig 1990 hatt en jevn økning i antall personer som har tatt den i bruk.

Siden fixed design bygger på kvantitative data og statistiske generaliseringer, så er det enkelt for deres tilhengere å vise til at denne tilnærmingen er vitenskapelig. Når det gjelder flexible design så er det fortsatt noen som debatterer om denne er vitenskapelig eller ikke. Blant annet finnes det personer som har brukt kvalitative metoder i sitt arbeid, og ønsker ikke at dette skal ses på som vitenskapelig. Colin Robson (Robson, 2002) argumenterer for at det finnes sterke argumenter for at både resultater av fixed design og flexible design kan ses som vitenskap. Betingelsen for dette er at arbeidet er utført på en systematisk og prinsipiell måte.

## **3.2 Datainnsamling**

Arbeidet er gjennomført som feltstudie. Datainnsamlingen er gjort over en periode på 9 mnd som teknisk student på CERN. I hovedsak har jeg fokusert på observasjoner som hovedkilde. Jeg har gjort fortløpende observasjoner, og skrevet dette ned i en notatblokk – for så å transkribere det videre senere.

I datainnsamlingen har jeg skrevet logger av observasjoner, tatt foto, samlet epostlister, logget lynmeldinger på Windows Live Messenger, gjort uformelle intervju, og til slutt gjennomført noen semistrukturerte intervju.

### **3.2.1 Observasjon**

Observasjon ble gjort fortløpende under hele studietiden. Da jeg var på CERN som teknisk student, måtte jeg ikke forklare til personer om hvorfor jeg var med på møter, forelesninger osv. Jeg fikk en utmerket anledning til å studere alle rundt meg uten at noen stilte spørsmål om det jeg gjorde. Allerede første dagen ble jeg introdusert for alle i seksjonsgruppen, og jeg ble fort integrert i gruppen.

Det at jeg hadde "stillingen" som teknisk student, og skulle jobbe med automatisering av tjenester i tillegg til å studere de rundt meg var til tider en utfordring. Samtidig gjorde dette at jeg virkelig fikk en ekte opplevelse om hvordan det var å jobbe i seksjonen.

Observasjoner som jeg gjorde ble ført inn i en loggbok etter hvert. Jeg hadde ikke mulighet til å gå rundt å skrive notater åpenlyst, da det kunne ha skapt mistenksomhet om hva jeg skrev om.

Før jeg begynte med observasjonen vurderte jeg det etiske aspektet med å observere personer uten at de selv var klar over det. Som et resultat av dette har jeg i ettertid spurt personer som jeg skal referere fra om å få bruke deres utsagn. I tillegg har jeg ikke nevnt noen navn i oppgaven, og siden det er snakk om en stor organisasjon er det vanskelig å identifisere situasjoner og personer som viser til i oppgaven. Jeg har også brukt vignetter for å illustrere situasjoner som ligner på de som jeg opplevde. Hovedgrunnen til at jeg valgte å være anonym mens observasjonene foregikk var for å studere personene i deres naturlige miljø. Dersom de hadde visst at jeg observerte dem, kunne det vært vanskeligere for meg å få informasjon. I tillegg er det også en risiko for at personer hadde promotert sine personlige agendaer og gitt meg feilinformasjon.

### 3.2.2 Intervju

Intervju er mye brukt som forskningsmetode i samfunnsforskning. Avhengig av hvor detaljerte de er, så kan de deles inn i to hovedkategorier: strukturert- og ustrukturert intervju.

Et strukturert intervju grenser mot en ren spørreundersøkelse. I disse intervjuene er alle spørsmål definert på forhånd, og stilles i en bestemt rekkefølge i et standardisert språk. I tillegg er spørsmålene ofte laget slik at man har begrensede svaralternativer, slik at de er enkelt å sammenligne med andre svar med tanke på statistikk.

I et ustrukturert intervju har som regel ikke intervjueren definert alle spørsmålene på forhånd, men satt opp noen veiledende spørsmål. Det er også mulig for intervjueren å forandre spørsmålet eller komme med oppfølgingsspørsmål underveis i intervjuet. Dette kommer av at

intervjueren hører på svarene som blir gitt, og tilpasser intervjuet til intervjuobjektet. Denne type intervju blir mest brukt i sosiologi.

Innen kvalitativ forskning brukes det som regel en mellomting mellom strukturerte og ustrukturerte intervju, ofte kalt semistrukturerte intervju. Her setter intervjueren opp en liste spørsmål eller tema som man prøver å holde seg til under intervjuet. Hvis intervjueren merker noe interessant under intervjuet, har han mulighet til å stille oppfølgingsspørsmål for å få en bedre forståelse.

Hoveddelen av intervjuene jeg har gjennomført har vært uformelle samtaler (ustrukturerte intervju) i kaffepauser eller lignende. Jeg har også gjennomført noen semistrukturerte dybdeintervju med nøkkelpersoner i min seksjon og i fysikkmiljøet på CERN, men dette er av begrenset omfang helt på slutten av min studieperiode. Grunnen til at jeg ikke gjennomførte strukturelle intervju før på slutten, var fordi det da ville blitt vanskelig å ikke avsløre seg selv for de andre som jeg har jobbet sammen med og samtidig observert. Som nevnt under observasjonsdelen, så har jeg ikke informert personer om min forskning, fordi jeg ville ha mest mulig realistiske observasjoner.

Uformelle intervju og samtaler ble gjennomført under hele studietiden. For personene i seksjonen min ble det kaffepausen som var hovedarenaen for disse samtalen. Av og til var det jeg som stilte noen spørsmål, og av og til andre personer. Disse kaffepausene var noe av de mest interessante, da det var enkelt å stille spørsmål til de andre. I lunsjen spiste jeg sammen med studenter fra andre avdelinger, og fikk dermed mulighet til å få en innsikt i deres miljø. Denne kunnskapen har jeg brukt i noen av resultatene mine.

Jeg hadde også en rekke samtaler med personer i andre posisjoner på CERN. Blant annet med en person Karin Knorr Cetina (Cetina, 1999) refererer til i sine studier av partikkelfysikerne på CERN.

Semistrukturerte intervju ble gjennomført på 4 personer i seksjonen jeg jobbet med, samt 2 personer fra andre avdelinger på CERN. Til disse intervjuene noterte jeg etter hvert som samtalen utløp seg. Under intervjuene prøvde jeg å holde samtalen til de spørsmålene som jeg stilte, for å holde samtalen innenfor tidsgrensen på 20 min for hvert intervju. Det viste seg

at flere av intervjuobjektene begynte å prate om hva de personlig hadde oppnådd, og hele samtalen kunne spore litt av. Dette regnet jeg som akseptabelt da intervjuet var semistrukturert, og jeg lot intervjuobjektet komme med sine meninger uhindret.

Intervjuene ble avholdt på forskjellige steder, men ikke på intervjuobjektens kontor. Dette var viktig for meg, da det lett kunne være forstyrrende elementer som telefonsamtaler, folk som kom innom, eller andre på kontoret som lurte på noe.

Intervjuene ble grovt transkribert innen en time etter at de hadde blitt holdt. I løpet av 1-2 dager ble de gått gjennom på nytt.

### 3.2.3 Loggbok/Dagbok

Loggboken har jeg først og fremst skrevet elektronisk, med tanke på sikkerhetskopiering og muligheten til å søke i etterkant. Boken har normalt blitt oppdatert to til tre ganger daglig, som regel rett etter observasjoner av møter eller når det har skjedd mye. I tillegg til den elektroniske loggføringen har jeg brukt en arbeidsbok under møter med mye informasjon, og i eventuelle lengre perioder uten tilgang til maskinen. Disse notatene har blitt ført inn i elektronisk etter hvert som jeg har fått tilgang til datamaskin igjen.

For å bedre kunne skille mellom observasjoner, mitt arbeid, og mine refleksjoner de forskjellige hendelsene brukte jeg en mal som delte inn i følgende deler:

- Hva jeg har gjort: Beskriver hva jeg har gjort på prosjektet den aktuelle perioden.
- Observasjoner: Beskriver mine observasjoner i den aktuelle perioden.
- Refleksjoner: Beskriver mine personlige refleksjoner rundt observasjoner og arbeid jeg har gjort.

I tillegg er det et felt på skjemaet for å markere interessante hendelser.

### 3.2.4 Dokumentlesing

Gjennom studiet har jeg hatt tilgang til en rekke elektroniske dokumenter, om mine arbeidsoppgaver og

Mobiltelefonen min hadde jeg hele tiden med meg, så derfor tok jeg en del bilder med denne av ulike observasjoner. Dette gjorde jeg for selv å huske på observasjonen når jeg senere skulle skrive i loggboken, og for å kunne bruke bildene i oppgaven for å bedre kunne gi leseren et bedre inntrykk av de forskjellige observasjonene.

Epostlistene jeg abonnerte på var følgende:

- Liste for seksjonen, gruppen, og IT-avdelingen
- Liste for ELFMS Quattor utviklingsprosjekt
- Liste for tekniske studenter, og norske tekniske studenter
- Liste for oracle database administratorer<sup>4</sup>

### 3.3 Min forutinntatthet

Selv om all forskning ideelt sett skal være helt objektiv, så er ikke dette alltid tilfelle. Spesielt innen kvalitativ forskning kan dette være et problem, da forskeren uten sin vilje og viten kan påvirke resultatet. Da kvalitativ forskning har mye fokus på det å tolke resultatene, så kan forskerens egne meninger farge forskningen, selv om hun/han ikke gjør dette bevisst. Da mine datainnsamling har fokusert mest på observasjon som hovedkilde, så er dette et kritisk punkt for min forskning.

For å redusere graden av påvirkning kan forskeren beskrive sin forutinntatthet før hun/han begynner observasjonen. Dette hjelper både forskeren, og personer som skal forstå forskningsresultatet – da det kan være mer tydelig for de hvordan forskeren har tenkt.

---

<sup>4</sup> [oracle-l@freelists.org](mailto:oracle-l@freelists.org) : offentlig tilgjengelig epost liste.

#### **3.4 Mulige feilkilder og svakheter i min fremstilling**

Det er viktig å være klar over at mine observasjoner distribusjon av dreier seg om kun et begrenset miljø, og kan derfor skille seg ut fra andre miljøer. Ideelt sett skulle jeg gjerne ha studert flere organisasjoner med forskjellige bakgrunner, men på grunn av tidsaspektet har dette ikke latt seg gjøre. Men på den annen side har mange av observasjonene jeg har gjort, stemt godt overens med miljøer beskrevet i artikler av blant andre Walsham (2004), Orr (1996),

#### **3.5 Klein og Myers prinsipper**

Klein og Myers (1999) beskriver i sin artikkel syv prinsipper for å utføre og evaluere fortolkende feltstudier. Disse prinsippene er bygget på forskning innen antropologi, fenomenologi og hermeneutikk. Prinsippene er den kvalitative forskningens svar, på metoder som blir benyttet i positivisk forskning innen IS (for eksempel Lee 1989). Klein og Myers (heretter kalt K&M) skriver selv at disse prinsippene er veiledende, og at man må bruke de prinsippene som er mest aktuelle for det forskningsprosjektet man driver med. På grunnlag av dette har jeg tatt utgangspunkt i følgende prinsipper

##### **3.5.1 Det fundamentale prinsipp om den hermeneutiske sirkel**

Dette er hovedprinsippet i fortolkende arbeid i en hermeneutisk sammenheng. Det er dette alle de etterfølgende prinsippene bygger på, og forsøker å utvide. Forståelse av feltstudiet oppnås ved å iterere mellom å se på meningen av deler og helheten av studiet. Ved å se på helheten bidrar dette til å bedre forståelse av delene, og vice versa. K&M definerer prinsippet på følgende måte :

*The process of interpretation moves from a precursory understanding of the parts to the whole and from a global understanding of the whole context back to an improved understanding of each part, i.e., the meanings of the words.*

*(Klein & Myers, 1999)*



K&M bruker analysen av en setning som eksempel på den hermeneutiske sirkel. Dersom man først ser på en bokstav eller et ord, er det begrenset med kunnskap man får ut av dette. Hvis man derimot fortsetter og ser på sammensetningen av bokstaver og ord som et hele, vil man med en gang ha en mye bedre forståelse når hver bokstav eller ord skal analyseres på nytt. Da er det mye lettere å forstå betydningen av hvert ord, eller hvorfor en bokstav er plassert på denne måten. Gjennom en slik iterasjon vil man danne seg en samlet fortolket forståelse.

*Å noen ganger kunne se på CERN som helhetlig organisasjon istedenfor seksjonen alene, har vært nødvendig både gjennom min forskning, og for å kunne gjennomføre mine daglige arbeidsoppgaver i seksjonen. For eksempel var det mange av observasjonene mine som jeg ikke forstod så mye av da de skjedde, men etter hvert som jeg fikk mer kunnskap og så det hele i en større sammenheng – så var de lettere å tolke. Mine arbeidsoppgaver måtte også ses på i et større perspektiv, da de ikke gav så mye mening når man så på disse som en isolert service.*

### **3.5.2 Kontekstualisering**

Konteksten kan påvirke forskningsresultatene i forskjellig grad, og dette er veldig viktig å være klar over. Prinsippet er i følge K&M basert på Gadamer's innsikt at det er en forskjell mellom hvordan leseren og forfatteren av en tekst tolker det som er skrevet. Dette prinsippet krever at forskeren må vise leserne i hvilken sosial og historisk kontekst forskningsprosjektet blir satt i. Kontekstualisme skiller seg fra positivistisk måte å se historisk kontekst på, ved at den aksepterer at organisasjonelle mønster endrer seg hele tiden. Positivistiske forskere mener derimot at endringer i organisasjonen går igjen etter et mønster. Så hvis man har kartlagt historiske endringer, vil disse gjenta seg. Ved feltarbeid blir forskerens arbeid påvirket av historien til organisasjonen, samtidig som forskeren også påvirker organisasjonen selv. K&M mener det er viktig at forskeren eksplisitt reflekterer over påvirkningen i resultatet av forskningsarbeidet.

