



Handelshögskolan

VID GÖTEBORGS UNIVERSITET

Institutionen för informatik

31/3 20

UTMANINGAR FÖR KONTEXTMEDVETNA APPLIKATIONER

En studie av moderna bilar som användningsmiljö för kontextmedvetna applikationer

Abstrakt

Den tekniska utvecklingen rör sig mot mer mobila och intelligenta it-lösningar. Datorerna börjar ta steget ut från våra hem och skrivbord och in i vår mobila miljö. Den mobila miljön är mer föränderlig och oförutsägbar än den statiska miljön. Detta skapar ett ökat behov av applikationer som är medvetna om och kan reagera på omgivningen. Denna uppsats tittar närmare på hur moderna bilar lämpar sig som miljö för kontextmedvetna applikationer. Som metod användes en prototyp till ett kontextmedvetet spel som använder sig av omgivningsdata som samlas in av en SAAB 93:s sensorer. Uppsatsen försöker visa att genom att använda moderna bilar som miljö för kontextmedvetna applikationer så underlättas eller löses många av de problem och utmaningar med kontextmedvetna applikationer som beskrivs i litteraturen.

Nyckelord: Context-aware, ubiquitous, mobile computing, task-artifact cycle

Författare: Mark Lagerström

Handledare: Rikard Lindgren

Examensarbete I, 10 poäng

1	Inledning	3
1.1	Frågeställning.....	4
1.2	Disposition	5
2	Utmaningar för kontextmedvetenhet	6
2.1	Begreppet kontext	6
2.2	Kontextmedvetenhet är baserad på mer än positionsdata.....	7
2.3	Användbart gränssnitt	8
2.4	Behovet av en infrastruktur.....	9
2.5	Design frågor	9
2.6	Sociala aspekter	10
2.7	Utmaningar för ett vetenskapligt tillvägagångssätt inom ett visionärt område	10
2.8	Behovet av utvecklingsverktyg.....	11
2.9	Sensorer.....	11
2.9.1	Sensorer i bilar	12
3	Behov för förenklad kontexthantering	14
3.1	Separering av problem	14
3.2	Kontext tolkning	14
3.3	Genomskinlig, distribuerad kommunikation	15
3.4	Konstant tillgång till kontextdata.....	15
3.5	Kontext lagring och historia.....	15
3.6	Resurs upptäckande	16
3.7	Problem med kontext hantering.....	16
4	Metod	17
4.1	Uppgift-artifakt cykeln.....	17
4.2	Prototyp som metod	18
4.2.1	Low-fidelity prototype	18
4.2.2	High-fidelity prototyp	19
4.3	Metod val	19
4.4	Bakgrund till metoden.....	20
5	Resultat	21
5.1	Utvecklingsarbetet	21
5.2	CABdriver.....	22
5.2.1	Teknisk beskrivning.....	22
5.2.2	CABdriver i korthet	22
5.2.3	Framtida utvecklingsarbete på CABdriver	24
5.3	Användarscenario	24
5.4	Metodkritik	25
6	Diskussion.....	26
6.1	Vad är CABdriver?	26
6.2	Utmaningar för kontextmedvetenhet	26
	Utmaningar för ett vetenskapligt tillvägagångssätt inom ett visionärt område	29
6.3	Behov för förenklad kontexthantering.....	29
7	Slutsats	32
8	Referenser	33
9	Appendix.....	35

