

**Universitet i Oslo  
Institutt for informatikk**

**INF5261 V09**

**Sluttrapport**

**Bluetooth Messaging Service**

Kristian Sporsheim  
Rolf Erik Normann  
Karsten Jansen

**5. Mai 2009**



## **INNHOOLD**

1.0 INNLEDNING .....	3
2.0 TEORI .....	4
2.1 Kontekst-sensitivitet i det foranderlige rommet .....	4
2.2 Tilpasning til de digitale økologier .....	4
2.3 Kontekst-sensitiv kommunikasjon.....	5
2.4 Personlig tilpasning av informasjon.....	6
3.0 PERSONAL AREA NETWORK.....	8
3.1 Teknologier i PAN.....	8
3.2 Infrared Data Association .....	8
3.3 Ultra-wideband(UWB) .....	9
3.4 ZigBee.....	9
4.0 BLUETOOTH .....	10
4.1 Piconet.....	10
4.2 Vårt prosjekt.....	10
4.3 Eksisterende prosjekter .....	11
4.4 Sikkerhet .....	12
4.5 Blåtann som teknologi i framtiden.....	13
5.0 PROTOTYPE .....	14
5.1 Utvikling av Prototypen.....	14
5.2 Testing av prototype .....	16
5.3 Visuelt eksempel av prototype.....	16
6.0 BRUKERUNDERSØKELSER .....	18
7.0 OPPSUMMERING .....	20

8.0 REFERANSER..... 21

9.0 APPENDIX - PROTOTYPENS KILDEKODE..... 23

## 1.0 INNLEDNING

Vi ønsker med vår prosjektoppgave å finne ut om mobiltelefonbrukere er interessert i å motta relevant informasjon ut ifra hvilken lokasjon de befinner seg på. I denne oppgaven har vi sett på nærhetsnettverk og spesielt blåttann som en aktuell teknologi for dette. Vi ønsker å utnytte oss av denne teknologien siden den er meget utbredt. Dessuten er den virksom over kortere distanser, og vi kan derfor med rimelig sikkert vite hvilken lokasjon brukeren befinner seg på.

Vi vil undersøke hva som gjør blåttann til et aktuelt kommunikasjonsmedium for oss og hvorfor den kan være aktuell for brukerne. Dette går både på det tekniske og brukervennlige, samt villighet til bruk av tjenesten. Linjen mellom informasjon og spam kan være tynn og hva en bruker ser på som relevant informasjon kan variere. Vi vil undersøke om noen lignende prosjekter har blitt gjort før, eventuelt hva vi kan gjøre annerledes. Vi vil også diskutere andre kommunikasjonsmedier som vi kunne ha brukt.

Sikkerhet er et viktig tema, og vi må undersøke i hvor stor grad det er mulig å kunne misbruke teknologien. Vi ønsker også å finne ut om blåttann er en teknologi som vil utvikle seg eller om det er andre teknologier som tar over, og hvor lenge den eventuelle prototypen vi lager kommer til å være aktuell.

Vi var fra oppstarten av dette prosjektet klare på at vi ønsket å lage noe konkret i løpet av semesteret, og ikke bare gjorde empiriske undersøkelser og refleksjoner. Vi ønsket derfor å ende opp med en fungerende prototype av en eller annen sort, og foreta testing av denne på potensielle brukere for å kunne evaluere den på best mulig måte. I tillegg til en fin motivasjonsfaktor, kan en fysisk implementasjon tjene som et fokus for tankene våre, og hjelpe til å koble teori opp mot prosjektet.

## **2.0 TEORI**

### **2.1 Kontekst-sensitivitet i det foranderlige rommet**

Trådløse informasjons tjenester kompliserer måten vi er nødt til å så på kontekst sensitive tjenester. Dette gjør at vi er nødt til å se problemet på en helt annen måte. Kontekst har et fysisk aspekt, men også institusjonelle sosiale regler i tillegg til lokal sosial praksis. Men hva menes det egentlig med at en maskin er klar over sin egen kontekst[1].

Hvilken kontekst vi er i har forandret seg etter at mobiltelefonen er blitt et allemannseie. Et godt eksempel på dette er hvordan mobilen kan forandre konteksten i det å gå på et teater. Konteksten på et teater kunne før blitt definert som små sett med sosiale relasjoner mellom mennesker, men i dag med en mobiltelefon har en person på teateret muligheten til å kontakte hvem som helst, bare den andre personen også har med seg en mobiltelefon.

Blåtann kan være nyttig fordi den initierer kommunikasjon mellom to enheter basert på at de er i nærheten av hverandre. Hvis vi ser en annen teknologi som for eksempel bruke GPS til å finne ut om enheter er i nærheten av hverandre, så er den på en annen side nødt til å tenke at en enhet er plassert i posisjon(X,Y) og en annen enhet er posisjonert i (X,Y), altså er de i nærheten av hverandre. Blåtann enheter vil derfor på dette området gjøre det enklere fordi de kun er avhengig av og være innenfor rekkevidde.

### **2.2 Tilpasning til de digitale økologier**

Allestedsnærværende (ubiquitous) datamaskiner vil bli mer og mer vanlig, med datakraft overalt, og informasjon tilgjengelig i stadig større omfang og på flere og flere steder. Men i takt med utvidelsen av bruksområdene til digitale medier og applikasjoner, og utbredelsen av elektroniske gjenstander, ser vi en økning i antall tjenester en gitt gjenstand tilbyr – en mobiltelefon er gjerne også et kamera, en musikkspiller, en kalender og en spillmaskin, for å nevne noen. Dersom man har et kamera på mobiltelefonen sin, vil man kanskje ikke kjøpe en PDA-modell som har kamera.

I tillegg er det i skapelsen av nye teknologier eller kommersialisering av disse, rivaliserende standarder. Valgene av slike standarder under utviklingen av en tjeneste, samt brukernes muligheter – både når det kommer til hva de har råd til og hva slags gjenstander og teknologi de allerede eier – bestemmer hvilke, og hvor mange brukere man får til tjenesten.