Selv om en masteroppgave er relativt avgrenset i omfang, har jeg prøvd å sette mitt arbeid inn i den sosiale og historiske konteksten som arbeidet har foregått i. Likevel vil arbeidet bare gi et øyeblikksbilde av CERN i den tiden forskningen foregikk. I de neste årene som kommer vil

det bli store endringer på CERN innen organiseringen av ansatte, og dette kan forandre en del på situasjonen som er beskrevet i denne oppgaven.

#### **3.5.3 Interaksjon mellom forsker og subjekt**

I dette prinsippet blir det slått fast at forskningsdata ikke er noe som bare ligger å venter på at forskeren skal plukke det. Det kreves at forskeren setter seg selv og subjektene inn i et historisk perspektiv. Kunnskap skapes gjennom den sosiale interaksjonen mellom forsker(e) og subjekt(er). Videre må forskeren anerkjenne at de som skal studeres på lik linje med forskeren kan bli sett på som tolkere og analytikere.

Forskningssubjektene blir tolkere når de tilegner seg konsepter som blir brukt av IS forskerne (eller andre parter i prosjektet), som forandrer deres måte å se ting på. De blir analytikere dersom deres handlinger blir påvirket av den nye måten å se ting på. Et annet moment i dette prinsippet går på forskerens forutfattede meninger, og hvordan disse påvirker forskerens tolkninger av data. Det er derfor viktig at forskeren er klar over sine egne antagelser (assumptions) , og har evne til å stille spørsmål ved disse. Bakgrunnen til en forsker kan påvirke hvordan forskningen blir lagt opp, hvilke resultater som finnes og hvilke konklusjoner som trekkes. Selv om en forsker skal være nøytral når man går inn i studiet, så er dette så og si umulig. Ofte har man en ide om hva man som forsker forventer å finne, eller hva man ønsker å finne. Oppfatninger som forskeren har på forhånd, enten de er bevisst eller ubevisst kan komme fra erfaringer, kultur og lignende.

*Som teknisk student har det vært umulig for meg å ikke delta i diskusjoner, og komme med innspill på løsninger som diskuteres. Selv om jeg har hatt en mer avventende rolle i diskusjoner, så har jeg kommet med innlegg når jeg har følt det er forventet av meg. Walsham og Sahay (Walsham & Sahay, 1999) hevder at det er umulig for forskeren å ikke bli involvert i fortolkende studier som varer over lengre tid.*

*Når det gjelder hvilke forventninger og hvilke oppfatninger jeg hadde før jeg startet å observere, så har jeg tatt opp dette i en egen del under metoder kalt forutinntatthet<sup>5</sup>.*

### **3.5.4 Abstraksjon og generalisering**

Fortolkende forskning er mistenksom ovenfor positivistiske naturlover som sier at menneskelig oppførsel kan beskrives som en kausal sammenheng, uavhengig av kultur. Likevel hevdes det at det finnes en filosofisk basis for abstraksjon og generalisering i fortolkende feltstudier. Utfordringen til forskeren er å relatere unike resultater og hendelser til ideer og konsepter som kan brukes på flere situasjoner. K&M refererer til Walsham som sier det finnes fire typer generaliseringer fra fortolkende case studier:

1. Utvikling av konsepter
2. Generering av teori
3. Trekking av spesifikke implikasjoner
4. Bidrag til utvidet innsikt

Selv om jeg ikke har generert noen ny teori, så har oppgaven satt fokus på utfordringer i forbindelse med kunnskapsspredning i organisasjoner. Disse utfordringene er abstrahert og generalisert fra resultater av observasjon og fra eksisterende teorier.

### **3.5.5 Multiple tolkninger**

Hvordan personer oppfatter en hendelse kan være forskjellig fra person til person. Selv om disse personene har forskjellige oppfatninger om hva som har hendt, kan alle ha rett hvis man ser det fra deres syn.

Dette punktet krever at forskeren må undersøke hvilken påvirkning den sosiale konteksten har under studiet. Dette gjøres ved å dokumentere forskjellige synspunkt, og hva som er

---

<sup>5</sup> Av engelsk prejudice.

grunnene til disse. Dokumentasjonen kan inneholde forståelse av konflikter relatert til maktforhold, penger eller verdier.

#### **3.5.6 Mistenksomhet**

Forskeren må ikke være ukritisk og godta alle resultater som kommer frem under forskningen. Mens de i stedet bør man se på sosiale relasjoner, og hvordan disse henger sammen. For eksempel kan personer gi uriktige data under et intervju, dersom det gir fordeler. Selv om det er en generell konsensus i noe, kan det være grunn til å stille spørsmålstejn. Da dette prinsippet går utenfor tolkning av data, er det uenighet om hvor langt man bør drive det.

I løpet av studiet har jeg hele tiden vært oppmerksom på at det folk forteller meg er deres versjon av det de vil at jeg skal høre, og jeg har derfor ikke vært helt ukritisk til dette. Samtidig mener jeg at personer som jeg har hatt samtaler med ikke har hatt noen sterke personlige agendaer slik at de har foret meg med feilinformasjon. Jeg har måttet ta hensyn til at folk har hatt flere meninger, men dette er relativt vanlig i en organisasjon med forskjellige individer. I blant har jeg sjekket om informasjon som personer har fortalt til meg stemmer.

## 4 BAKGRUNN

Her vil jeg gi en introduksjon til CERN, IT-avdelingen og teknisk bakgrunnsmateriell. Dette gjøres for at leseren lettere skal forstå miljøet på CERN, og den tekniske delen i arbeidsprosessen.

### 4.1 CERN

CERN (Conseil Européen pour la Recherche Nucléaire) er den europeiske organisasjonen for atomforskning. Selve navnet atomforskning er litt misvisende, da CERN gikk over til å forske på partikler for flere tiår siden. I dag har organisasjonen verdens største partikkelfysikk-laboratorium. Den ble grunnlagt i 1954 av 12 medlemsland (derav Norge), og var et av de første fellesprosjektene i Europa. I dag teller organisasjonen 20 medlemsland i tillegg til å ha en rekke samarbeid med andre land, UNESCO og EU som begge har observatørplass.

CERN har hovedsete på grensen mellom Frankrike og Sveits vest fra Genève ved foten av Jurafjellene. Organisasjonen holder hovedsakelig på med høyenergi partikkelfysikk, og har omtrent 2500 fast ansatte. I tillegg kan det være over 7000 besøkende, alt fra studenter, gjesteforskere og eksterne samarbeidspartnere som jobber på oppdrag fra CERN. Til sammenrepresenterer vitenskapsfolkene som er tilknyttet CERN over 500 universiteter og 80 nasjoner.



Figur 4: Medlemsland tilknyttet CERN

### 4.1.1 Forskning

Hovedoppgaven til CERN i dag er å drive forskning innen partikkelfysikk, og finne svar på spørsmål som:

- Hvordan så universet ut rett etter the Big Bang?
- Hva er materie bygd opp av?

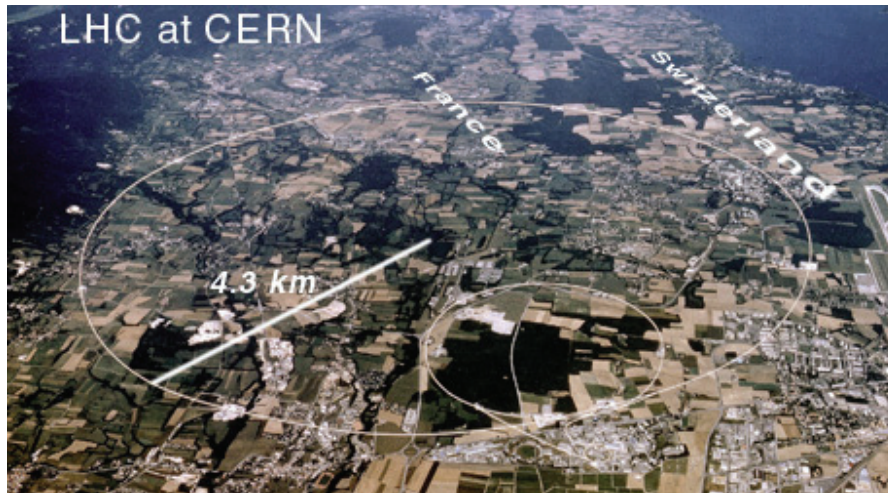
I jakten på stadig mindre partikler brukes partikkelakseleratorer til å akselerere partikler opp i en fart nær lysets hastighet, for så å kollidere dem med hverandre. Ved å analysere hva som skjer i kollisjonsøyeblikket og rett etter kan forskere få svar på hva som skjedde etter the Big Bang, og dermed finne ut mer hvordan alt rundt oss er bygd opp.

#### ***4.1.1.1 LHC verdens største vitenskapelige instrument***

Innen fysikken beskriver standardmodellen all kjent form av materie og hvordan de samhandler med hverandre. Alt dette er observert i eksperimenter foruten Higgs boson, en partikkel som kan kaste lys over spørsmålet om hva masse egentlig er.

Siden 1994 har CERN jobbet med et prosjekt kalt LHC (Large Hadron Collider), en partikkelakselerator som ligger 100 meter under jorden og har en omkrets på hele 27 kilometer. Dette gjør LHC til verdens største fysikkprosjekt, verdens største maskin og verdens største kryogeniske laboratorium. Alle disse rekordene sammen med antall personer som er involvert i prosjektet vitner om hvilket enormt arbeid som er gjort, og fortsatt gjøres i konstruksjonen av LHC. Opprinnelig var planen at prosjektet skulle vært ferdig i 2005, men tekniske og finansielle utfordringer har satt den foreløpige dato til ferdigstilling og testing i slutten av 2007, og med endelig oppstart planlagt i midten av 2008.

I et miljø der så mange kunnskapsrike personer fra alle mulige forskjellige fagmiljø og kulturer skal samarbeide genereres det og utveksles enormt mye kunnskap, og samtidig skapes det mye innovasjon.



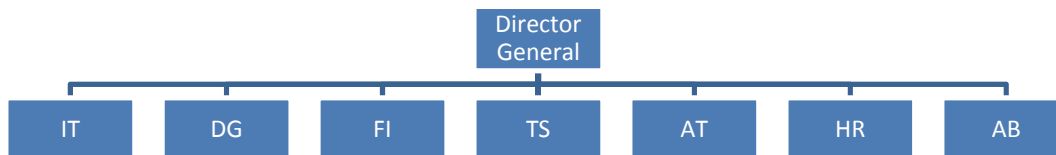
Figur 5: Large Hadron Collider (LHC)

Et bevis på dette er alle systemene som har dukket opp i kjølvannet av de forskjellige eksperimentene på CERN. For eksempel ble World Wide Web unnfanget som en følge behovet for utveksling av kunnskap mellom avdelingene på CERN, og fysikkmiljøet generelt. En litt nyere teknologi som CERN har ført frem utviklingen av er GRID, ideen om distribuert datakraft som for brukeren skal være like abstrakt som levering av strøm i stikkontakten hjemme. Det vil si at brukeren slipper å tenke på hvem som leverer regnekraften, bare hva han skal bruke det til.

#### 4.1.2 Det tekniske studentprogrammet

Det tekniske studentprogrammet er et tilbud CERN i samarbeid med hvert enkelt medlemsland gir studenter som ønsker å skrive sin masteroppgave i et spennende høyteknologisk miljø. Studentprogrammet gir mulighet for opphold på et halvt år eller ett års varighet. Hvert år tar CERN inn en rekke studenter fra forskjellige land. Samtidig som studenten gjør oppgaver for CERN får han/hun tilført verdifull erfaring, og kunnskap i form av kurs, presentasjoner og deltagelse i arbeidsgrupper. Dette er en del av CERN sin strategi for gi noe tilbake til medlemslandene og samfunnet ellers.

### 4.1.3 Organisasjon



Figur 6: Avdelinger på CERN

Til sammen jobber det rundt 2500 fast ansatte på CERN. Totalt er det syv forskjellige avdelinger på CERN. Hver avdeling er delt inn i grupper av spesialisering, og videre inn i seksjoner. IT-avdelingen består av 12 grupper, og 40 seksjoner. Jeg har vært utplassert i seksjonen for System Infrastructure Services (SIS) som ligger i Database and Engineering Services (DES) gruppen. Totalt i seksjonen var det (5 ansatte og 2 studenter), mens det i gruppen var 40 personer. I tillegg til SIS består DES av Database Infrastructure Services seksjonen og seksjonen for Engineering tools. Hovedfokuset på gruppen er databasetjenester og støtte til tekniske programvareverktøy som AutoCad og andre spesialverktøy.