9.1 Projektdeltagere	35
9.2 Sensorer i SAAB 93.....	35

1 Inledning

För 40 år sedan var det otänkbart med en dator i varje hem. Det är idag en realitet. Nu börjar datorerna att ta steget ut från våra hem och andra fasta platser som kontoret. Mobiltelefonerna får fler funktioner än att bara ringa. Mobiltelefoner med kameror blir vanligare och funktioner som gör att användaren kan komma åt Internet finns på ett flertal mobiler. Handdatorer och laptops blir vanligare och kan utrustas med en rad olika tillbehör som trådlösa nätverk (WLAN) eller mobila modem och så vidare. Bilar börjar utrustas med olika informations- och telematik system. Automatiska kartläsare börjar bli vanliga i nya bilmodeller men också rena nöjesfunktioner börjar dyka upp, den nya generationens kopphållare, exempelvis kan Opel Signum köpas med en DVD-spelare i baksätet och köparen får dessutom en PlayStation 2 på köpet så att denne kan spela. Toyotas Will Cypha levereras med en karaoke anläggning dock endast på den Japanska marknaden. Ett område som blir intressant när datorerna blir portabla och finns med användaren hela tiden är kontextmedvetenhet. Den typiska användaren sitter inte längre med en stationär dator i den ganska förutsägbara kontorsmiljön. Istället använder denne sig av en rad olika föremål, både fasta och mobila skriver Dey et al (2001). Det verkar som om vi rör oss mot realiseringen av Weisers (1991) "ubiquitous computing paradigm" eller tredje vågens datorisering som är uppfyllt när antalet specialiserade datoriserade föremål överskrider antalet användare. Det kan tyckas att så redan är fallet om man tittar på kassaapparater, bankomater, parkeringsmätare, mobiltelefoner, laptops, handdatorer, TV, video/DVD, stereo och så vidare. Dock är inte fallet så, det saknas ännu många viktiga bitar för att förverkliga Weisers (1991) vision. Den största bristen, anser Dey et al (2001), är att interaktionen mellan föremål och användare fortfarande liknar den mellan användaren och en stationär dator. Föremålen, främst de mobila men även de stationära, används alltmer i en föränderlig omgivning men de anpassar sig inte till den. Finns det mobiltelefoner som själva anpassar ljudvolymen till vad som är lämpligt eller handdatorer som anpassar skärmens ljusstyrka efter hur ljust det är ute? De datoriserade föremålen är i stor utsträckning fortfarande omedvetna om sin omgivning. En hypotes som många forskare inom ubiquitous computing området delar är att om föremålen automatiskt kan anpassa sig till omgivningen så kommer användarvärdet att öka. En del av området ubiquitous computing behandlar kontextmedvetenhet och en ökande del av forskningen inom området görs om kontextmedvetna applikationer och föremål. Det är denna del som uppsatsen kommer att fokusera på. Dey et al (2001) definierar kontextmedvetenhet som information som karakteriserar en situation relaterad till interaktionen mellan användare, applikationer och omgivningen. En kontextmedveten applikation kan till exempel vara ett navigationssystem.

Dey et al (2001) och Olsson (2003) har identifierat flera olika utmaningar som försvårar skapandet av kontextmedvetna applikationer, bland annat de problem som kan uppstå med tillgången till sensorer. För att en applikation skall vara kontextmedveten krävs det att den är medveten om omvärlden på något sätt. Detta sker nästan alltid genom att applikationen är ansluten till en eller flera sensorer. I Moderna bilar finns det en mängd

olika sensorer vilket kan göra dem intressanta som ”testbänkar” för kontextmedvetna applikationer.

Det är viktigt att testa kontextmedvetna applikationer i en naturlig miljö för att få bra resultat skriver Grudin (2001). Detta gör det intressant att använda till exempel bilar som en testmiljö för kontextmedvetna applikationer eftersom de är ett vanligt och naturligt inslag i vår vardag.

Bakgrunden till denna uppsats är ett projekt som jag under våren 2003 kom i kontakt med. Projektet är ett samarbete mellan Viktoriainstitutet, SAAB automobil, Mecel och Vodafone. Detta Storprojekts syfte är att ”...identifiera och analysera hur innovativa telematiktjänster för bilen kan integreras och komma till nytta i olika användningssammanhang.” Mitt delprojekt i detta storprojekt var att, tillsammans med en kollega, skapa ett kontextmedvetet spel för baksätetspassagerare. Detta resulterade i spelet CABdriver (Context Aware Backseat driver). CABdriver spelas på en handdator i en SAAB 93 och kontextinformation från bilen används hela tiden för att påverka spelet.