Man kan altså si det er verdifullt å forstå mekanismene bak forbrukernes valg. Et forsøk på å gjøre dette kommer fra Jung et. al. (2008) [2] hvor de lanserer begrepet *økologi* om den samlede mengden digitale gjenstander en person eier og bruker, og samspillet mellom disse. Ofte og oftere er gjenstander ikke bare frittstående, men deler av et større nettverk som gir synergier, og som ikke ville blitt brukt på samme måte eller i det hele tatt hvis den sto alene.

Eksempelvis vil en bruker som er interessert i digitalfotografi anskaffe gjenstander som arbeider sammen for å forsterke denne interessen. Et digitalkamera, en pc for redigering og en fotoprinter vil da essensielt være et "nettverk" av gjenstander. Anskaffelse eller utbytting av noen av komponentene vil være avhengig av de andre.

En økologi er langt på vei individuell, og en gitt bruker vil ha en unik økologi og en unik bruk av gjenstandene sine. Kombinasjoner av gjenstander gir uante muligheter, kanskje langt utenfor hva designeren i sine villeste fantasier hadde tenkt seg.

For å få flest mulige brukere må en gjenstand passe inn i flest mulige økologier. En måte å få til dette på, kan være å tilrettelegge for mange mulige måter av bruk, selv slike man ikke ser umiddelbar nytte av selv. I tillegg kan det synes lurt å velge å benytte utbredte standarder, for eksempel ikke-proprietære.

En informasjonstjeneste i det offentlige rom, slik som vår prototype eventuelt kan utvikle seg til å bli, kan ses på som en digital tjeneste som har potensiale til å passe inn i mange slike økologier. For å utfylle dette potensialet, bør designvalgene reflektere en gjennomtenkt vurdering av punktene ovenfor.

## **2.3 Kontekst-sensitiv kommunikasjon**

Vi vil tilpasse informasjonen vi gir ut så godt som mulig. Dette gjøres ved å vite så mye som mulig om brukeren, og også så mye som mulig om plassen og miljøet brukeren beveger seg i. Schilit et. al. (2002)[3] forklarer hvordan «context awareness», det å sette eller la applikasjoner sette og bruke kunnskap om lokasjon, miljø, hvem som er i nærheten og hva man driver med, kan bedre kommunikasjon og bryte barrierer for kommunikasjon mellom grupper og individer.

Schilit et. al. analyserer context aware kommunikasjon over to dimensjoner, på et spekter fra manuell til automatisk. «Context acquisition» dreier seg om hvordan systemet fanger opp konteksten. En manuell context acquisition tilsier at en menneskelig operatør må angi konteksten, mens automatisk tilsier at systemet gjør det uten hjelp. Et system som Olivettis Active Badge er en brikke som de ansatte bar på seg, og som gav signaler til sensorer rundt om i bygningen. Slik hadde systemet rede på hvor de ansatte til enhver tid befant seg, og kunne handle utfra dette.

«communication action» vil si hvordan systemet agerer på kontekstuell informasjon. Manuell vil si at en menneskelig operatør velger og utfører en kommunikativ handling, mens automatisk vil si at systemet gjør dette helt selv. Olivetti hadde Active Badge, men telefonsystemet deres var manuelt, slik at en sentralbordoperatør ga beskjeder og rutet samtaler manuelt ut ifra informasjonen fra Active Badge-systemet. Senere ble et helautomatisk system implementert, men dette førte med seg egne problemer. Som artikkelforfatterne sier, det er meget vanskelig å automatisere begge dimensjoner på en tilfredsstillende måte, fordi den menneskelige sunne fornuft forsvinner. Det helautomatiske systemet kunne for eksempel ikke finne på å ringe de nærmeste kollegaene dersom den aktuelle ansatte ikke tok telefonen.

Kontekst er noe mer enn lokasjon. Dess flere relevante fakta om miljøet og brukeren, dess mer målrettet og kraftfull kan informasjonen bli. En automatisk innhenting av slik informasjon kan være meget vanskelig, argumenterer Schilit, fordi man ikke kan se hva som foregår inne i hodet på folk. En maskin kan vanskelig avgjøre om en person er deprimert eller fornøyd, sulten eller mett, trøtt eller opplagt. En manuell innhenting er en annen sak. Et systems handlinger kan lett tilpasses en brukers møteplan. En bruker kan til og med angi hvilket humør han eller hun er i.

## **2.4 Personlig tilpasning av informasjon**

For å forsterke potensialet til en informasjonstjeneste, kan det være nyttig å tilpasse informasjonen til hver enkelt bruker. En bruker som er interessert i lyd og bilde, kan ved inngangen av et kjøpesenter få informasjon om hvor nærmeste hi-fi-butikk befinner seg. En som er interessert i action-filmer, men ikke romantiske komedier, kan få informasjonen han ønsker seg når han går inn i en kino. Dette blir en mer brukervennlig tjeneste.

Hvordan man skal tilpasse slik informasjon kan være en utfordring. Hvordan skaffe seg slik informasjon? Og hvordan gi riktig informasjon til riktig person? En måte kan være å registrere bruksmønstre og/eller preferanser i en sentral database, og så tilpasse informasjonen hvis systemet gjenkjenner den aktuelle brukeren. Noe lignende ble gjort av Müller et. al. (2008)[4] og deres system MobiDiC, som er et system for å gi lokaltilpasset reklame i offentlige terminaler. En bruker kan laste ned en veiviser-applikasjon via blåtann, som vil løse brukeren til en gitt butikk / plass. Systemet lagret opplysninger som brukerens mac-adresse og den ønskede destinasjonen. Andre terminaler som er spredt rundt i byen scannet kontinuerlig etter blåtann-enheter, og hvis de fant en enhet med samme mac-adresse som den som nettopp hadde lastet ned applikasjonen, koblet de seg til enheten og oppdaterte veibeskrivelsen.