### 4.1.4 IT-avdelingen

Som nevnt tidligere ligger CERN både på den franske og sveitsiske delen av grensen. Den franske avdelingen ligger i Preessin, og den sveitsiske i Meyrin. Det er på denne delen IT-avdelingen ligger og jeg har hatt mitt daglige arbeid. IT-avdelingen er fordelt over to bygninger, der den ene også inneholder datasenteret. Som de fleste andre bygninger på CERN er de preget av å ha blitt bygget for godt over 40 år siden. Bygningen der jeg har tilbrakt mesteparten av tiden, består av fire etasjer med kontormiljø. Seksjonen er samlet i tredje etasje. Arbeidsplassene er vanlige kontorarbeidsplasser, med opptil tre personer på hvert



kontor. Datasenteret er en av de viktigste delene på CERN. Når LHC starter opp, vil man gjennomføre forsøk som vil generere enorme mengder data. Hovedoppgaven til datasenteret og dermed IT-avdelingen er å lagre disse dataene. Totalt er det planlagt at datasenteret skal ha 8000 maskiner innen slutten av 2007.



Figur 7: CERN datasenter

#### 4.1.5 ELFMS (Extremely Large Fabric Management Software)

ELFMS<sup>6</sup> er CERNs egenutviklede "Fabric Management" system, og er et rammeverk for drift av datamaskinene deres. ELFMS består av fire hoveddeler; Lemon, SLS, LEAF og Quattor.

Lemon er et rammeverk for monitorering av datamaskiner. Arkitekturen på tjenesten er klient-tjener basert. På hver node som blir monitorert er det en monitorerende agent som ved

---

<sup>6</sup> Extremely Large Fabric Management System

hjelp av en push/pull protokoll henter og videreformidler monitorert informasjon fra sensorene på den enkelte node. All informasjon fra de respektive nodene blir lagret i en sentral DB. Klientprogrammet aksesseres fra en nettleser, og gir informasjonen som brukeren spør om.

SLS er et web-basert verktøy for vise informasjon om IT tjenester både for brukere og system administratorer. Informasjon som tilbys om tjenestene er blant annet tilgjengelighet, statistikk, og avhengigheter mellom dem. SLS tar i bruk LEMON for å finne informasjon, og bruker web-teknologi som AJAX, DOJO og DWR for å presentere informasjonen.

LEAF (LHC-Era Automated Fabric) er en samling verktøy som holder orden på service level på den enkelte maskin (i.e om maskinen er i produksjon, testing, feilsøking etc). Det består av SMS (State Management System) og HMS (Hardware Management System). SMS tilbyr høynivå kommandoer for å sette status på hver enkelt node, slik at man med en kommando kan sette maskinen i en ønsket status. For eksempel dersom maskinen skal reinstallerer, må man fjerne produksjonsstatus og sette den til service. Dette gjøres slik at LEMON ikke skal sende ut alarmer når den merker at maskinen blir startet på nytt under installasjonen.

Quattor står for "QUattor is an Administration Toolkit To Optimize Resources" og er et rekursivt akronym på samme måte som GNU står for "GNU is not Unix". Det er et rammeverk som består av en samling modulære verktøy for å støtte opp om håndtering (management), installasjon og konfigurering av cluster og datasentre som kjører Linux/Solaris.

Quattor ble til i forbindelse med EU sitt datagrid prosjekt EDG<sup>7</sup> (2001-2004) som skulle finne løsninger for neste generasjon vitenskaplig forskning. Dette krever enorme mengder prosesseringskraft og analyse av data fra storskala databaser i et distribuert forskningsmiljø. Quattor kan sies å være det grunnleggende limet som gjør det mulig å håndtere slike store datasentre.

---

<sup>7</sup> <http://eu-datagrid.web.cern.ch/eu-datagrid>

Filosofien bak Quattor er at det skal støtte alle noder<sup>8</sup> uavhengig av hvilke tjenester de kjører (batch node, disk tjener, tape tjener, Databasetjener etc), og uavhengig av hardware (Harddiskstørrelse, minne etc).

*The framework was created with a goal of developing new automated system management techniques that will enable the deployment of very large computing fabrics constructed from mass market components with reduced system administration and operation costs. The fabric must support an evolutionary model that allows the addition and replacement of components, and the introduction of new technologies, while maintaining service.*

EDG Technical Annex - Description of Work, 2000, <http://www.eu-datagrid.org/>

#### Hovedelementer i Quattor

- Selvstendige noder.
- Lokale konfigurasjonsfiler.
- Ingen script som blir styrt sentralt (remote management scripts).
- Ikke avhengig av globale filsystemer som AFS eller NFS.
- Sentral kontroll
- Konfigurasjon blir oppebevart sentralt (en kopi blir sendt til hver node).
- En sentral kilde for all informasjon om konfigurasjoner.
- Reproduserbarhet der alle operasjoner utført i produksjonen er idempotente og atomiserte.
- Validering av konfigurasjon informasjon før den blir satt i produksjon.
- Skalerbarhet med fokus på "load balanced" tjenere, og skalerbare protokoller.
- Portabilitet med støtte for både Linux, og Solaris<sup>9</sup>.

---

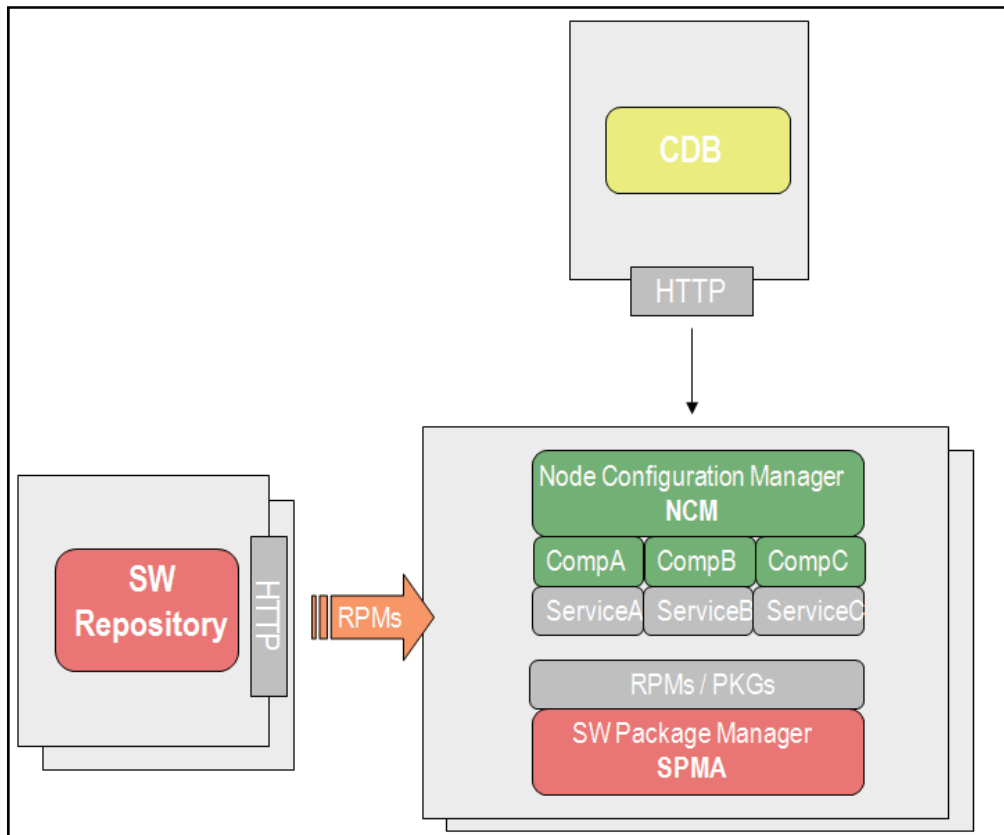
<sup>8</sup> Med noder menes en enhet i systemet som kan konfigureres av Quattor.

<sup>9</sup> Utvikling for Solaris har stoppet, og tjenester som kjører Solaris skal i etter hvert utfases.

Quattor bygger på en informasjonsmodell som skiller mellom ønsket tilstand og den faktiske tilstanden. Ønsket tilstand blir beskrevet i konfigurasjonsfiler kalt maler som er samlet i en sentral database CDB. Disse malene inneholder alt fra informasjon om de fysiske komponentene i enhetene, til software og konfigurasjoner for den enkelte enhet. Malene blir definert i et høgnivåspråk kalt PAN, og lagret i filtypen tpl (kortnavn for template). Selv om ønsket tilstand er beskrevet i CDB, kan den variere med den faktiske tilstanden til noden/clusteret. Oppdatert informasjon fra CDB blir oppdatert på den enkelte node en gang per dag. I tillegg er det også mulig å kjøre en manuell oppdatering på den enkelte maskin.

Som et delsystem av Quattor er det NCM (Node configuration manager) som sørger for at all informasjon er oppdatert. Det vil si at tilstanden til noden skal reflektere den tilstanden som er beskrevet i malen på CDB. Dette systemet starter en prosess for å hente ned den nyeste malen, og kontrollerer om det har skjedd endringer. Ved endringer så er det flere verktøy som blir kjørt, avhengig av hvilken endring det er snakk om.

SPMA (Software Package Manager) holder orden på hvilke programmer som er installert. Dersom programvare som er definert i malen ikke er installert, tar SPMA kontakt med SWRep som er lagringsstedet for programvare. Den aktuelle programvarepakken blir så hentet ned og installert ved hjelp av SPMA. Dersom programmet må konfigureres blir det gjort av NCM som igjen kaller en såkalt NCM-komponent for det aktuelle programmet og/eller tjenesten.



Figur 8: Quattor arkitektur (forenklet versjon)

En NCM-komponent er en plug-in for NCM systemet som blir kjørt når noden starter opp (booter), ved hjelp av en cron job eller når det er gjort endringer i CDB tilknyttet denne komponenten. Komponentene kan editere eller opprette nye konfigurasjonsfiler på systemet, stoppe og starte tjenester og sette avhengigheter for å sikre at en tjeneste blir konfigurert før en annen. Tabellen nedenfor viser et utvalg av de NCM-komponentene som blir finnes til Quattor, og hva de blir brukt til.

Navn	Beskrivelse
<b>ncm-quota</b>	Konfigurerer kvoter på noder.
<b>ncm-automounter</b>	Konfigurerer noder for automatisk mount.
<b>ncm-loadbalancing</b>	Konfigurerer lastbalansering mellom noder.

Figur 9: Tabell over NCM komponenter

### 4.1.6 RPM Package Manager

RPM er et pakkehåndteringssystem for å gjøre det enklere å installere og holde styr på programmer på noen Linux versjoner. Opprinnelig ble det utviklet av Red Hat for bruk i deres Linux distribusjoner, men blir i dag brukt av flere andre distribusjoner, og er også portet til Novell Netware og IBM AIX. På CERN blir det brukt som en del av Linux miljøet som Scientific Linux Cern (SLC) og Red Hat Enterprise Server.

Selve RPM-pakken inneholder programvaren (filene), metadata med informasjon om filene som den inneholder, en beskrivelse av programmet, og eventuelle script som skal kjøres ved installering og avinstallering. I tillegg så kjører RPM systemet en database med oversikt over hvilke programmer (rpm-pakker) som er installert.

For å bygge en RPM av et program må man sette opp et rpmbuild område med følgende kataloger:

SOURCES: inneholder programmet, eller kildekoden til dette.

SPECS: inneholder spec filen for å bygge den aktuelle RPM filen.

RPMS: området der ferdige RPM pakker blir lagt.

SRPMS: område for ferdige RPM pakker som inkluderer originale kildefiler i pakken.

Sammen med programmet er spec filen det viktigste i prosessen med å lage en RPM. I denne filen spesifiserer man hvordan programmet skal installeres, og eventuelt avinstalleres. Filen er delt inn i seks hoveddeler, som tar seg av alt fra å generere RPM filen til å installere og avinstallere den:

- Introduction section: Her er det mulighet til å spesifisere informasjon om programmet som pakken skal inneholde. Dette er informasjon som dukker opp når man gjør en spørring av hva som er i pakken, for eksempel med kommandoen `rpm -qi <pakkenavn>`. Eksempel på informasjon er versjon, utgivelse, navn, kopirett og beskrivelse.
- Prep section: Definerer kommandoer som skal bli brukt når RPM filer blir generert. Hvis for eksempel programmet i en komprimert tar fil, må dette pakkes ut her.

- Build section: Inneholder kommandoene for å bygge programmet.
- The install section: Inneholder kommandoene for å installere programmet.
- The clean section: Fjerner filer som blir laget av de to foregående delene.
- The files section: Denne delen lister opp alle filene som skal være med i RPM filen, og setter hvilke attributter de skal ha.

Når man har en ferdig spec fil, og riktig konfigurasjon – så kan man bygge RPM filen. Dette gjøres ved å kjøre rpmbuild med de parametrene man vil ha. Rpmbuild ser først om alt stemmer i spec filen, så lokaliserer den filene som ligger i SOURCE katalogen, og bygger selve RPM filen.

#### **4.1.6.1 Noen fordeler med RPM**

- Standardisert og enkel måte å installere/avinstallere programmer.
- Det finnes allerede tusenvis ferdiglagde RPM pakker av programmer.
- Støtter mange ulike plattformarkitekturer.
- Muligheter for verifisering og spørring av programvare som er installert.
- Oppgraderbare pakker.