1.1 Frågeställning

En naturlig infallsvinkel vid utvecklingen av nya informationslösningar är att studera vilka miljöer som kan ha behov av detta. Vilka miljöer som kan förbättras eller förenklas av en it-lösning, exempelvis kan en revisorfirma förbättras av ett bokföringsprogram eller en skola förbättras av ett textbehandlingsprogram. Detta är en bra infallsvinkel eftersom det är meningslöst att utveckla produkter för miljöer som inte behöver dem.

När det gäller kontextmedvetna applikationer finns det en mängd olika sätt de kan förbättra miljön, till exempel navigationssystem för bilar. Problemet med kontextmedvetna applikationer är att de, till skillnad från ”vanliga applikationer”, inte bara behöver en dator och en nätverksanslutning utan också en eller oftast flera olika sensorer för att kunna få information om kontexten. Dessa sensorer ökar kostnaden på produkten avsevärt. Dessutom kan de i vissa fall behöva installeras av kompetent personal vilket ytterligare ökar kostnaden. Ur ett forskningsperspektiv kan det vara svårt att testa kontextmedvetna applikationer i en naturlig miljö eftersom denna behöver vara utrustad med sensorer. Detta leder till att prototyperna ofta testas i en labbmiljö. Grudin (2001) skriver om behovet att testa kontextmedvetna applikationer i en naturlig miljö och problemen med att testa prototyper i en konstgjord labbmiljö: ”The price paid for relying on simple prototypes in research settings is well known: This is not user-centered design...Until real applications are tried, we do not know what is in making them work, or whether or not they are workable”.

Detta skapar förutsättningar att titta på vilka miljöer som lämpar sig för kontextmedvetna applikationer istället för att titta på vilka miljöer som har behov av kontextmedvetna lösningar. Detta kan leda till att det är möjligt att skapa kontextmedvetna applikationer till en bråkdel av kostnaden. Även om kostnaden inte är lika viktig ur ett forskningsperspektiv kan det vara av intresse att veta vilka miljöer som finns lätt tillgängliga att genomföra naturliga tester i. Om undersökningar kan genomföras i sin naturliga miljö underlättar detta för att få ett bra resultat. Till exempel är det en förutsättning för etnografi att observera fenomen i dess naturliga miljö (Hughes et al 1994). Det skulle gagna både forskning och näringsliv om det går att identifiera miljöer som är mer lämpade för kontextmedvetna applikationer än andra.

Om sådana miljöer finns kommer nästa stora utmaning nämligen att skapa meningsfulla applikationer för den. Det är alltså nödvändigt att hitta en miljö som inte bara är lätt att designa kontextmedvetna applikationer för utan som även har *nytta* av sådana. Även om det inte går att skapa meningsfulla applikationer kan en sådan miljö vara intressant ur forskningsperspektiv eftersom den kanske kan användas för att enkelt och snabbt testa kontextmedvetna applikationer.

Redan vid projektets start stod det klart att moderna bilar kanske kunde vara just en sådan miljö eftersom de är en vardagsmiljö som är rik på sensorer. Det finns än så länge inte så mycket skrivet om moderna bilars lämplighet för kontextmedvetna applikationer. Jag valde därför att inrikta studien och min uppsats på att undersöka bilen som miljö för kontextmedvetna applikationer eftersom den potentiellt kan lösa många av de utmaningar för kontextmedvetenhet som Olsson (2003) och Dey et al (2001) skrivit om. Min frågeställning lyder:

Hur lämpar sig moderna bilar som användningsmiljö för kontextmedvetna applikationer?

Med denna frågeställning som utgångspunkt så kommer uppsatsen att fokusera på de mera tekniska aspekterna av användningsmiljön, främst tillgången till sensorer och kommunikationsmöjligheter.