Hvorvidt folk vil godta at systemet lagrer informasjon om bruk som kan spores tilbake til dem er uvisst, men en indikasjon kommer på en litt annen måte fra den samme undersøkelsen. De

offentlige displayene viste veibeskrivelsen på større skjermer, og brukerne var likegyldig til om andre mennesker så hvor de skulle. Dette var uavhengig om displayet viste veien til et toalett eller en kafé. Unntaket var hvis bruken skjedde sent på kvelden, da ville ikke folk vise andre hvor de skulle gå av frykt for å bli overfalt. Utenom dette viste det seg altså at folk ikke var så opptatt av privacy som man kunne tenkt seg.

Allikevel er det selvsagt uvisst om disse funnene holder utenfor en forsøkssammenheng, eller når det kommer til annen informasjon enn hvor de skulle. Om brukere kan tenke seg å registrere preferanser, vil vi spørre om i brukerundersøkelsen.

En annen løsning på problemet kan være å la brukerne ta med seg preferansene sine selv. Rhodes et al. (1999)[5] tar for seg noen av problemene med lokaliserte ressurser, som at brukerinformasjon som lagres eller hentes via en lokalisert ressurs, kan være tilgjengelig for utenforstående, og at slik informasjon kan være håpløst utdatert når brukeren treffer systemet etter en tid. De foreslår å kombinere styrkene ved ubiquitous og wearable computing, og slik eliminere svakhetene.

Ved å bære med seg enhver personlige preferanser på sin personlige device, kan man ha denne oppdatert hele tiden, man slipper å registrere den noe sentralt sted, og man har full kontroll på når og hvordan denne informasjonen blir brukt. Måten Rhodes et. Al foreslår å gjøre dette på er ved et distribuert system av software-agenter som kjører på alle devicer, og som snakker sammen og forhandler om bruk av ressurser, data og tjenester. Ulempen ved dette er at hver device må laste ned og kjøre dette systemet, det er ikke noe som ligger standardisert fra fabrikken.



## **3.0 PERSONAL AREA NETWORK**

Personal area network(PAN) er et nærhetsnettverk som kan brukes til kommunikasjon mellom for eksempel datamaskiner og telefoner. Det eneste kravet er at de deler samme teknologi og at de er i nærheten av hverandre. Enkelte teknologier krever at de må være innen synlig rekkevidde mens de fleste krever bare at man er i en rekkevidde fra 1-100meter.

En av grunnene til at man kaller det PAN er at rekkevidden sjelden er mer enn 10 meter, men enkelte teknologier har en rekkevidde opp til 100m. PAN kan brukes til kommunikasjon innen nettverket selv(to eller flere enheter) eller for å koble til et annet nettverk som er tilkoblet internett, skrivere og lignende. Kommunikasjon mellom enheter er ofte brukt til headset, streame musikk, overføre filer og mye mer[6].

### **3.1 Teknologier i PAN**

PAN er ikke nødvendigvis trådløs, for ved bruk av USB eller firewire kan du enkelt sette opp ditt eget PAN-nettverk. Mac-brukere har god kjennskap til bruk av firewire for å kunne overføre filer mellom to maskiner.

Når teknologien blir trådløs bruker man navnet Wireless personal area network eller WPAN(IEEE 802.15) [7]. Ved bruk av teknologier som IrDA, Bluetooth, UWB, Z-Wave og ZigBee kan man enkelt sette opp et WPAN, mens enkelte av teknologiene er mer utbredt enn andre.

### **3.2 Infrared Data Association**

Infrared Data Association(IrDA) er en protokoll-standard for å kunne overføre data over korte avstander ved at man bruker infrarødt lys. IrDA har veldig kort rekkevidde og må ha fri sikt til enheten den prøver å koble til. Hastigheten kan variere fra 2.4kbit/s til 16MBit/s, men det er færre og færre enheter som har IrDa siden den blir byttet ut med blåtann i de fleste tilfeller. Den største grunnen til dette er nok at det er umulig å sende data uten at man har fri sikt under hele overføringen og at man ikke kan overføre til flere enheter samtidig[8].

### 3.3 Ultra-wideband(UWB)

UWB er en radioteknologi som er veldig strømsparende. Teknologien brukes til korte avstander og har en høy båndbredde. UWB bruker en stor del av radio-spektrumet, faktisk så bruker UWB en båndbredde på over 500MHz. Dette er fortsatt på forskingsstadiet og er ikke i kommersielt bruk i mobile enheter som mobiltelefoner og PDAer, men kanskje UWB kan løse noen av problemem som blåtann har[9].

### 3.4 ZigBee

ZigBee er spesifikasjon som passer høynivå kommunikasjonsprotokoller ved at man bruker lavvolts digitalradio som baserer seg på IEEE 802.15.4-2006 standarden for WPAN. Siden ZigBee har et veldig lavt effektforbruk så egner denne teknologien seg veldig bra til enheter med små batterier. ZigBee bruker *mesh networking*, som er en måte å rute pakker mellom noder som gir en høy driftssikkerhet og en lengre rekkevidde[10].

## 4.0 BLUETOOTH

Bluetooth, eller blåttann, er en av de mest utbredte trådløse teknologiene i dag. De aller fleste har en eller annen enhet som har blåttann installert, for eksempel mobiltelefonen, hodetelefoner eller en bærbar pc. Blåttann er en trådløs protokoll for å utveksle data over korte distanser mellom stasjonære og mobile enheter. Bluetooth har muligheten at den kan koble til flere enheter på en gang slik at man overkommer problemer med synkronisering. Rekkevidden på bluetooth er fra 1-100m avhengig av sender/mottaker [11].

Class	Maximum Permitted Power mW (dBm)	Range (approximate)
Class 1	100 mW (20 dBm)	~100 meters
Class 2	2.5 mW (4 dBm)	~10 meters
Class 3	1 mW (0 dBm)	~1 meter

### 4.1 Piconet

Piconet er navnet på Bluetooth PAN og kan bestå av 8 aktive enheter og kan ha veldig mange noder som er parkerte/ikke aktive. Oppbygningen i piconet består av master- og slavenoder, der den første noden er master mens resten av nodene er slave, og slavenodene kommuniserer med masternoden. Rekkevidden på et piconet er som regel 10m, men man kan lage nettverk på som har rekkevidde på 100m under perfekte forhold.