#### **4.1.6.2 Ulemper med RPM**

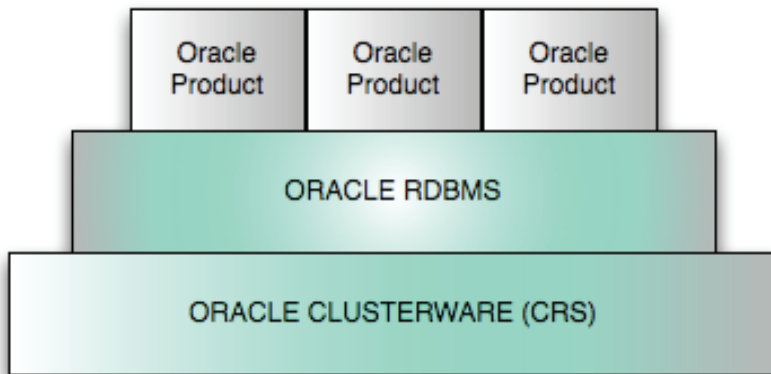
- RPM kan ha problemer når det gjelder å løse avhengigheter, man må stort sett selv løse slike problemer manuelt ved å laste ned pakkene som en annen pakke er avhengig av, men dette krever at man er klar over problemet på forhånd.

#### **4.1.7 Oracle databaseprogramvare**

Oracle tilbyr et stort utvalg i forskjellige databaseprogramvare i form av en rekke muligheter til å kombinere de forskjellige produktene deres.

DES gruppen bruker i hovedsak Oracle Database 10g, med Real Application Cluster (RAC) som valg. I tillegg bruker de også Oracle sin Enterprise Manager til å administrere mange av tjenestene som kjøres. Disse programvareløsningene krever et spesielt oppsett, og bygger på

hverandre. Hovedsystemet for en Oracle databaseserver er det som kalles RDBMS. Dette kan best sammenlignes med operativsystemet som må til for å kjøre alle de andre Oracle programvareløsningene. For å gjøre det mulig å installere databaseoperativsystemet som cluster løsning trenger man Oracle Clusterware.



Figur 10: Oracle programvareprodukter brukt på CERN

### 4.2 Min arbeidsoppgave

Sammen med tre andre personer fikk jeg ansvar for å se å sjekke ut hvilke muligheter det fantes for å lage en mest mulig automatisert installasjon av Oracle RDBMS. Spesifikt skulle vi se på om det var mulig å lage en RPM pakke installasjon av Oracle programvare, og hvordan denne kunne integreres inn i fabrikkssystemet ELFMS/Quattor. Hvis man fikk til en suksessfull integrering ville dette bety store besparelser i form av installasjonstid og arbeidskraft. En erfaren ingeniør bruker som regel 3-4 timer på en manuell installasjon av programvaren for å sette opp en Real Application Server (RAC). Prosessen krevde mye manuell intervensjon. Selvsagt var det mulig å spare noe tid ved å gjøre flere installasjoner samtidig, men pga den interaktive installasjonsmetoden krevde så mye svar fra den som installerte begrenset det seg fort til et fåtall maskiner som kunne installeres samtidig. En annen bakgrunn for denne oppgaven var at gruppen skulle utvide med over 100 Oracle maskiner i løpet av 2007, og da var det en naturlig valg å se på mulighetene med automatisering.



## 5 RESULTAT

### 5.1 Arbeidet

Selve arbeidsprosessen som jeg har studert startet 8. desember 2006 og gikk til 4 juni 2007. I tillegg hadde jeg på høsten 2006 gjort små arbeidsoppgaver med ncm-komponenter tilknyttet en annen oppgave som hadde som mål å automatisere konfigurasjonen av en databasetjeneste. Dette arbeidet ble lagt ned i desember 2006, da bare en del av konfigurasjonene hadde blitt automatisert. Begrunnelsen fra ledelsen var at man fra nå av skulle satse på automatisering av fremtidige tjenester og ikke allerede kjørende. Dette gav god mening, da en automatisering krever mye innsats, og nytteverdien er best når man setter opp systemet for første gang. En av de ansatte gav følgende kommentar om databasesystemene i gruppen:

*Once the database software is installed, most of our machines will never be reinstalled. They will just run until they get replaced, or they are needed for other services.*

Et av hovedsatsningsområdene for gruppen for 2007 var oppsett av NAS<sup>10</sup> baserte databaseklynger (av engelsk databasecluster) av maskiner der gruppen skulle tilby en rekke tjenester. I første omgang skulle 40 noder settes opp sammen med 4 NAS lagringsenheter, og senere skulle løsningen utvides med ca 100 noder til og rundt seks NAS enheter. Løsningen ble designet for både å tilby hastighet og sikkerhet i form av redundante løsninger (strømforsyninger, nettverksrutere), brannmurer osv. Da flesteparten av disse skulle kjøre en eller annen form av Oracle databaseprogramvare var gruppen veldig interessert i å se på mulige løsninger for å automatisere installasjon og konfigurasjon av disse produktene. Jeg og to fast ansatte fikk i desember 2006 ansvaret for å undersøke hvilke muligheter som var for automatisering av Oracle. Den ene av disse hadde veldig mye kunnskaper om Quattor, noe som ville komme til nytte da planen var å senere integrere installasjonen med dette. Den andre hadde automatisering generelt som sitt spesialfelt, og hadde gjennomført

---

<sup>10</sup> NAS: Network Attached Storage. En betegnelse på maskinvare for lagring av datafiler på et lokalt nettverk.

automatiseringsprosjekter for LHC@home<sup>11</sup>. Meg inkludert, så hadde ingen av disse noen kunnskap om Oracle databaser. Så det første vi gjorde var å delta på et innføringskurs for Oracle databaseadministratorer. Under resten av perioden tilegnet vi oss kunnskap ved først og fremst å lese dokumentasjon og ta kontakt med andre personer i gruppen. Hvordan dette skjedde vil jeg komme tilbake til senere.

### 5.1.1 Alternativer

Vi startet opp med møte for å kartlegge hvilke utfordringer vi stod ovenfor, og ikke minst hvilke alternativer som fantes for å løse disse utfordringene. Etter dette møte hadde vi bestemt oss for å følge opp to mulige løsninger som hadde kommet frem; Installasjon ved hjelp av response filer, eller installasjon ved å bruke kloning.

#### 5.1.1.1 Installasjon ved hjelp av response filer

Dette så vi for oss som den enkleste løsningen. Responsefiler var Oracle sin løsning for å gjøre en halvautomatisk installasjon, der brukeren slipper å velge forskjellige innstillinger å skrive inn konfigurasjonsopplysninger i installasjonsveiviseren. Denne løsningen ble omtalt som "unattended installation", det vil si en installasjon som gjøres uten tilsyn. Responsefilene var en type malfiler som inneholdt alle konfigurasjoner for en gitt Oracle installasjon. Når man starter installeringsprosessen gir man opp navnet på responsefilen med følgende parametre:

```
OraInstall12007-03-28_04-49-00PM/ OraInstall12007-07-03_02-51-29PM/  
OraInstall12007-03-30_02-20-22PM/ ssh-1ZLbU18012/  
OraInstall12007-03-30_02-29-56PM/ y/  
[oracle@dbsrvd117 database]$ ./runInstaller -silent -responsefile rdbms.rsp
```

Figur 11: Skjerm bilde fra database installasjon

Når installasjonsprosessen er ferdig, trenger man å kjøre to skript og da skal man ha fått en installasjon i samsvar med det som er beskrevet i responsefilen. For å gjøre denne installasjonen automatisk kartla vi hvilke parametre som det senere kunne være behov for å endre i responsefilen, for eksempel hvilke komponenter som skulle være med, nodenavn etc.

---

<sup>11</sup> <http://athome.web.cern.ch/athome/>

Ideen var at vi kunne lage en ncm-komponent som genererte en responsfil på bakgrunn av de opplysninger vi ga i en malfil. Så ville komponenten starte installasjonen med den nye responsfilen og på slutten kjøre skriptene.

Det ble tidlig kjent at denne ideen medførte store ulemper med seg, og var derfor vanskelig å gjennomføre. Blant annet ble det vanskelig med så mange avhengigheter, der ncm-komponenten først skulle generere, starte installasjon og kjøre to skript. Hvis et av de forskjellige leddene feilet, ville man ikke ha noen god oversikt over når det skjedde, og hvor det skjedde. Selv om dette kunne forbedres ved å lage sjekker under prosessen, og generere loggfiler så var dette en ulempe som ville medføre mye ekstraarbeid, og gi flere risiko med at noe gikk galt under installasjonsprosessen. Som beskrevet tidligere finnes det forskjellige feilrettinger til Oracle, der de største blir installert på samme måten som grunninstallasjonen (ved å bruke runInstaller), mens de minste krever at man bruker Opatch eller følger en manuell instruksjon. Da ikke alle kunne installeres med responsefiler, vil det si at man må manuelt oppdatere hver enkelt maskin med patcher. Dette ville da gitt en slags halveis automatisering, der første del gikk fint, men så måtte man installere resten for hånd. Det er mulig å automatisere installasjonen av slike feilrettinger, men siden hver feilretting er forskjellig må man i så fall lage et skript til hver enkel feilretting. I tillegg så finnes det avhengigheter rundt disse feilrettingene, så i noen tilfeller så må de installeres manuelt. For oss var ikke dette noe alternativ, og var avgjørende til at vi valgte å se bort fra denne metoden.

#### ***5.1.1.2 Installasjon ved hjelp av kloning og RPM***

Dette alternativet hadde vi ikke lagt så mye vekt på i begynnelsen, da det involverte både kloning, rpm, og ncm-komponenter. Kloning hadde ingen av oss noe kjennskap til. Jeg fikk ansvaret for å se på hvordan vi skulle bruke denne teknikken, og ble derfor bedt om å lage en demonstrasjon i form av et "proof-of-concept".

Det første jeg gjorde var å registrere en konto på Metalink. Oracle Metalink er kunnskapsdatabase som Oracle tilbyr sine største kunder, deriblant CERN. Her fant jeg en artikkel om kloning på (Oracle, 2007), og gjorde en installasjon etter denne oppskriften. Uten

å gå i for mange detaljer er hovedideen å lage en referanseinstallasjon kalt en "Golden Master" som er slik man ønsker at en installasjon skal være. Denne pakker man så ned til en komprimert fil, overfører og pakker den ut på målmaskinen – for så å starte kloneprosessen. Selve kloningen startes ved å kjøre et skript som heter clone.pl med parametre for plassering av programvaren på harddisken og navn på dette området, og andre spesifikke settinger for nettopp den noden. Kloningen relinker alle filer, slik at programvaren kan brukes på den bestemte maskinen. Vi lagde flere tester tester rundt kloningen, og fant ut at det var et konsept som kunne fungere. Selv om dette fungerte, så advarte en i arbeidsgruppen oss med at vi hadde mye igjen:

*Now we have to make an rpm packet out of this, and get that one to work. Later we will have to find out how to configure the nodes with node specific settings. In one month we will probably have a working version, and then we can have a presentation. People will say how stupid we are, and point to all the faults we made. Then we have to improve the version again, and do the same thing again. After three iterations we will probably have something that can be tested.*

Denne prediksjonen stemte ganske bra. Etter en mengde jobb med prøving og feiling, møtte vi ganske stor skepsis og kritikk da vi presenterte vårt resultat for første gang. Folk så på installasjonen som en svart boks der man ikke kunne se hva som skjedde, og hadde noen kontroll over installasjonen selv.

Etter å ha endret det meste som ble påpekt, og forklart hvordan prosessen foregår til de andre – hadde vi laget en versjon som var klar for test. Denne ble testet av to erfarne personer i gruppen som var de som oftest installerte og oppgraderte Oracle programvare. Resultatene var utrolig positive. Alt så ut til å fungere, og de hadde bare noen få ønsker om endringer.

Så det vi hadde nå var en rpm fil som kunne installere Oracle RDBMS etter de standardene gruppen hadde satt opp. Det gjenstod likevel en del for å få alt til å settes opp automatisk. Blant annet måtte RPM pakken legges inn i det lokale SWrep, og det måtte lages en ncm-komponent som konfigurerte individuelle innstillinger for hver node. Status for 4. juni 2007

var at vi hadde to RPM basert installasjon av Oracle RDBMS. Vi hadde også en ncm-komponent som tok seg av oppdatere konfigurasjoner på de forskjellige nodene.

## 5.2 Observasjoner

### 5.2.1 Misforståelser

Selv om arbeidet har gått veldig bra, har det ikke vært fritt for misforståelser og endrede meninger.

11 til 15 juni var jeg på en konferanse, og hadde ikke mulighet til å være med de andre i arbeidsgruppen. En i gruppen ble bedt om å lage en ny rpm versjon av Oracle RDBMS 10.2.0.3, da den skulle installeres på et nytt sett av maskiner. Personen fulgte oppskriften som var beskrevet på TWIKI siden for arbeidet. I tillegg installerte han alle feilrettingene som var beskrevet i den offisielle siden for feilrettinger som skulle brukes i vår gruppe. Da jeg kom tilbake fra konferansen hadde to databaseadministratorer begynt å bruke denne RPM versjonen. Vi fikk klager av at flere viktige oppgraderinger manglet, og at den ikke hadde støtte for Oracle RAC. Her var det flere ting som hadde gått galt. Blant annet hadde jeg glemt å skrive om hvordan man gjorde slik at installasjonen støttet RAC. Et annet problem var at listen over nødvendige feilrettinger manglet en oppføring, så en kritisk feilretting hadde ikke blitt installert.