När jag skriver ”användningsmiljö för kontextmedvetna applikationer” så menar jag miljön som applikationen skall användas i. Jag skriver alltså inte om hur en person, till exempel Kalle, använder kontextmedvetna applikationer i en viss miljö. Jag menar snarare hur applikationen *i sig självt* använder miljön till exempel: Windows användningsmiljö är en PC, en cykels användningsmiljö är en cykelbana och så vidare. Den mänskliga användaren är inte fokus för denna uppsats utan det är applikationen som användare som uppsatsen huvudsakligen kommer att handla om.

1.2 Disposition

Uppsatsen är disponerad enligt följande: I kapitel 2 redogör jag generellt för vilka utmaningar som kontextmedvetenhet står inför och i kapitel 3 tittar jag specifikt närmare på vilka behov som finns för att förenkla kontexthanteringen i kontextmedvetna applikationer. I kapitel 4 redogör jag för det teoretiska ramverket runt den metod, skapandet av en prototyp, som jag har valt och i kapitel 5 redovisas resultatet av metoden, ett kontextmedvetet spel vid namn CABdriver (Context Aware Backseat driver). I kapitel 6 diskuterar jag resultatet av metoden och huruvida CABdriver har lyckats besvara de problem som ställs upp i kapitel 2 och 3. I kapitel 7 drar jag slutsatser av diskussionen och resultatet.

2 Utmaningar för kontextmedvetenhet

Olsson (2003) har identifierat 8 stycken utmaningar inom kontextmedvetenhetsområdet. Nämligen:

1. Begreppet kontext
2. Kontextmedvetenhet är baserad på mer än positionsdata
3. Användbart gränssnitt
4. Behovet av en infrastruktur
5. Design frågor
6. Sociala aspekter
7. Utmaningar för ett vetenskapligt tillvägagångssätt inom ett visionärt område
8. Behovet av utvecklingsverktyg

Dessa punkter behandlas i tur och ordning i det följande kapitlet. Dessutom utvecklas punkt 8 ordentligt i kapitel 3.

Sist i kapitlet följer ett avsnitt om sensorer eftersom de är väsentliga för att göra en applikation medveten om omvärlden, det vill säga kontextmedveten.

2.1 Begreppet kontext

Begreppet kontext har debatterats mycket och kan för olika forskare ha olika mening, därför är det viktigt att på ett tidigt stadium reda ut hur jag ser på kontext. Svenska akademins ordlista (SAOL) Definierar kontext som ”text i sitt sammanhang (ofta i motsättning till: not l. anmärkning); (ett skriftställes) sammanhang med det föregående o. efterföljande; (en skrifts) innehåll; vanl. i best. form.” Denna definition är för generell för att kunna användas i kontextmedvetenhets sammanhang. Olika definitioner och argument diskuteras i Human-Computer interaktion journal, volume 16. Där presenterar Dey et al (2001) en artikel som definierar kontext som ”information som kan användas för att karakterisera situationen av entiteter som anses relevanta för kommunikationen mellan användare och applikation, inklusive användaren och applikationen själva”. Benerecetti et al (2001) omdefinierar kontext för att bättre reflektera distribuerade kontextmedvetna system: ”Kontext är inte bara en samling av egenskaper hos omgivningen, utan en partiell och approximativ representation som används av en agent för att interagera med omgivningen och andra agenter.” Dorish (2001) understryker i sin tur att de sociala aspekterna av kontext inte fångas av Dey et al (2001) och hävdar istället att ett socialvetenskaplig angreppssätt såväl som ett tekniskt från Weiser (1991) och Ishii & Ullmer (1997) båda kommer från ett ömsesidigt beroende av att förkroppsliga konceptet. De olika författarnas olika definitioner kan vara både svårtolkade och otydliga därför har jag valt att göra en egen definition av kontextmedvetna applikationer som gäller denna uppsats. *En kontextmedveten applikation är en applikation som är medveten om delar av omgivningen och kan reagera på detta.*