### 4.2 Vårt prosjekt

Vi valgte blåttann som teknologi siden den er en av de mest utbredte innen for mobilenheter og datamaskiner. Samtidig er blåttann veldig godt implementert i de fleste operativsystemer og mobiltelefoner. Disse punktene gjør at vi maksimerer antallet digitale økologier tjenesten passer inn i [2]. Blåttann gjør det enklere for oss å implementere vår ide, siden vi kan bruke eksisterende blåttannprotokoller. En annen viktig faktor er den begrensende rekkevidden på blåttann som gjør at

signalene holdes innenfor den aktuelle lokasjonen. Siden vi kun ønsker at brukere som er i den aktuelle lokasjonen skal få tilsendt informasjon som angår dem og ikke tilfeldig forbipasserende (F.eks utenfor en butikk).

Vårt system vil i utgangspunktet ha faste posisjoner som kan defineres når det blir satt ut i bruk. Konteksten i betydning lokasjon er dermed ikke noe som trenger å oppdateres eller sjekkes, men systemet kan bruke lokasjonen for å gi tilpasset informasjon. Systemet fanger også automatisk opp når noen er i nærheten, og sender informasjon til den aktuelle enheten dersom dette ikke har blitt gjort tidligere innen et definert tidsrom. Sett ut ifra dette kan vi kalle «context acquisition» og «communication action»[3] automatiske.

Vi må foreta noen valg når det kommer til implementasjon av systemet vårt. Hvordan skal for eksempel kommunikasjonen foregå? En løsning kunne være et system med skjermer og bruke disse til å velge hva slags informasjon en får, noe lignende MobiDiC-systemet [4]. Allikevel setter dette begrensninger på systemet, som ekstra kostnad, skalerbarhet og større kompleksitet. Ved å la en blåtann-enhet gi meldinger direkte til en mobil enhet, slipper vi denne forvanskningen.

Eventuell personalisering kan komme på andre måter. En distribuert programvare-løsning liknende den Rhodes et al [5] foreslår, krever en standardisering av mobile enheter til å støtte dette rett fra fabrikken, eller en nedlasting for førstegangsbrukere. Vi ønsker ikke å kreve noen ekstra nedlasting eller konfigurasjon, da dette kan skremme folk bort fra tjenesten, og det er dessuten et kompliserende ledd. Løsningen som ligger nærmere vår tankegang er å koble opp en mac-adresse og et mobiltelefonnavn, og på den måten få en potensielt unik bruker-id, noe ala det vi så i MobiDiC-systemet. Brukerne som ønsker det, kan registrere seg og sine preferanser, og få tilpasset informasjon.

### **4.3 Eksisterende prosjekter**

En viktig del av forarbeidet er å gå igjennom hva som har blitt gjort tidligere i lignende prosjekter. En kan være så innovativ en bare vil, men sannsynligheten for at noe lignende er blitt gjort før er veldig stor.

Overraskende nok trenger du ikke gå lenger enn til DeliDeluca for å finne et ferdig system som allerede er godt i bruk. Denne tjenesten bruker nettopp blåtann teknologien for å oversende mer eller mindre nyttig informasjon til en bruker. I enkelte DeliDeluca butikker i Oslo bruker de i dag denne tjenesten for å markedsføre forskjellige produkter. Det fungerer på den måten at skjermene kontinuerlig viser de utvalgte produktene, og i tillegg foreslår at du slår på blåtann for å mota

”gode tilbud”. Etter å ha aktivert blåttann får du etter kort tid en forespørsel om å ta imot en fil (video, flash, bilde, etc). Slik vi har erfart fungerer tjenesten overaskende bra til sitt bruk.

Når det gjelder markedsføring på mobil via blåttann fant vi fort ut at det var flere aktører som tilbyr ferdige løsninger. En av de mest profesjonelle aktørene vi kom over var den polske aktørene BluAir [12]. Dette firmaet tilbyr hele ferdige løsninger, de tilbyr hardware desktop enheter og mobile enheter som er laget for å kunne opprettholde mange blåttann kontakter. Og de tilbyr ferdige web grensesnitt hvor du kan styre hva blåttann enhetene skal sende ut til brukerne. Du kan også følge med på statistikker etc.

## 4.4 Sikkerhet

Siden blåttann overføres ved radiosignaler hvor alle være potensielle lyttere er det viktig med sikkerhet for å motvirke misbruk. I de siste årene har populariteten og bruken av blåttann-teknologi steget, men det har samtidig også interessen for å misbruke den.

I 2003 ble det av Adam Laurie fra A.L Digital Ltd oppdaget mangler i autentisering av dataoverføringsmekanismen i enheter med blåttann-støtte[13]. Og i 2004 kom det første viruset som spredde seg fra mobil til mobil ved hjelp av blåttann, det trengte da kun en godkjenning fra brukeren for å bli installert. I 2005 kom mobilviruset ”Lasco.A” som var laget for symbian OS mobiler. Dette viruset installerte seg selv og spredde seg automatisk etter installasjon.

Et av de enkleste grepene for en sikker blåttann forbindelse mellom to mobiltelefoner er å bruke autentisering[14]. Dette skjer enkelt ved å taste inn en avtalt PIN kode på begge enhetene. Siden begge enhetene må være innefor rekkevidde for at dette skal skje, er det sjeldent et problem, og det virker som en bra løsning. Samtidig er det ofte en dårlig implementasjon av dette som er det vanligste formen for blåttann-angrep. Et eksempel på dette skjedde i 2005 da milliardærarvingen Paris Hilton fikk mobilen sin hacket via blåttann, kontaktlisten og mobilbilder stjålet og lagt ut på nettet.

Det er oppstått en rekke metoder for å angripe en telefon gjennom blåttann-teknologien; blueSnarf, blueBug, blueSmack, blueStab, blueBump, bluesnarf++, blueSpooof, blueDump, blueChop, bluePrinting, blueTooone og bloover for å nevne noen. [15]

Vi lurer på om blåttann på mobil er en sikker teknologi i forhold til våres prosjekts prototype og hvilke måter det kan misbrukes på. Vi vil i et senere avsnitt presentere en undersøkelse der vi blant annet har spurt hva brukere mener om blåttann som en sikker teknologi.