### 5.2.2 Språk og Kulturforskjeller

Jeg hadde ventet at de ville by på en del utfordringer når man skulle utveksle kunnskap med personer med ulik kulturell bakgrunn. Blant annet var det seks forskjellige nasjonaliteter i SIS seksjonen, da jeg begynte den 1. september. Selve kommunikasjonen og utveksling gikk overraskende smertefritt, og arbeidsspråket som ble brukt var en blanding av engelsk og fransk. Det vil si at fransk hadde dominert på CERN før, men ble mindre og mindre brukt i IT-avdelingen. Personer som hadde jobbet her i lengre tid brukte Fransk når de pratet med hverandre, men slo over til Engelsk når de skulle prate med resten av seksjonen. Fortsatt fantes det avdelinger ved CERN der all kommunikasjon foregikk på Fransk, særlig da på den

franske delen. Når offisielle dokumenter ble sendt ut, ble det benyttet både engelsk og fransk, med den franske teksten først.

### 5.2.3 Innføringen av et nytt kunnskapsystem

I begynnelsen av november 2006 ble vi introdusert for et nytt system for feilhåndtering og prosjektstyring. Jira<sup>12</sup>, som programmet het skulle gi personer i gruppen og avdelingen oversikt over hvilke de andre jobbet med og status på disse. Det ble spesielt fremhevet hvor nyttig denne oversikten var for de forskjellige lederne (seksjons, gruppe og avdelingsleder). Alle fikk sin egen konto på systemet, og ble bedt om å oppdatere informasjon om sine prosjekter. Et problem med systemet allerede i begynnelsen var at det var ustabil. En annen ulempe var at vanlige brukere ikke kunne legge til sitt eget prosjekt dersom det manglet. I slutten av observasjonsperioden logget jeg meg igjen inn på Jira systemet, og så da at det bare var to aktive brukere.

### 5.2.4 Konkurransen

Midtveis i arbeidet med den automatiske installasjonen fikk vi vite at en annen person tilknyttet vår gruppe også så på muligheter med automatiske installasjoner. Selv om han hadde en annen fremgangsmåte, ble dette en direkte konkurrent til vårt arbeid. Det er vanskelig å si om dette hadde noe innvirkning på arbeidet som gjenstod, men det ble en konkurransesituasjon alle tok seriøst. Om dette var en noe som var initiert gjennom ledelsen, eller om personen gjorde det for sin egen skyld visste vi ikke noe om.

Senere ble jeg fortalt at det var vanlig praksis blant fysikerne å sette opp to konkurrerende prosjekter med samme mål. Dette skulle bidra til mer effektivitet og en kvalitetssikring. Dersom to prosjekter fikk samme resultat når fremgangsmåten var forskjellig, var dette med på å styrke funnet ytterligere. Hans F. Hoffman kommenterer det slik:

*Because of the complexity of the collaborations' tasks and as ultimate quality assurance for the physics findings there is competition in all their doings. For example the ATLAS and CMS collaborations have the same overall scientific goals, schedule and*

---

<sup>12</sup> <http://www.atlassian.com/software/jira/>

---

*potential. They will compete for better results throughout the lifetime of the LHC programme. Hans F. Hoffman (Hoffmann, 2007)*

### 5.2.5 Det polske nettverket

20 april kl 15:00: Under installasjon av det underliggende Oracle Clusterware (CRS) fikk jeg og annen teknisk student problemer, da vi ikke husket kommandoen for å starte en del av CRS igjen med en spesiell instilling. Jeg fortalte studenten at jeg visste at denne fantes i dokumentasjonen, men jeg visste dessverre ikke hvor. Vi hadde et ganske stort press på oss for å få denne tjenesten opp i løpet av dagen, og den andre studenten så nok at jeg var litt stresset da han sa:

**Studenten:** No worries, I'll just use Kadu Kadu<sup>13</sup>

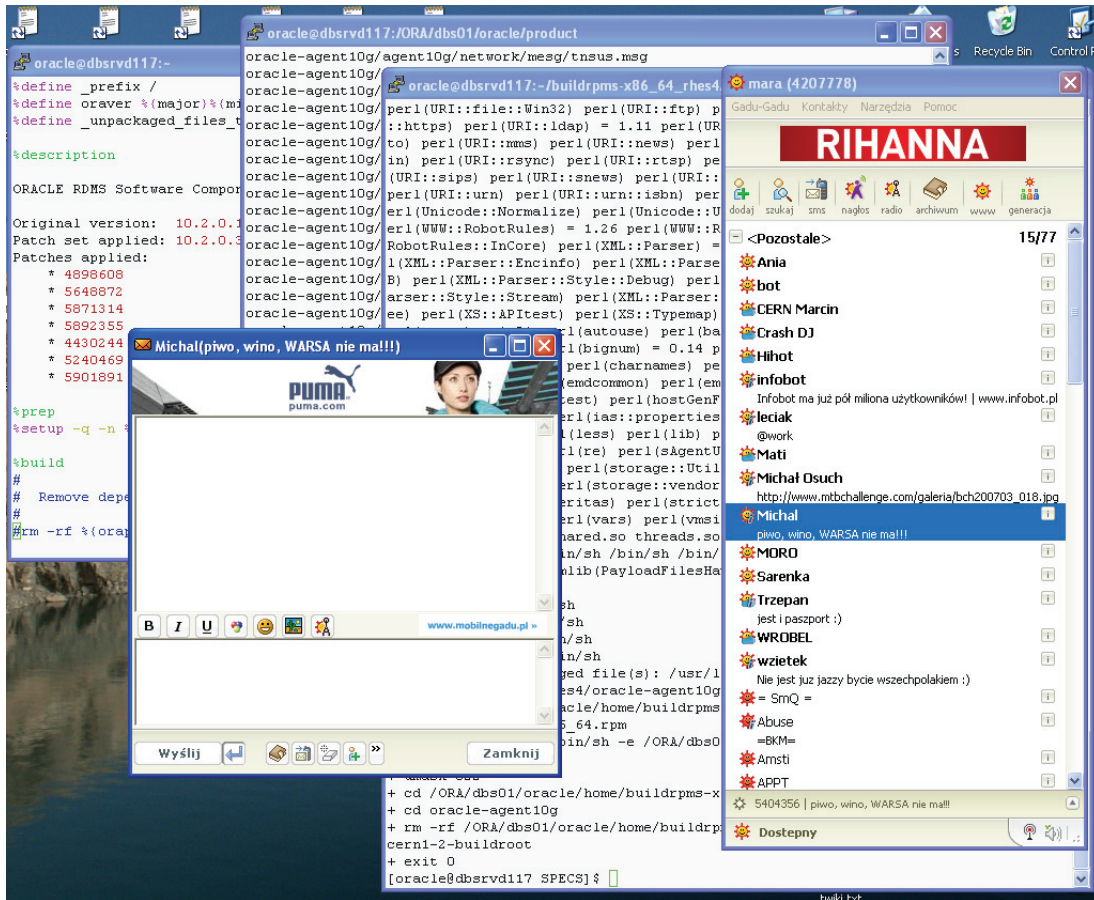
**Meg:** What is that? Some kind of frequently asked questions for Oracle software?

**Studenten:** No no, it's a polish instant messaging program. Just like Microsoft messenger, only better. All the Polish people in the group (IT-DES) use it to exchange knowledge about Oracle, agree on lunch and other social things. I'll ask them about the command"

Noen sekunder senere fikk vi med den rette kommandoen, og kunne fortsette arbeidet. Studenten forklarte at alle hadde sin spesialitet innen Oracle, det vil si noen jobbet med sikkerhetskopiering og gjenoppretting, applikasjonstjenester, installasjon, monitorering etc. Ved å lage en gruppe i Kadu Kadu kunne hver enkelt hjelpe med sitt spesialfelt når man lurte på noe, og siden alle andre så samtalen fikk også de den kunnskapen. De postet også informasjon som de kom over uten at noen hadde spurt etter den, men mer som tips til nyheter om Oracle eller noe de selv drev på med. En annen fordel med dette nettverket var at man slapp å spørre en person direkte. Det var ingen garanti at den personen var tilstede på kontoret, kunne svare på spørsmålet eller hadde tid. Så ved å bruke Kadu Kadu når man mange personer, da meldingen blir sendt ut til alle som er pålogget.

---

<sup>13</sup> <http://www.kadu.net>



Figur 12: Skermbilde av Kadu Kadu

De nordiske studenter som var på CERN brukte lynmeldingstjenesten Microsoft Live Messenger til å kommunisere, men dette var på tvers av avdelinger. Da ikke alle jobbet innen samme område ble den utvekslede kunnskapen av begrenset nytteverdi. Jeg ble flere ganger kontaktet av fysikkstudenter som hadde forskjellige tekniske problemer. De kontaktet meg fordi jeg jobbet på IT-avdelingen. Nettverket ble ellers brukt til å avtale lunsj og middag.

### 5.2.6 Epost

Epost ble brukt i utstrakt grad, og da særlig epostlister. Hvert prosjekt hadde sine egne epostlister, slik at deltakerne kunne diskutere og dele informasjon seg imellom. Spesielt epostlisten for utviklingen av NCM komponenter til Quattor og ELFMS var preget av mye



aktivitet. Daglig ble det postet meldinger der en eller annen hadde et forslag til utvikling av en ny komponent. Personen skisserte som regel opp en

### 5.2.7 TWIKI

TWIKI er CERN sin wiki variant for samhandling. Wiki er en hovedbetegnelse på gruppeprogramvare der innholdet består av hypertekstdokumenter som er relatert til hverandre. Ideen er at alle skal ha tilgang til hverandres dokumenter, slik at utveksling av kunnskap og samarbeid skal være enklere. I tillegg kan alle opprette nye dokumenter, og editere gamle. Alle disse wikivariantene stammer utfra wikipedia, som er en wikibasert encyclopedia på Internet.

På CERN brukes TWIKI blant annet til å dokumentere tjenester, praksiser, og som gruppesider for prosjekter. De fleste avdelinger er representert på TWIKI, og bruker denne til å presentere hva de holder på med av prosjekter. Ved nye prosjekter eller arbeidsoppgaver blir det som regel opprettet sider rundt dette. Det er da enkelt for alle å se hva gjøres på det enkelte prosjektet.

I begynnelsen av arbeidsprosessen ble det opprettet sider på TWIKI. Her beskrev vi hva som var målet vårt, fremdriftsplan, referanser til system vi tok i bruk osv. Vi benyttet sidene som en arbeidslogg, og skrev inn rett hva vi gjorde under de forskjellige installasjonene. Disse sidene ble merket som utkast (draft), slik at vi ikke behøvde å bekymre oss over at andre skulle se på de som ferdige sider. Tilleggsfiler som script, og patcher ble også lastet opp på prosjektsidene. Hvis en av personene i arbeidsgruppen ikke var på jobb, var det mulig å sjekke hva denne personen hadde gjort på prosjektet i mellomtiden.

Et av problemene med TWIKI var at detaljgraden i dokumentasjonen var veldig varierende. For eksempel var det noen som skrev veldig detaljert, der alt ble forklart grundig med pekere til andre sider for mer forklaring. På den andre siden var det personer som satt på ekspertsystemer, som bare kopierte inn kommandoene som de hadde brukt, og skrev et par ufullstendige setninger om systemet.

### **5.2.8 Møter**

Møter er et annet moment som jeg vil belyse når det gjelder mine observasjoner på hvordan kunnskap ble utvekslet. Jeg vil her gi en oversikt over ulike faste møter jeg deltok på med fokus på gruppen jeg jobbet i:

#### **5.2.8.1 Arbeidsgruppemøter**

Møter med arbeidsgruppen har pågått under hele oppdraget, der noen har blitt holdt til faste tider og andre som ad hoc møter. I gjennomsnitt har det blitt holdt møte hver andre uke, der minst to fra arbeidsgruppen har deltatt.

#### **5.2.8.2 Seksjonsmøter**

Seksjonsmøter ble arrangert i snitt hver annen uke, og ble holdt av seksjonssjefen. Han pleide å åpne med å gi beskjeder fra IT ledelsen, fortelle om hendelser i organisasjonen siden forrige møte og hva som var planene fremover. På møtet fikk man også oversikt over hva de andre i seksjonen jobbet med, og hva statusen var på dette.

#### **5.2.8.3 Gruppemøter**

Gruppemøter ble arrangert relativt sjelden, i gjennomsnitt hver tredje måned. Allerede den første uken jeg var på CERN ble det holdt et gruppemøte. Det ble oppsummert den gjeldende økonomiske situasjonen, og hva som trolig kom til å skje fremover. Det store gjennomgangstemaet var nedskjæringer, og hvordan dette kom til å påvirke gruppen. Det ble også arrangert et spesielt gruppemøte en gang i året som ble kalt "Program of Work". Dette møtet gikk over en hel arbeidsdag, og var en oppsummering på hva som hadde skjedd det siste året, samtidig en diskusjon om hva man skulle satse på av tjenester i årene som kommer. På forhånd hadde alle seksjonene forberedt seg ved å kartlegge all kunnskap de hadde og hvilke tjenester de tilbydde.

#### **5.2.8.4 Avdelingsmøter**

Hver sjettede måned holdt avdelingslederen et felles møte for alle i IT-avdelingen. Dette var en presentasjon av hva de forskjellige gruppene og seksjonene jobbet med, slik at alle i IT organisasjonen fikk en oversikt.