Det är omdebatterat när den första kontextmedvetna applikationen kom till. Det kan sägas att ett Air Condition system är en kontextmedveten applikation. Det känner av temperaturen i omgivningen och kan anpassa sig därefter. Ett exempel på applikationer som interagerar mer med kontexten är Cafe Trekk (2001); Ett spel för handdatorer som ger användaren tillgång till olika funktioner beroende på var denne befinner sig. Ett annat exempel är ett spel av Brunnberg & Juhlin (2003) som låter användaren använda sin handdator som ett virtuellt fönster mot verkligheten där position och plats är av stor betydelse. När spelarens position överensstämmer med rätt plats uppfylls vissa villkor. Det är svårt att säga när en applikation blir kontextmedveten, det går att argumentera för att vanligt tangentbordsinput är kontextpåverkan som applikationen reagerar på.

2.2 Kontextmedvetenhet är baserad på mer än positionsdata

Många kontextmedvetna applikationer använder sig av positionsdata för att skapa kontextmedvetenhet men det finns många andra faktorer i omgivningen som kan vara av intresse för användaren. Schmidt et al (1998) skriver att medvetenhet om omgivningen är det viktigaste för kontextmedvetna mobila föremål. Den vanligaste typen av kontextmedvetenhet för dessa föremål är positionen, till exempel genom GPS eller mobilpejling. Positionen är bara en av flera aspekter hos omgivningen skriver Schmidt et al som argumenterar för att ju mer en kontextmedveten applikation vet om sin omgivning desto bättre.

Schmidt et al (1998) drar från tidigare studier av mobila kontextmedvetna applikationer slutsatsen att:

- Användbarheten hos kontextmedvetna applikationer är uppenbar men att det är svårt att jämföra olika applikationer på grund av en generell bild av vad kontext är saknas
- Användandet av position är den dominerande funktionen och används ofta i samband med andra funktioner. Till exempel ett navigations system som kombinerar kartor med användarens position.
- Sensorer kan användas för att ta kontextmedvetenheten bortom position.

Schmidt et al skriver att mobila kontextmedvetna applikationer kan använda kontexten på 2 huvudsakliga sätt. Ett, applikationen anpassar sig till omgivningen, till exempel genom att automatiskt välja det trådlösa nät som har bäst signalstyrka. Två, förbättra användargränssnittet, till exempel genom att öka skärmens ljusstyrka om det är ljus ute eller rotera texten på skärmen så att den alltid är vänd mot användaren oavsett vilket håll denne ser skärmen. Stationära datorer befinner sig i en omgivning som är ganska förutsägbar medan mobila föremål ofta befinner sig i en föränderlig omgivning, därför finns det behov av att de mobila föremålen kan anpassa sig till sin omgivning. Schmidt et al drar slutsatserna att: kontextmedvetenhet kan användas för att öka användbarheten hos mobila föremål, främst genom att förbättra interaktionen mellan föremål och användare och att genom att använda en mängd olika sensorer blir det troligare att användarnyttan kan ökas mer än med bara kunskap om användarens position.