## **4.5 Blåtann som teknologi i framtiden**

Blåtann er levende og utvikler seg. Digi.no meldte 23.april 2009 [16] at Bluetooth Core Specification Version 3.0 High Speed (HS) (Blutetooth 3.0) hadde blitt godkjent, og at bluetooth skulle bli enda raskere ved å ta i bruk wlan-protokollen 802.11 PAL. Maksimal hastighet skal nå kunne gå fra dagens 3megabits per sekund til hele 24megabits per sekund.

## 5.0 PROTOTYPE

Bluetooth messaging service er først og fremst en tjeneste som sprer informasjon i tilknytning til ulike lokasjoner, som for eksempel kino, kantine, parkeringshus osv. Er du på kino kan du få informasjon om eventuelle nye filmer og få filmtrailere sendt til din mobil. En viktig forutsetning er at brukeren alltid kan avslå en forespørsel og vil da ikke bli kontaktet på nytt fra samme lokasjon. Tanken bak tjenesten er at det ikke skal være en form for irriterende reklame, men til informasjon for brukeren. Selv om skille mellom reklame og informasjon ikke alltid er like klar, hadde vi satt oss et mål om at vi ikke skulle bli oppfattet som avsender av reklame eller SPAM. Dette er et punkt vi vil ta med i spørreundersøkelsen.

### 5.1 Utvikling av Prototypen

Når det kom til selve utviklingen av programmet var dette en prosess med mye prøving og feiling og veldig mye testing. For at tjenesten skulle fungere optimalt hentet vi informasjon om blåtann fra ca 40-50 forskjellige mobiltelefoner. Vi hentet ut informasjon om hvilke tjenester telefonen hadde og hva slags porter hver tjeneste befant seg på. Med dette fant vi fort ut at det er stor forskjell mellom telefoner fra samme leverandør og enda større forskjeller mellom leverandørene. Vi valgte at vårt fokus på telefoner skulle være Nokia [17], Sony Ericsson[18] og Samsung[19], siden disse er en av de største aktørene i mobiltelefonmarkedet. Telefoner som Apple iPhone har veldig dårlig blåtann-støtte, mens HTC har for streng sikkerhet slik at det ikke lot seg gjøre å sende beskjeder til telefonen.

Når det kom til valg av programmeringsspråk valgte vi å lage prototypen i Python grunnet god støtte for blåtann under Ubuntu og tilgangen til forskjellige biblioteker og programmer. Vi undersøkte også andre programmeringsspråk og observerte at det var fullt mulig å bruke Java og C, men dette krevde mye mer arbeid og gav ingen fordeler i ytelse eller funksjonalitet. Vi støtter oss på BlueZ [20], som er blåtann-biblioteket for Linux og PyBluez [21] som er pythonbiblioteket som er knyttet opp til BlueZ. For å kunne sende data bruker vi ussp-push[22] som er et lite program for å kunne sende filer via blåtann på OBEX Push protokollen.

Python programmet kjører på en stasjonærenhet som skal søke etter enheter som har aktivert blåtann. Programmet skal da returnere en Media Access Control(MAC)adresse og navnet på enheten. Vi kan også bestemme hvor lenge vi skal søke etter enheter.

Et kode eksempel på dette er:

```

nearby_devices = bluetooth.discover_devices( duration = 4)#Scann i 4 sekunder
print nearby_devices

```

Eksemplet returnerer en liste med par: `[("00:1C:B3:33:B5:68", "Test")]` der enheten sitt navn er "Test" og MAC-adressen er 00:1C:B3:33:B5:68. Det er MAC-adressen må vi ha for å kunne sende informasjon til enheten, men dette er ikke det eneste vi trenger.

For å kunne sende informasjon til en enhet trenger vi å få vite hvilke protokoller enheten har tilgjengelig. Blåtann er oppbygd slik at om en enhet er satt synlig, kan man spørre enheten om hvilke protokoller/tjenster den har å tilby. Her er et forenklet kodeeksempel der vi bruker MAC-adressen til enheten for å spørre enheten hvilke protokoller/tjenster den har:

```

service_matches = bluetooth.find_service(address = nearby_devices[0])
for services in service_matches:
    print " Name: %s" % (services["name"])
    print " Host: %s" % (services["host"])
    print " Description: %s" % (services["description"])
    print " Protocol: %s" % (services["protocol"])
    print " Provider: %s" % (services["provider"])
    print " Port: %s" % (services["port"])
    print " Service id: %s" % (services["service-id"])
print ""

```

Dette gir oss en pen oversikt over de forskjellige protokollene/tjenestene blåtann-enhetene har.

Et eksempel på en protokoll/tjeneste er:

```

Name: OBEX Object Push
Host: 00:18:13:47:0A:7F
Description: None
Protocol: RFCOMM
Provider: None
Port: 6
Service id: E006

```

Det er nettopp *OBEX Object push* vi kommer til å bruke for å kunne sende informasjonen vi ønsker via blåtann.

Denne informasjonen har vi hentet fra de forskjellige mobiltelefonene i startfasen av utviklingen. På bakgrunn av denne informasjonen lagde vi en metode som hentet ut informasjonen automatisk fra telefonen og bestemte selv hvilken port den skulle bruke.

En viktig del i prototypen er evnen til å lagre enheter den har sendt til og enheter som avslø forespørselen. Tanken er som sagt at du skal bare få en melding pr tidsintervall og personer som ikke ønsker å bruke tjenesten, men allikevel ønsker ha på blåtann skal få ha muligheten til det, uten å bli forstyrret av vår prototype. Under brukertesting observerte vi at dette ikke var et stort problem, men allikevel ser vi på det som en viktig detalj.



Vi valgte også å legge inn en del feilhåndtering, siden dette skal være et helautomatisk system. Så hvis problemer oppstår under kjøring som skyldes kollisjoner med maskinvare som er opptatt eller lignende, vil bare programmet vente på tur enn å stoppe helt opp. Dette er for å sikre at tjenestekvalitet og oppetid på prototypen.