### 5.2.8.5 Organisasjonsmøter

Hver sjette måned holdt generaldirektøren et organisasjonsmøte. Her informerte han om CERN sin finansielle status, og en statusoversikt over hovedforskningsprosjektene. Til slutt gav han en presentasjon av hver avdeling med en oversikt over hva de jobbet med. Denne oversikten baserte seg på avdelingslederne sine presentasjoner, som igjen baserte seg på gruppelederne sine presentasjoner.

Type	Tema	Deltakere	Hyppighet	Antall	Lengde
<b>Arbeidsgruppemøte</b>		4-10	Hver 3. Uke	9	30-60 min
<b>Seksjonsmøte</b>	Orientering fra seksjonsleder og gjennomgang av seksjonsmedlemmers rapporter.	6-12	Hver 2. Uke	19	30-60 min
<b>Gruppemøte</b>	Orientering fra gruppeleder.	30-40	Hver 3. Mnd	4	1-2 timer
<b>Avdelingsmøte</b>	Orientering fra avdelingsleder.	150-200	Hver 6. Mnd	1	2-3 timer
<b>Organisasjonsmøte</b>	Orientering fra generaldirektør.	2000+	Hver 6. Mnd	2	2-3 timer

Figur 13: Oversikt over møter

### 5.2.9 Kaffepauser

Klokken 10 hver dag går de fleste i IT-avdelingen over veien til restaurant 2 for å drikke kaffe. Dette er et daglig ritual som samler folk rundt i kaffekrokene i restauranten. Under pausen diskuteres det i alt fra jobbrelaterte emner til det siste valget i Frankrike. Jeg har lagt merke til at dette virker som en god mulighet til å spørre kollegaer om arbeidsrelaterte spørsmål, og samtidig utveksle erfaringer. Det er mye lettere å spørre, da man ser om personen er tilgjengelig eller ikke. Selv på workshoper der hovedhensikten er å diskutere og løse

problemer, skjer mye av problemløsningen i pauser. En grunn til dette kan være at det er lettere å stille spørsmål i en slik avslappet og uformell setting, samtidig kan det være spørsmål som ikke trengs å diskuteres i store fora. På en av samlingene i IT-auditoriumet observerte jeg noen personer som diskuterte hvor nyttig slike samlinger var. Den ene av dem mente at til og med pausene var veldig produktive. Den andre svarte:

*"The whole seminar is about the coffee breaks. Forget about the talks!"*



**Figur 14: Bilde fra kaffepause**

### 5.2.10 Google

I en kaffepause 13. april fortalte en doktorstudent noe interessant:

*"I use Google for anything now, even as a spellchecker"*

Jeg ba derfor han om å fortelle mer om hva han mente med å bruke google som stavekontroll. Doktorstudenten fortalte at om han var i tvil om hvordan et ord staves skrev han det inn i google. Hvis han hadde to alternativ, søkte han etter begge ordene. Hvis for eksempel den ene skrivemåten fikk 100 sider i treff, og den andre 10000, var det mest sannsynlig at det

ordet med flest treff var den rette måten å skrive ordet på. Selv om dette var interessant, var det ikke overraskende. Gjennom min observasjon har sett flere eksempler på bruke Google i arbeidet, og jeg har ved flere anledninger brukt det selv.

De første dagene på CERN gikk med på å ordne det praktiske, slik som adgangskort, attester og brukerkontoer. Jeg fikk to kontoer for to påloggingstjenester, NICE for Windows og AFS for UNIX/LINUX. Da jeg logget på Windows ble jeg bedt om å bytte passord med en gang, slik at jeg ikke skulle bruke mitt autogenerated. Da jeg logget inn på LINUX fikk jeg beskjed om at passordet mitt ville gå ut i løpet av noen dager, og at jeg måtte bytte. Jeg spurte noen i min seksjon om hvordan jeg skulle bytte dette, men de visste ikke. Men bør ikke dette stå en eller annen plass spurte jeg. Jo, du kan ta å google<sup>14</sup> etter det sa de. Jeg ble noe forundret, men tastet inn følgende søkestreng: "change password AFS CERN". I andre treff stod følgende forhåndsvisning:



Figur 15: Skjermbilde av google søk

Da jeg gikk inn på siden forstod jeg at det allerede var grundig dokumentasjon rundt de forskjellige innloggingsstandardene. Problemet var at det ikke var like lett å finne frem til dem. På forhånd hadde jeg også lett etter informasjon fra cern.ch som er en slags portal man kommer inn på. Her fant jeg ikke noe. Så når jeg har lett etter noe og ikke funnet det på CERN sine sider har jeg prøvd Goggle. Her følger et par eksempler på situasjoner:

<sup>14</sup> I tillegg til å referere til firmaet google og søkemotoren, brukes google som synonym for søke etter.

Mandag 2. november 2006 skulle jeg flytte til et nytt kontor. Jeg fikk beskjed om å flytte datamaskinen som stod på kontoret samtidig. Siden en datamaskin er registrert mot et nettverksuttak, måtte jeg melde fra om at jeg skulle flytte maskinen til et annet. For å finne skjema for dette søkte jeg på google igjen, fant det elektroniske skjemaet og fylte dette ut. Jeg flyttet inn på kontoret samme ettermiddagen, og alt fungerte.

Lørdag 28. januar 2007 var jeg med skiklubben på CERN opp i de franske alpene. I lunsjen satt jeg ved et bord med studenter fra de fleste avdelinger på CERN. Jeg tenkte derfor at dette var en gyllen mulighet til å høre om de brukte Google på samme måte. Jeg spurte så om hvordan de fant informasjon angående rutiner som skifte av passord, legge til skrivere, booking av rom, eller annen informasjon i sammenheng med studiet deres. Alle sa at de brukte Google til det meste av dette, noe som forsterket det inntrykket jeg allerede hadde av Google som svar på det meste på CERN. Som et tilleggsspørsmål spurte jeg om de brukte noe som helst intranett eller lignende for å finne svar på ting, men nei – det var Google som gjaldt for de fleste. En jente fra USA mente at det var enda enklere å google etter informasjon i USA, enn Frankrike og Sveits. I USA var det for eksempel veldig enkelt å finne åpningstider på butikker og lignende, noe som hun opplevde som mye vanskeligere i Sveits og Frankrike.

Google er ikke alltid din venn. Det fant jeg ut 13. februar 2007. Ved en feil så hadde jeg søkt om fri en annen dag en det jeg hadde tenkt meg på EDH systemet. Så for å finne ut hvordan jeg slettet denne søkte jeg på Google. Et av de første treffene var en pdf fil med brukerveiledning for EDH, og et av kapitlene gav en trinnvis veiledning for å kansellere fridager. Jeg fulgte veiledningen flere ganger, men fikk ikke kansellert mine fridager. Kanskje det var en feil på systemet tenkte jeg, og bestemte meg for å prøve igjen neste dag. Men samme resultat da også. Tilfeldigvis trykte jeg på hjelp knappen på EDH som et siste skrik om nettopp hjelp. Jeg fikk da opp en PDF som lignet på den jeg hadde funnet på Google. Men dette var ikke den samme, og hadde ordene new version i seg. Da jeg fulgte denne oppskriften, fikk jeg kansellert friperioden. Selv om dette var første gang jeg har fått "feilinformasjon" ved å søke etter hjelp på Google, så viser det at man ikke må ta all informasjon man finner på denne måten for gitt. Informasjonen kan være utdatert, eller direkte feilinformasjon publisert av andre.

## 6 ANALYSE OG DISKUSJON

### 6.1 Distribusjon av kunnskap på CERN

Observasjonene som ble gjort i dette studiet viser at delingen av kunnskap på CERN foregår på mange forskjellige plan, mellom forskjellige personer, i forskjellige kontekster og ved å ta i bruk forskjellige hjelpemidler. Jeg vil her gå gjennom noen av de mest markante resultatene og knytte disse opp mot teorien som ble beskrevet i teorikapittelet.

#### 6.1.1 Praksisfellesskap

I resultatkapittelet viser jeg til flere praksisfellesskap som jeg observerte i min studie av IT-avdelingen på CERN. Jeg har selv hvert medlem av flere av disse nettverkene. Praksisfellesskap blir betegnet som en sammensatt gruppe av personer med bakgrunn i noe de har til felles (for eksempel jobber sammen), og at de bruker hverandre for å utvikle sin kunnskap om det de gjør (Brown og Duguid, 1991; Lave og Wenger, 1991).

##### 6.1.1.1 Kaffepauser

På samme måte som teknikerne Julian E. Orr (1996) beskriver, blir mye av kunnskapen i IT-avdelingen på CERN generert og distribuert i kaffepauser og andre uformelle settinger. Også Karin Knorr Cetina (1999: p 174) viser til at de tre restaurantene på CERN generer teknisk prat. Hun skriver blant annet at det ikke er uvanlig for noen at halve arbeidstiden går bort i engasjerende kaffeprat. Men som hun avslutter med, så er arbeidsdagen lengre enn den offisielle arbeidstiden.

##### 6.1.1.2 Det polske nettverket

Et av de mest suksessfulle praksisnettverkene jeg kom over var det polske nettverket i IT-avdelingen. Ved bruk av lynmeldingsprogrammet<sup>15</sup> Kadu Kadu hadde alle de polske personene som jobbet i gruppen utviklet et praksisfellesskap med den bakgrunn at de alle jobbet med Oracle løsninger og kom fra samme land. Når en ny person begynte i gruppen ble han

---

<sup>15</sup> Norsk oversettelse av begrepet instant messaging

introdusert for fellesskapet, og han fikk nytte av all den kunnskapen som fantes der. På samme måte som Lave og Wenger beskriver så vokser deltagelsen til personen etter hvert som han får mer og mer kunnskap gjennom fellesskapet.

### 6.1.2 Seksjonsgruppen

Alle som jobbet i seksjonen dannet et naturlig fellesskap som følge av at vi omgikk hverandre så mye og jobbet stort sett innen samme område. Det beste eksempelet på utveksling var det ukentlige seksjonsmøte. Som nevnt i observasjonen, så måtte alle skrive en liten rapport om hva de hadde arbeidet med siden sist møte. På møtet ble rapportene gjennomgått og hver enkelt kunne forklare nærmere hva han eller hun hadde funnet ut, hadde problemer med eller hadde et forslag om. Dette ble til en samtale rundt det aktuelle temaet, og forslag eller kommentarer ble gitt. Dette fungerte veldig bra, og gav en følelse av lagarbeid for den enkelte som hadde skrevet rapporten. Noen av medlemmene brukte også historier for å beskrive hendelser som hadde skjedd og resultatet av disse. Blant annet startet en av de mer erfarne i gruppen historien på følgende måte:

*Once there was this EMC storage that was connected to the AIS RAC and needed to be upgraded with additional disks. This jolly good guy who was going to do this had looked for specifications on the Internet, but did only find the specs for the previous model.....*

Historien fortsatte i samme stil for å forklare hva som skjedde når han i god tro satte inn en disk i systemet som han mente var hot-swappable<sup>16</sup>. En av databasetjenestene vi kjørte måtte tas ut av drift en hel dag, og rammet over 3000 brukere.

Medlemmene i gruppen holdt også presentasjoner om hvilke tjenester de jobbet med, og på denne måten fikk alle i gruppen mer kunnskap om de aktuelle tjenestene. Det ble også arrangert små sosiale sammenkomster i lunsj og på kveldstid, slik at man ble lettere kjent med

---

<sup>16</sup> Egenskap hos en enhet som gjør at den kan tas ut eller settes inn uten å slå av strømmen, eller konfigurere systemet på forhånd.



folk. Dette var med på å forsterke praksisfellesskapet, og gjorde det lett å bli integrert i form av rettmessig periferisk deltagelse.

### **6.1.2.1 Vignette på praksisfellesskapet**

David begynner i IT-avdelingen i en organisasjon. Han er svært spent på den første dagen, da dette er helt ny situasjon for han. Han kjenner ingen i organisasjonen fra før, og han har et hav av nye systemer, rutiner og kulturer som han må lære å kjenne og forholde seg til. Etter å ha fått tildelt kontor begynner han å jobbe. Han skrur på datamaskinen og prøver å logge seg inn. Dette gir bare en feilmelding om at brukernavn og passord er ikke god tatt på systemet. Desperasjon begynner å melde seg hos han, da han har lovt sjefen at å ordne noe før lunsj og nå begynner tiden å bli knapp. Pannen hans begynner å bli svett og han kaster et blikk ut på gangen. Da får han øye på en person som kommer mot han. Personen stopper opp og sier: Hei, deg har jeg ikke sett før. Du er ny her, er du ikke? David nikker bestemt med hodet og ser på personen som strekker frem en hånd og sier: Jeg er Jon, og sitter i kontoret ved siden av. Han fortsetter:

Har du problemer med innloggingen på systemet? For å gå inn på dette må du legge til en tallkombinasjon fra ditt ansattnummer til brukernavnet. Legg til de to siste sifrene, så må du spesifisere systemet du skal logge på, avdelingskode og passord. La meg vise deg.

Jon viser David hvordan det gjøres. Sånn nå kan du gjøre det selv sier Jon. David fomler litt i begynnelsen, men til slutt får han logget seg inn. Ta det med ro sier Jon. Systemene er litt forvirrende i begynnelsen. Bare spør meg om det er noe du lur på.