## 5.2 Testing av prototype

For oss var testingen av prototypen en stor del av selve utviklingsprosessen, både for å lære hvordan blåttann oppfører seg og for å se hvordan brukere oppfatter systemet. En av de første tingene vi oppdaget, var at få likte å få tilsendt meldinger på telefonene uten at de var klar over at det fantes en slik tjeneste i området. Det var derfor viktig at det fantes noe form for informasjon til brukeren slik at de som hadde blåttann påskrudd ble klar over at de mottok informasjon og ikke så på informasjonen som uønsket reklame. Videre oppfordrer dette også til å få brukere til å skru på blåttann, slik at de kunne benytte seg av tjenestetilbudet.

Videre fant vi fort ut at relevant informasjonen for person A er langt fra like relevant for person B, selv om de befinner seg på samme lokasjon. Det var derfor veldig viktig å ha en så informativ og så relevant melding som mulig for den bestemte lokasjonen uansett person.

## 5.3 Visuelt eksempel av prototype



Figuren hentet fra bluair [12]

Figuren viser hvordan prototypen skal fungere ved at det er en stasjonær basestasjon som sender ut meldinger, sanger eller filmsnutter.

## 5.4 utfordringer i utviklingen

Under utviklingen av prototypen var det flere tekniske og brukermessige utfordringer som vi måtte løse, og vi skal ta for oss et par av de tekniske.

En av de største tekniske utfordringene var eldre mobiler, der vi hadde problemer med å finne portinformasjonen. Dette kunne variere fra mobil til mobil, selv om modellen var den samme, noe som gjorde problemet enda vanskeligere. Løsningen ble rett og slett å lage en liten hack der vi valgte å bruke de fire mest vanlige portene for sending. Problemet med dette er at det tar lang tid å teste alle fire portene i forhold til å vite porten på forhånd. Så om man har veldig mange telefoner man ikke klarer å hente ut portinformasjonen på, kan systemet bli veldig tregt.

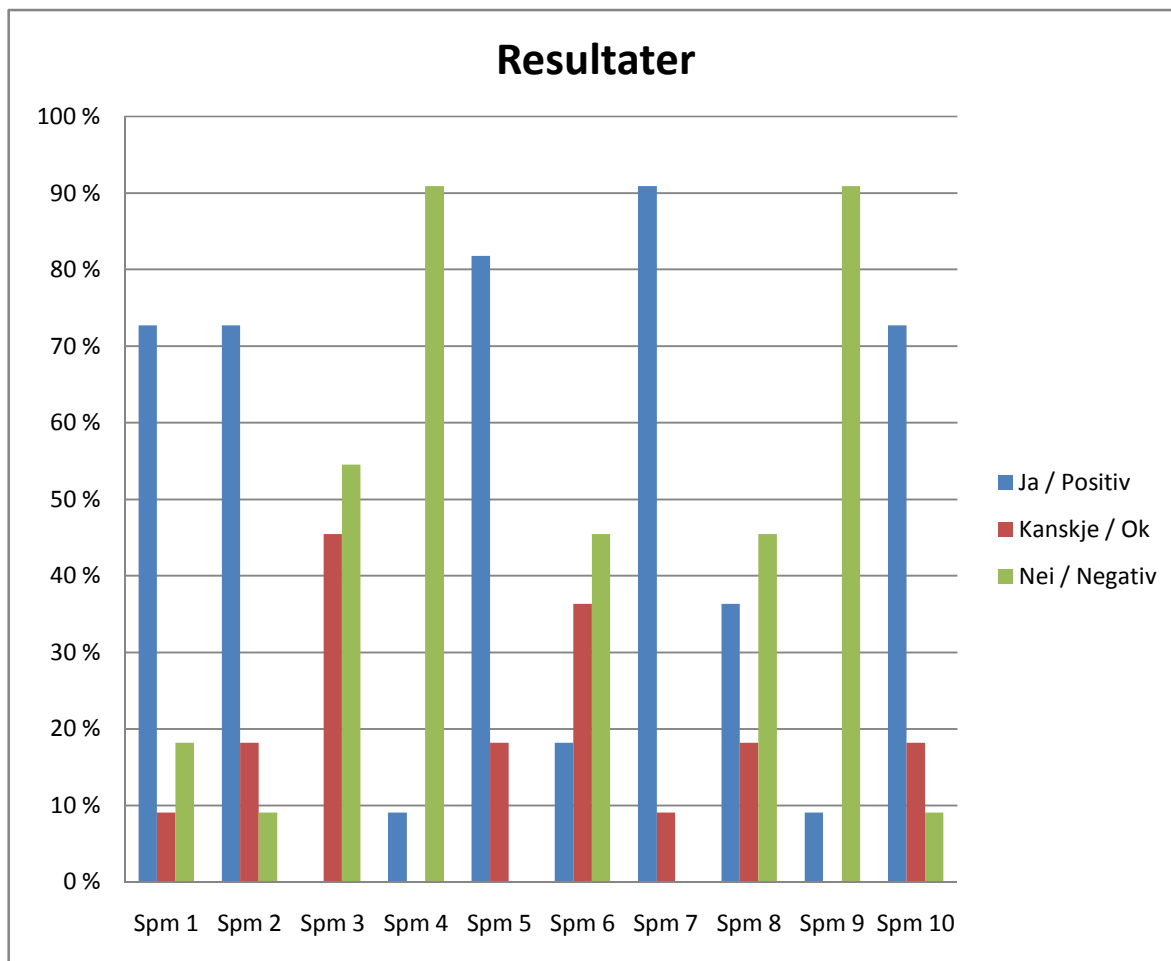
Et problem som vi ganske sent fant ut av, var at blåtann blir forstyrret av WLAN signaler. WLAN bruker nemlig det samme frekvensområde (2,4Ghz) som blåtann. Dette la vi ekstra godt merke til under fremføringen foran klassen, der vi hadde store problemer med å koble opp og hente ut portinformasjonen, noe som var igjen veldig uheldig for oss. Løsningen ble å anskaffe en klasse 1 blåtann enhet med rekkevidde på teoretiske 100m, men som fortsatt holder seg innenfor rommet, slik at den ikke sprer seg utenfor den aktuelle lokasjonen.

## 6.0 BRUKERUNDERSØKELSER

For å kunne besvarer en del av våre spørsmål i forhold til prototype og problemstilling, valgte vi å ha en spørreundersøkelse, der vi spurte tilfeldige forbigående på Institutt for informatikk. Personene vi spurte hadde en aldersspredning fra 21 til 63år, der flertallet var menn og hadde tilknytning til UIO. Fremgangsmåten med undersøkelsen var at vi satte opp en plakat der det sto ”skru på blåtann”. Det første som slo oss, var at plakaten i seg selv ikke var nok, vi måtte aktivt stoppe mennesker for å få dem til å aktivere blåtann.

Etter at prototypen hadde sendt dem en vennlig melding, stilte vi dem følgende spørsmål:

1. Hva er ditt førsteinntrykk av tjenesten?
2. Vil du være interessert i informasjon fra kantiner på UIO på f. eks dagens middag eller andre tilbud?
3. Vil du være interessert i tilbud fra butikker/forhandlere?
4. Har du vanligvis blåtann på- eller avslått?  
Hvorfor har du den på/av?
5. Vil du slå på blåtann for å kunne motta relevant informasjon for deg?  
(F. eks i en butikk, kino, musikkbutikk)
6. Synes du dette er på grensen til SPAM/reklame?
7. Kunne du tenke deg å bruke blåtann som et adgangskort?
8. Ser du på blåtann som sikkert med tanke på hacking?
9. Har du noen gang opplevd at noen har misbrukt blåtann?
10. Kunne du tenke deg å registrere deg for å motta informasjon skreddersydd til deg?



Det er et par ting som er viktig å legge merke til, det første var at over 70% var positive til tjenesten vi leverte. Videre var nesten alle mindre positive til å motta meldinger fra forhandlere og butikker, noe som ikke kom som en veldig stor overraskelse.

Men et problem for oss, er at ca 90% av de spurte sa at de ikke hadde blåttann aktivert og de fleste oppgav strømsparing som hovedargument.

Men selv om folk hadde blåttann deaktivert, var de fleste villige til å aktivere blåttann for å kunne motta relevant informasjon for sin lokasjon, hvis det var informert om det. Det positive var at over 90% ikke hadde opplevd misbruk av blåttann og over 70% var villig til å motta skreddersydd informasjon, selv om dette også var reklame. Vi fikk også bekreftet at 90% syntes at meldingen var positiv eller ok, og at den ikke ble direkte oppfattet som spam.

## 7.0 OPPSUMMERING

I denne oppgaven har vi sett på nærhetsnettverk og spesielt blåttann som en aktuell teknologi for å utvikle et meldingssystem basert på brukerens lokasjon. Vi har sett litt på eksisterende prosjekter og utviklet en prototype.

Målene våre med prototypen og prosjektet mener vi er oppnådd. Vi har klart å automatisk sende relevant informasjon til brukere ved å benytte en alment kjent teknologi, som etter vår mening representerte den mest hensiktsmessige totalløsningen både når det gjaldt utbredelse og bruksmuligheter. Blåttann gjorde det mulig for oss å lage en fungerende prototype som illustrerer prinsippene med en lokasjonsbasert meldingstjeneste meget godt.

Vi har også diskutert ulike metoder for å tilpasse informasjon til hver enkelt bruker av tjenesten. I et kommersielt system kunne dette vært interessant og utforske videre, spesielt brukernes villighet til å la seg registrere eller til å dele personlige preferanser. Vår spørreundersøkelse ga i denne sammenheng en interessant indikasjon.

## 8.0 REFERANSER

- [1] - Aagre, P: *Changing place: Contexts of awareness in computing*, 2001. Interactions.
- [2] - Jung, H., Stolterman, E., Ryan, W., Thompson, T., Siegel, M: *Toward a framework for ecologies of artifacts: how are digital artifacts interconnected within a personal life?* , ACM.
- [3] - Schilit, B., Hilbert D., Trevor, J.: *Context-Aware Communication*, 2002. IEEE Wireless Communications.
- [4] - Müller, J., Jentsch, M., Kray, C., Krüger, A.: *Exploring factors that influence the combined use of mobile devices and public displays for pedestrian navigation* , 2008. ACM.
- [5] - Rhodes BJ, Minar N and Weaver J: *Wearable Computing Meets Ubiquitous Computing: reaping the best of both worlds*, 1999. Symposium on wearable computing.
- [6] [http://en.wikipedia.org/wiki/Personal\\_Area\\_Network](http://en.wikipedia.org/wiki/Personal_Area_Network)  
besøkt: 2009-30-4
- [7] [http://en.wikipedia.org/wiki/IEEE\\_802.15](http://en.wikipedia.org/wiki/IEEE_802.15)  
besøkt: 2009-30-4
- [8] [http://en.wikipedia.org/wiki/Infrared\\_Data\\_Association](http://en.wikipedia.org/wiki/Infrared_Data_Association)  
besøkt: 2009-30-4
- [9] <http://en.wikipedia.org/wiki/Ultra-wideband>  
besøkt: 2009-30-4
- [10] <http://en.wikipedia.org/wiki/ZigBee>  
besøkt: 2009-30-4
- [11] <http://en.wikipedia.org/wiki/Bluetooth>  
besøkt: 2009-30-4
- [12] <http://www.bluair.pl/>  
besøkt: 2009-05-05
- [13] <http://www.computerworld.com/printthis/2004/0,4814,90131,00.html>  
besøkt: 2009-05-05
- [14] <http://student.grm.hia.no/master/ikt00/ikt6400/shartvig/Rapport.htm>  
besøkt: 2009-30-4
- [15] [http://gsyc.es/~anto/ubicuos2/bluetooth\\_security\\_and\\_hacks.pdf](http://gsyc.es/~anto/ubicuos2/bluetooth_security_and_hacks.pdf)  
besøkt: 2009-05-05
- [16] - <http://www.digi.no/811406/bluetooth-blir-mye-raskere>  
besøkt: 2009-23-4