Etter noen måneder i organisasjonen begynner David å mestre alle systemer, og gjør en god jobb. Han tenker at det er utrolig hvor mye han har lært disse månedene siden han begynte å jobbe. En dag da han går i gangen på vei til sitt kontor ser han en ukjent person inne på et kontor som ser ganske fortvilet ut. David går inn og sier: Hei, deg har jeg ikke sett før. Du er ny her, er du ikke? Personen nikker, og hilser på David. Har du problemer med innloggingen på systemet fortsetter David. Per, den nyansatte nikker nok en gang bekræftende. David forteller

og viser Per hvordan det skal gjøres. Da går det plutselig opp for han. Dette er jo RPD, rettmessig periferisk deltagelse.

### 6.1.3 Fortellinger

Orr (1991) forteller om hvordan service teknikere brukte fortellinger for å finne løsninger på tekniske problemer som ikke var dekket av service manualene. Gjennom å fortelle historier om tilsvarende problem når de var samlet, delte teknikerne kunnskap seg i mellom. Hver enkelt fikk da tilført kunnskap og erfaring som de andre hadde. På samme måte har jeg vært vitne til dette med de forskjellige databaseadministratorene.

For utenforstående kan slike fortellinger virke banale og en lite effektiv måte å utveksle kunnskap på, jmf. T&W der ledelsen så på "Bardic Tradition" som en merkelig praksis. Men for de som deltar i praksisfellesskapet bidrar historiene til mer kunnskap, skaper en sosial fellesfølelse og kan dermed ses på som en kulturell brobygger.

Både Orr (1996), Brown og Duguid (1991) og Ellingsen og Monteiro (2003) knytter fortellinger sterkt opp mot distribusjon av kunnskap, og prosessen rundt dette. Fortellinger er for eksempel hovedkomponenten i Orr sin beretning om hvordan kopimaskinreparatører jobber og løser problemer. Gjennom fortellingene delte de erfaringer med hverandre, for så å sammenlikne denne kunnskapen sammen med tidligere erfaring og gjennom diskusjon skapes det ny kunnskap i form av løsninger på problemene. T&W på sin side fortalte om en organisasjon der de hadde såkalte "Bards". Dette var personer med mye kunnskap om organisasjonen som fungerte som levende kunnskapsbaser av organisasjonell kunnskap. Denne kunnskapen var ikke kodifisert, da den hadde blitt gitt til en "Bard" ved hjelp av fortellinger. På samme måte fikk ansatte kunnskap fra slike "Bards".

Når det gjelder mine observasjoner så finner man slike fortellingstradisjoner flere steder i organisasjonen. Det som skilte seg mest ut var hvor sentral plass fortellinger hadde i seksjonsgruppemøtene. Her ble fortalte alle om sine problemer i form av statusoppsummeringer, og på bakgrunn av dette startet en diskusjon med de andre i

seksjonen om mulige løsninger. De mest erfarne tok frem sine erfaringer i lignende situasjoner. Disse situasjonene lignet veldig på de Orr skisserer i sitt arbeid.

#### **6.1.4 Kontekst**

Her vil jeg bruke Blackler sin typologi for å plassere observasjonene mine inn i forskjellige kategorier av kontekstuelle elementer.

##### **6.1.4.1 Hodebårne (*embrained*) kontekstuelle komponenter**

Blackler definerer hodebåren kunnskap som "knowledge that is dependent on conceptual skills and cognitive abilities" (1995, p. 1023).

Thompson og Walsham viser til psykometri, der en gjør en analyse av psykologiske tester. Dette hadde jeg dessverre ikke tid, ressurser eller kompetanse til under mine feltstudier. Dessuten er det også et spørsmål om dette hadde blitt godtatt av personene jeg studerte. Når det er sagt, så la jeg merke til spesielt et par personer i gruppen som hadde veldig gode evner til å ta til seg all informasjon som ble gitt på møter og sette denne i sammenheng der og da for så å stille kritiske spørsmål til foredragsholder. De samme personene var også de som hadde best oversikt over gruppens tjenestetilbud, som var begynt å bli relativt kompleks med mange avhengigheter. Dette viser deres latente anlegg til å danne seg komplekse regler og forstå komplekse sammenhenger. Et eksempel på slik kunnskap er programmeringskunnskapene til de forskjellige personene i seksjonen. Deres evne til å se løsninger på komplekse problem og overføre dette til algoritmer. Dette er kunnskap som de kunne overføre til andre også.

##### **6.1.4.2 Kroppsbårne (*embodied*) kontekstuelle komponenter**

Den andre kategorien er kroppsbåren kunnskap og defineres som "action oriented and [.....] likely to be only partly explicit" (1995, p. 1024).

De læresituasjonene som oppstod mellom personer på kontor, der det jobbet en erfaren og en nyansatt kan sies å være av en kroppsbåren kontekstuell form. Det blir et slags lærling eller

fadderordning der den erfarne personen forteller og viser hvordan man utfører visse oppgaver. Som beskrevet i delen om arbeidet, så fikk jeg selv hjelp og sammen gikk jeg og "læremesteren" gjennom en sak. Det var ikke så mye mangel på teori som gjorde at jeg måtte ha hjelp, men mer å se hvordan man gjorde oppgavene i praksis.

### **6.1.4.3 Kulturbårne (encultured) kontekstuelle komponenter**

Den tredje kategorien som Blackler identifiserer er kulturbåren kunnskap. Han mener at dette er kanskje den mest problematiske rent analytisk. Kategorien er definert som "the process of achieving shared understandings" (1995, p. 1024). Praksisfellesskapene som jeg studerte er representanter for denne kategorien. Felles for alle fellesskapene var at gjennom en sosial deltagelse i fellesskapet gjorde at man alle fikk en mer eller mindre felles forståelse av gruppen og dens interesseområde. Selve seksjonen er et eksempel på et slikt fellesskap, der alle diskusjoner på seksjonsmøtene var med på å bygge opp en felles kultur.

### **6.1.4.4 Nedfelt (embedded) kontekstuelle komponenter**

Fjerde kunnskapskategori er nedfelt kunnskap og Blackler definerer denne som "analyzable in systems terms, in the relationships between, for example, technologies, roles, formal procedures, and emergent routines" (1995,p. 1024).

Her er det mest naturlig å rette fokus på de mange praksisfellesskapene som var i IT-avdelingen, der hvert fellesskap hadde sin egen kultur og praksis. Gjennom min deltakelse i disse sosiale nettverkene fikk jeg enormt mye kunnskap, og senere kunne bidra med kunnskap selv. Praksisfellesskapene skapte formet etter hvert rutiner og regler, og det er disse som er de nedfelte kontekstuelle elementene i organisasjonen.

#### **6.1.4.5 Symbolbårne (encoded) kontekstuelle komponenter**

Den siste kategorien for kunnskap som Blackler identifiserer er symbolbåren og defineres som "information conveyed by signs and symbols" (1995, s. 1025). Denne er samtidig den mest eksplisitte formen.

Arbeidsrutiner og beste praksis er eksempler fra CERN på denne kategorien. Blant annet inneholder TWIKI systemet (som ble beskrevet gjennom arbeidet) rutiner og prosedyrer for hvordan systemer skal installeres, standarder som brukes, personer som er ansvarlig for tjenestene.

#### **6.1.5 Epistemiske kulturer**

Karin Knorr Cetina (Cetina, 1999) har gjort en studie der hun sammenligner fysikkmiljøet på CERN opp mot molykuler biologi. Hovedfokuset hennes var å se på hvordan de to forskjellige miljøene utførte forskningen, da særlig med tanke på epistemiske kulturer.

*"It is surely surprising that experiments of this size work at all; and they work, by scientific standards, very successfully. For a sociologist it is even more surprising that the ones observed work in nonbureaucratic ways, without overbearing formal organization, without hard-set internal rules, and without the management problems apparent in industrial organizations of comparable size.*

(Cetina 1999, p 160); beskriver ATLAS prosjektet på CERN.

Jeg vil her gjøre en kort sammenligning mellom noen av resultatene jeg fant, med Karin Knorr Cetina sine oppdagelser. Da jeg i hovedsak har sett på miljøet innen IT-avdelingen, og Cetina undersøkte fysikkmiljøet på CERN, så kan det være noen forskjeller i utgangspunktet. Likevel mener jeg at det er relevant å sammenligne disse miljøene. Blant annet er IT-avdelingen en naturlig del av CERN som organisasjon, og hele organisasjonen jobber mot et mål. Samtidig har de fleste i IT-avdelingen kombinert utdanningen sin med noe fysikk, og faktisk er mange utdannet tekniske fysikere. Et annet punkt er kontakten som IT-avdelingen har med fysikerne. Siden de er den største "kunden" til IT-avdelingen, så har IT mye kontakt med fysikermiljøet i

form av ukentlige møter, prosjekter osv. CERN arrangerer også en rekke forelesninger der personer blir oppdatert på fysikkmiljøet.

Cetina sammenligner fysikkmiljøet med Open Source miljøet. Dette gjør hun av en rekke grunner. Blant annet er det mindre byråkrati, og en mindre hierarkisk organisasjonsform. Det mer demokratisk, og det er sjelden en spesifikk person får æren for det som er oppnådd. For eksempel er det flere hundre medforfattere på publikasjoner som omfatter de største prosjektene. Av den grunn er heller ikke forskningen så personfiksert, i forhold til molekylær biologi som ofte operer med laboratorieforsøk med få personer.

Når det gjelder det åpne og lite hierarkiske miljøet som Cetina beskriver, så stemmer det godt overens med mine observasjoner som teknisk student i IT-avdelingen. Blant annet var det en av de tekniske studentene som sa

*”Det som er merkelig er at jeg ikke får noen instruksjoner om hvordan man skal løse en oppgave som jeg får tildelt”*

*Teknisk student*

## **6.2 IT systemers rolle for å støtte distribusjon av kunnskap**

Så hvordan kan observasjonene som jeg gjorde på CERN brukes til å bedre forstå IT-systemers rolle til å støtte distribusjon av kunnskap? Selvsagt er det vanskelig å generalisere ut i fra bare en feltstudie, men gjennom å kombinere disse observasjonene sammen med teorien som ble gjennomgått, vil jeg nå tegne opp en del av utfordringene som finnes, og til slutt komme med noen anbefalinger på hvordan IT-systemer kan bli brukt.

### **6.2.1 Maktforhold**

Maktforhold er viktig å være oppmerksom på, og kan påvirke mange deler av en organisasjon. Hvis et system endrer på maktforholdet i en organisasjon, så vil de som har en maktfordel fra før mest sannsynlig ikke ta i bruk systemet. Det kan også være en del skepsis mot overvåkning fra ledelsen i slike system, og ansatte sin redsel for å gjøre noe feil. Som beskrevet i

observasjonen så ble det forsøkt å innføre et nytt system kalt Jira for prosjektstyring og feilrapportering i avdelingen. Under min observasjonsperiode var det kun to personer som tok i bruk dette systemet. Selv om ingen sa noe direkte om at de var skeptiske til et system som var pålagt å brukes fra ledelsen i IT-avdelingen, så var det ingen begeistring over dette nye systemet. Dette er også et av momentene som Walsham (2001) fremhever som problematisk. I første omgang er det å være klar over at dette kan være et problem et skritt i riktig retning. Walsham foreslår at det lages trygge begrensede områder for elektronisk debatt. Dette kan fungere bra, men det kommer selvsagt an på en del andre faktorer. En ulempe er at kunnskap bare blir delt mellom personene innenfor disse "trygge" områdene, mens alle andre ikke har mulighet til å være med. Da begrenser man en del av ideen med kunnskapsdeling. Man kan også få missnøye med de som ikke får være med, for eksempel at lederne tror at ansatte har noe å skjule. Alternativt til å lage slike trygge diskusjonsfora er å ha en aktiv dialog med ledelsen, der det blir lagt vekt på tillit. Dette er enklere i teorien enn praksis, men kan fungere i noen organisasjoner.

En annen maktårsak til at folk ikke tar i bruk systemet kan ligge i individuelle bonusordninger. Walsham (2001) viser til et eksempel der ansatte i et reklamebyrå var veldig skeptiske til å dele sine ideer med andre ansatte. Grunnen til dette var et bonussystem som baserte seg på individuell belønning, og derfor ble det en konkurransesituasjon der alle beskyttet sin egen kunnskap. Orlikowski pekte også ut belønningssystemer, prosedyrer, kultur og normer i organisasjonen som faktorer som kunne påvirke innføringen av gruppevare. Ut fra mine observasjoner og litteratur (blant annet (Cetina, 1999)) vil jeg hevde at CERN har lite fokus på prestasjonen til hver enkelt, og heller fokuser på resultatene som oppnås som gruppe og organisasjon. Som et eksempel kan det nevnes de flere hundre personene som blir kreditert i artikler fra fysikkforsøkene ved CERN.

### **6.2.2 Nytteverdi**

Jeg vil argumentere for at et system skal bli tatt i bruk i en organisasjon, så må personene i organisasjonen se nytteverdien av systemet. Systemet bør bidra med fordeler for alle i organisasjonen, og ikke bare favorisere en gruppe (for eksempel ledere). Hvis en ansatt må

bruke mer tid og ressurser på et system uten at han har noen direkte nytteverdi av det, så er risikoen stor for at han slutter å bruke systemet.

### 6.2.3 Kontekst

T&W (Thompson & Walsham, 2004) viser at all kunnskap blir tolket og skapt av den konteksten som er i det øyeblikket. Mine resultater viste også at de forskjellige kontekstuelle komponentene påvirker hvordan kunnskap ble skapt og tolket.

### 6.2.4 Problemet med å samle all kunnskap i et system

Tradisjonelt innen kunnskapsforvaltning har det lenge vært fokus på hvordan man kan samle all kunnskap i et stort system, gjerne et databasebasert system. Det har til tider blitt fokusert mer på teknikken som "kunnskap", enn at den faktisk bare er verktøyet som skal brukes for å støtte opp om kunnskapsutveksling. I de siste fem til ti årene har vi heldigvis sett en endring, og stadig flere organisasjoner har funnet ut at et slikt at mange ting som spiller inn på et slikt system. Før jeg begynte på CERN hadde jeg dannet meg noen forestillinger om at et slikt høyteknologisk laboratorium sikkert brukte veldig avanserte systemer, som ofte var selvutviklet av ekspertene som jobbet der. Dette var også forestillingene mine om IT-avdelingen og jeg så for meg avanserte IKT systemer for å støtte deling av kunnskap i et miljø preget av mange forskjellige nasjonaliteter, kulturer og fagpersoner. Virkeligheten ble en litt annen.

Som vist i resultatdelen, så var majoriteten av "systemene" som ble brukt til kunnskapsutveksling svært enkle. Epost, søking med Google, og lynmeldingsprogrammer (MSN Live messenger og Kadu Kadu) var noen av hjelpemidlene for kunnskapsdistribusjon på CERN som skilte seg mest ut.

Epost var en av de viktigste kildene til diskusjon, planlegging, videreformidling av prosjektdata, problemløsning etc. Kort sagt var epost veldig viktig for all kommunikasjon med andre personer innad på CERN og med omverden. Man kan spørre seg hvorfor så "enkle" system ble brukt for eksempel til å diskutere programmeringsløsninger for avanserte systemer. Noe av svaret vil jeg si ligger i akkurat enkelheten og tilgjengeligheten for alle



personer. En epostklient var noe alle hadde på sin maskin, og er operativsystemuavhengig på lik linje med websider. Siden dette er noe alle personer er vant til å bruke (i alle fall på en IT-avdeling), så gir det et godt utgangspunkt i motsetning til spesifikke samarbeidsapplikasjoner i form av groupware. Slike samarbeidsapplikasjoner er også en del mer avanserte, da de dekker samarbeid på en helt annen måte. Epost gir også de forskjellige brukerne valgfrihet i når de vil svare. Hvis en person er opptatt med noe så slipper han/hun bli avbrutt, og kan svare på eposten når man får tid.

En stor ulempe med eposter er den litt kaotiske organiseringen som blir av dem, og at de er mer utilgjengelige for andre enn den som eier dem. Det kan være vanskelig å finne frem til nettopp den eposten som spesifiserer datastrukturen i en viss komponent. Selvsagt kan man søke gjennom alle epostene i arkivet, men er tungvindt og gir ikke alltid resultat. Man må for eksempel vite hva man skal søke etter, gjerne eksakte ord. En annen ulempe er at eposter i epostlister vanligvis kun er tilgjengelig for personer som abonnerer på den spesifikke listen, så alle andre vil ikke ha tilgang på den kunnskapen disse representerer. Noen diskusjoner er også vanskelig å ta over epost, da det kreves helt andre egenskaper av mediet som benyttes.

I det siste tiåret har det vært mye fokus på Open Source prosjekter, og deres suksess i relativt avanserte og store programvareprosjekt. Gutwin med flere (2004) gjennomførte en studie av tre slike prosjekter, der det ble sett på hvordan distribuerte utviklere opprettholdt gruppeforståelse. Det ble sett på hvordan prosjektet ble koordinert ved å distribuere kunnskap om hvilke kodedeler de jobbet på, hva den enkelte gjorde og hvilke planer han/hun hadde. Til dette arbeidet ble det brukt epostlister og pratekanaler<sup>17</sup>. Begrunnelsen for dette var at da ble alle medlemmene bevisst på hva de andre i gruppen gjorde.

Det er viktig å understreke at epostlister og lynmeldingstjenester ikke er de eneste elektroniske verktøyene som ble brukt til å distribuere kunnskap i IT-avdelingen på CERN. Som nevnt før er det en rekke situasjoner som det ikke blir tilstrekkelig å bruke epostlister alene.

---

<sup>17</sup> Av engelsk chat. Eksempel på slike er IRC, MSN Live Messenger.

### 6.2.4.1 *Analyse*

Da hver organisasjon er unikt sammensatt av personer med forskjellig kunnskap, oppfatninger, kultur etc, så er det viktig å ta med seg dette når man skal se på systemer for å støtte distribuering av kunnskap i en organisasjon. For å få en lettere oversikt over organisasjonen, vil det være hensiktsmessig å gjøre en analyse av hvordan arbeidet gjøres. Da vil man avdekke allerede eksisterende praksis, maktforhold i organisasjonen og andre forhold som må tas med ved planlegging av et støttende system. Min feltstudie av arbeidet i IT seksjonen viser en rekke måter som kunnskap blir distribuert på i organisasjonen. Noen av disse hadde ikke vært mulig å oppdage uten å ha gjort en slik undersøkelse.

Et annet viktig poeng er å erkjenne at det ikke er teknologien i seg selv som er hovedmålet med et slikt system, men hvordan denne kan best supportere en slik kunnskapsprosess. Teknologien er bare et verktøy som muliggjør eller forbedrer prosessen. På den annen side er teknologien viktig på sluttresultatet, fordi man er avhengig av denne for å få til et teknisk fungerende system.

Det er også viktig at systemet støtter opp om eksisterende kunnskapsnettverk, både i form av praksisfellesskap eller andre. T&W advarer mot hva som skjer dersom en prøver å erstatte eksisterende kultur med nye retningslinjer og et nytt system. I eksempelet med A1, så ble fortellingstradisjonen til de ansatte fra selskapet som ble kjøpt opp sett på som en tungvind og uprofesjonell måte å dele kunnskap på. Da denne ordningen ble avviklet startet en rekke problemer, fordi ingen hadde full oversikt over hva de andre i avdelingen gjorde. Den gamle "Bardic tradisjonen" hadde vært limet som holdt organisasjonen oppe.

Walsham (2001) presenterer en rekke spørsmål til de som skal bruke IKT til å supportere kunnskap i en organisasjon. Dette gjøres nettopp fordi det ikke finnes noen eksakt fremgangsmåte for hvordan man skal gå frem, da hver organisasjon er forskjellig og krever sine egne metoder og prosesser. Det som er interessant er at Walsham stiller spørsmålet om hvordan IKT kan brukes til å støtte og forbedre deling av kunnskap i en organisasjon helt til slutt. Han viser tydelig at for å kunne svare på dette spørsmålet må man ha gått gjennom de andre på forhånd. Walsham sier selv at han med vilje har satt IKT spørsmålet til slutt, da han

mener en prosess med IKT baserte systemer for kunnskapsdeling bør begynne med de menneskelige prosesser som er involvert istedenfor teknologien.

*"Information and communication technologies are not the answer to improved knowledge-sharing within and between people and organisations."*

*Walsham (2001: p 607)*

Samtidig med dette er det viktig å understreke teknologiens rolle som støttende verktøy, og ikke noe som skal erstatte allerede eksisterende tradisjoner, kulturer etc. Istedenfor å erstatte eksisterende praksisnettverk som oppbevaringssted<sup>18</sup> for kunnskap, så må man støtte de eksisterende praksisfelleskapene og andre tradisjoner for å utveksle kunnskap.

---

<sup>18</sup> Av engelske uttrykket repositories



## 7 AVSLUTNING

På samme måte som det er vanskelig å komme med konsensus definisjon på kunnskap, så gjelder dette også eksakte fremgangsmåter for hvordan man best kan støtte utveksling av kunnskap ved hjelp av IKT.

I denne oppgaven har jeg sett på hvordan kunnskap distribueres i en høyteknologisk organisasjon. Observasjonene og analysen av disse har vist at arbeidet er overraskende lite styrt, og støttesystem for utveksling av kunnskap på langt nær så avanserte som man skulle tro, med tanke på de avanserte systemene som blir utviklet i organisasjonen. På bakgrunn av analysen av mine observasjoner har jeg her diskutert IT-systemers rolle i å støtte distribusjon av kunnskap, og hva som bør tas hensyn til når IKT baserte systemer skal brukes til dette. Jeg velger å kalle dette for forslag, da det er ingen fasit verken på hvordan et system kan støtte kunnskapsutvikling og distribusjon i en unik organisasjon som er komplekst sammensatt av unike enkeltindivider, kultur, praksiser, formelle regler etc. På bakgrunn av dette er det derfor er det ingen garanti for at en organiseringsmåte med et IKT basert kunnskapssystem som er vellykket for en organisasjon, vil fungere like bra for en annen.

Resultatene av observasjonene viser at distribusjon av kunnskap er en veldig kompleks prosess, som til stadighet endrer seg. Kunnskap blir ikke delt gjennom en kanal i organisasjonen, det skjer på tvers av polske nettverk, samtaler i kaffekroken og gjennom pratkanaler på nettet. Man må derfor ikke se på kunnskap som et objekt som skal fanges, kodifiseres og lagres i en stor kunnskapsdatabase. Kunnskap og det å kunne må behandles med tanke på alt rundt, og derfor er det lurt å begynne med å se på hvordan kunnskap som prosess foregår.



## 8 KILDER

- A, M., Fielding, R., & Herbsleb, J. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM ToSEM* 11(3) , 309-346.
- Anastas, J., & MacDonald, M. L. (1994). *Research Design for Social Work and the Human Services*. New York: Lexington.
- Brown, J. (1998). Internet technology in support of the concept of 'communities-of-practice': the case of Xerox. *Accounting, Management and Information Technologies* 8(4) , 227-236.
- Brown, J., & Duguid, P. (1991). Organizational learning and communities-of-practise: toward a unified view of working, learning and innovation. *Organization Science*, 2(1) , 40-57.
- Cetina, K. K. (1999). *Epistemic cultures : how the sciences make knowledge*. Cambridge, Mass.: Harvard University Press.
- Ciborra, C. (1994). The grassroots of IT and strategy. In C. C. (eds.), *Strategic information systems: A European perspective* (pp. 3 – 24). New York: John Wiley & Sons Ltd.
- Collins, H. (1993). The structure of knowledge. *Social Research* 60: 95-116.
- Day, R. E. (2005). Clearing Up "Implicit Knowledge": Implications for Knowledge Management, Information Science, Psychology, and Sosial Epistemology. *Journal of the American Society for Information Science and Technology*, Vol 56, N 6 , 630-635.
- Ellingsen, G., & Monteiro, E. (2003). Mechanisms for producing working knowledge: enacting, orchestrating and organizing, Information and Organization. *Information and Organization* 13 , 203–229.
- Foster-Johnson, E. (2005). *RPM Guide*. Retrieved 03 15, 2007, from Fedora Project: <http://docs.fedoraproject.org/drafts/rpm-guide-en/>
- Gutwin, C., Penner, R., & Schneider, K. (2004). Group awareness in distributed software development. *CSCW 2004* (pp. 72-91). Chicago: ACM Press.

---

Hoffmann, H. (2007). *The Role of Global Communication in Particle Physics*, rev. 1., (pp. 1-16). Berlin.

Klein, H., & Myers, M. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly*, Vol 23, No. 1 , 67-93.

Lave, J., & Wenger, E. (1991). *Situated learning, Legitimate Peripheral Participation*. Cambridge: Cambridge University Press.

Nonaka, I., & Takeuchi, H. (1998). A Theory of the Firm's Knowledge-Creation dynamics. In A. Chandler jr, Hagstrøm, & Ø. (. Sølvell, *The dynamic firm. The role of technology, strategy, organization and regions* (pp. 214-241). Oxford: Oxford Univ. Press.

Oracle. (2007, Mai 23). *How To Clone An Existing RDBMS Installation Using OUI Note:300062.1*. Retrieved from Oracle Metalink: <http://www.metalink.oracle.com>

Orlikowski, W. (2002). Knowing in Practice: Enacting a Collective Capability in Distributed Organizing. *Organization Science* , 249-273.

Orr, J. (1996). *Talking about machines : an ethnography of a modern job*. Ithaca, N.Y.: Cornell University Press.

Polanyi, M. (1969). *Knowing and being: Essays by Michael Polanyi*. Chicago: University of Chicago Press.

Polanyi, M. (1962). *Personal Knowledge*. Chicago: The University of Chicago Press.

Polanyi, M., & Prosch, H. (1975). *Meaning*. Chicago: The University of Chicago Press.

Robson, C. (2002). *Real world research, Second Edition*. Oxford: Blackwell publishing.

Stenmark, D. (2002). Information vs. Knowledge: The Role of intranets in Knowledge Management. *Hawaii International Conference on System Sciences* (pp. 1-10). Hawaii: IEEE.

Thompson, M., & Walsham, G. (2004). Placing Knowledge Management in Context. *Journal of Management Studies*, 41:5 , 725-747.



Tsoukas, H. (2005). Do we really understand tacit knowledge? In H. Tsoukas, *Complex Knowledge: Studies in Organizational Epistemology* (pp. 141-163). Oxford: Oxford University Press.

Tuomi, I. (2001). Internet, innovation and open source: actors in the network. *First Monday* , 1-20.

Walsham, G. (1996). Knowledge management: The benefits and limitations of computer systems. *European Management Journal* , 599-608.

Wenger, E. (1987) *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. San Francisco: Morgan Kaufmann.

Wenger, E. (2004). Knowledge management as a doughnut: Shaping your knowledge strategy through communities of practice. *Ivey Business Journal* , 1-8.