[17] <http://www.nokia.no/finn-produkter>  
besøkt: 2009-28-04

[18] <http://www.sonyericsson.com/cws/products/mobilephones>  
besøkt: 2009-30-4

[19] <http://www.samsung.com/no/consumer/subtype/subtype.do?group=mobilephonemp3&type=mobilephonemp3&subtype=mobilephones>  
besøkt: 2009-30-4

[20] <http://www.bluez.org/about/>  
besøkt: 2009-28-04

[21] <http://org.csail.mit.edu/pybluez/index.html>  
besøkt: 2009-28-04

[22] <http://xmailserver.org/ussp-push.html>  
besøkt: 2009-28-04

## 9.0 APPENDIX - PROTOTYPENS KILDEKODE

```
#!/usr/bin/env python
import bluetooth, sys, os, time, datetime
#Usage ./BMS <File> <"1" for DEBUGMODE, "2" for scanning mode, <none> for normalmode>

# First we try to scann for bluetooth devices
# Then we either go to sendFile or to deBuggServices if we get "1"
# as argument nr 4.
# If we get anoter 4. argument we just print out
# the nearby_devices that we found, and then we sleep
def scannBlueTooth():
    #If the HW is bussy, we cant use bluetooth
    try:
        nearby_devices = bluetooth.discover_devices(lookup_names = True, flush_cache = True, duration = 3)
    except Exception:
        print("Bluetooth bussy, sleeping");
        goSleep(6);
        return;

    if len(nearby_devices)!=0:
        print nearby_devices

        # "1" is for debugning
        if len(sys.argv)<3:
            sendFile(nearby_devices);

        elif sys.argv[2]== "1":
            deBugServices(nearby_devices);
            #When done, we sleep for 1sek
            goSleep(1);

    #If we didn't find a bluetooth device we print out "No BlueTooth"
    #else:
        #print "No BlueTooth"
        #goSleep(10);

    #When done, go and sleep

def findObexPush(bdevice):
    port =-6;
    smatches = bluetooth.find_service(address = bdevice[0]);
    for line in smatches:
        if line["name"] == "OBEX Object Push":
            port = line["port"]

    return port;
```



```

def sendFile(nearby_devices):
    for bdevice in nearby_devices:
        if checkBUUser(bdevice[0])==0:

            foundPort=0;
            #Trying to find the port, retry 3 times
            while foundPort<3 :
                port = findObexPush(bdevice)

                #If we found the port then we move on
                byttPort=0;

                #If we did not find the port, try one more time, max3 times
                if port <0:
                    foundPort +=1;

                    #If we after 3 times did not found the port
                    #Then we have to guess the port
                    if foundPort == 3:
                        byttPort=1;
                        port = port * (-1)

                else:
                    foundPort =3;#Hvis vi finner porten saa
                    #settes variabelen til 3 slik at l0kka stopper

            #Then we try to send the message 4 times
            tryBlue=0;
            while tryBlue <4:
                if byttPort == 0:
                    goSleep(1);

                #Remove Whitespace fom Bluetoothname
                bName = bdevice[1].replace(" ", '').strip();

                #Make the command with: BLUETOOTH ADDRESS, PORT, FILENAME
                cmd = "%s@%d %s %s"%(bdevice[0],port, sys.argv[1], sys.argv[1])
                print cmd #DEBUGG

                #Runs the command
                r,w,e = os.popen3("nice -18 ussp-push %s"%(cmd));

                #If ther is any errors, the length of error > 0
                error = e.readlines()
                r.close;w.close;e.close;

            if len(error)!=0:

                tryBlue += 1;

                if port == 6 and byttPort == 1:
                    port=9;

                elif port == 9 and byttPort == 1:
                    port=5;

                elif port == 5 and byttPort == 1:
                    port=11;
            else:
                tryBlue = 4

        if tryBlue == 4:
            addBUUser(bdevice[0],bName)

```

```

#Adds a device/user to the list
def addBUser(mac,name):
    t = datetime.datetime.now()
    timestamp = klokka = t.strftime("%Y%j%H%M");
    b = open("BUsers.txt",'a');
    b.write("%s_%s_%s\n"%(mac,timestamp,name))
    b.close();

#Se if a device/user is in the list
def checkBUser(mac):
    #Henter timestamp
    t = datetime.datetime.now()
    timestamp = klokka = t.strftime("%Y%j%H%M");

    #Leser fila
    if not os.path.isfile("BUsers.txt"):
        return 0;
    b = open("BUsers.txt",'r')
    oldFile = b.readlines()
    b.close

    newFile = "";
    newUser = 0;update = 0;

    for line in oldFile:
        if line.startswith(mac):
            newUser =1;
            update =1;
            if int(line[18:29])+100 < int(timestamp):
                newFile = "%s%s"%(newFile,"")
                update = 0;
            else:
                newFile = "%s%s"%(newFile,line)
        else:
            newFile = "%s%s"%(newFile,line)

    if newUser==1:
        c = open("BUsers.txt",'w')
        c.write(newFile);c.close()
    return update;

```

```

#Just a wait command
def goSleep(tid=3):
    time.sleep(tid)

#This is a debugmode to se what services a phone got
def debugServices(nearby_devices):

    device=0;service_matches =[]

    for bdevice in nearby_devices:
        smatches = (bluetooth.find_service(address = nearby_devices[device][0]))

        for i in smatches:
            service_matches.append(i)
            device +=1

    if len(service_matches) == 0:
        print "couldn't find the FooBar service"

    for services in service_matches:
        print " Name: %s" % (services["name"])
        print " Host: %s" % (services["host"])
        print " Description: %s" % (services["description"])
        print " Protocol: %s" % (services["protocol"])
        print " Provider: %s" % (services["provider"])
        print " Port: %s" % (services["port"])
        print " Service id: %s" % (services["service-id"])
    print ""

#This program runs forever
while(1):
    scanBlueTooth()

```