2.3 Användbart gränssnitt

Ett användbart gränssnitt är viktigt så att användaren lätt kan kommunicera med applikationen och när det gäller kontextmedvetna applikationer öppnas det många nya möjligheter för denna kommunikation. Siewiorek (2002) undersökte wearabel computers, alltså datorer som användaren har *på sig*, som ett sätt att skapa gränssnitt för användaren och kom fram till att "...biggest challenge merging ubiquitous and wearable computing deals with fitting the computer to the human in terms of interface, cognitive model, contextual awareness, and adaptation to tasks being performed." På samma sätt resonerar Lyytinen & Yoo (2002) när de diskuterar pervasive computing, de skriver att datorn har kapacitet att hämta och använda information från omgivningen och att omgivningen, i sin tur, bör vara intelligent och kunna upptäcka när datorer kommer in i den. Med detta menas att omgivning och applikationer bör kunna interagera på ett meningsfullt sätt. Brunnberg och Juhlin (2003) har skapat "The backseat gaming prototype" som är ett mobilt kontextmedvetet spel för handdatorer. Spelet använder sig av ett innovativt gränssnitt där handdatorns position, riktning och lutning påverkar spelet. Spelet använder sig av den föränderliga omgivningen och känslan av rörelse runt bilen för att skapa ett underhållande spel. Brunnberg och Juhlin (2003) skriver att mobila spel ofta bara är portabla varianter av klassiska datorspel men att det finns möjlighet att också inkorporera olika aspekter av mobilitet i designen för att skapa mer involverande/engagerande mobila spel. De föreslår att mobila spel kan bli mer attraktiva om de är medvetna om den färgstarka och dynamiska kontexten i mobila sammanhang till exempel när användaren färdas i en bil. The backseat gaming prototype spelas på en handdator som med hjälp av sensorer känner till vertikal och horisontell vinkel och GPS koordinaten. Vid specifika platser får spelaren instruktioner om att leta efter ett specifikt objekt längs vägkanten, till exempel ett hus, och att rikta handdatorn mot detta så att virtuella objekt kan visas på handdatorns skärm, skärmen blir på detta vis ett fönster mot verkligheten med ett naturligt och enkelt gränssnitt. Målgruppen för spelet är barn som åker i baksätet. Dessa är nästan i samma upplevelsesituation som föraren men de kan ägna sig åt andra aktiviteter än att köra bilen. De kan njuta av färden och titta ut genom bilens fönster, ofta är de också engagerade i andra aktiviteter samtidigt. De läser, pratar eller spelar traditionella kontext relaterade spel som till exempel att räkna bilar. Sedan 1980talet är också mobila spel en möjlighet. En rapport från Andersen Consulting 2002 visar att mobila spel huvudsakligen spelas under resor och att bilen är den populäraste platsen för detta. Med backseat gaming adderar Brunnberg och Juhlin ett nytt spelkoncept för att göra situationen mer njutbar för spelaren.

I spelet är specifika objekt vid vägkanten av största betydelse eftersom det är de som initierar de



Figur 1. Virtuell verklighet.

olika momenten i spelet och står för kopplingen mellan verklighet och spel. Dessa platser är hårdkodade i spelet vilket gör att spelet endast är meningsfullt att spela på vissa redan förberedda sträckor. Brunnberg och Juhlin konstaterar att objekten måste ligga inom ett kortare avstånd från vägkanten och vara lätta för spelaren att peka ut, det kan vara objekt som hus, träd eller ett område med kolonilotter. Objekten måste också vara lämpliga att förmedla en specifik mening som spelaren lätt kan förstå. Brunnberg och Juhlin använder bland annat ett gammalt träd för att starta en händelse med spöken i spelet. Se Figur 1. Virtuell verklighet.

Brunnberg och Juhlin testade spelet på 3 barn och konstaterade att de hade svårt att förstå spelet i början men att de efter ett tag uppskattade spelet. Brunnbergs och Juhlins slutsats är att det är möjligt att använda det föränderliga scenariot utanför en bil för att skapa en rolig och underhållande mobil upplevelse. De konstaterar också att förhållandet mellan verkliga objekt och virtuella i spelet är viktigt.

The backseat gaming prototype visar att det är möjligt att framgångsrikt använda andra typer av gränssnitt än de traditionella tangentbordet och musen.

2.4 Behovet av en infrastruktur

Det finns ett behov av en infrastruktur för mjukvara, hårdvara och nätverks-kommunikation. Det finns ett antal kontextmedvetna applikationer som behöver kunna kommunicera med omvärlden på andra sätt än att bara använda sina egna sensorer. Dessa applikationer behöver en infrastruktur som ger dem möjlighet att hitta, anpassa sig, och skicka rätt information till användaren beroende på användarens kontext. Banavar & Bernstein (2002) tar upp ett illustrerande exempel med en chef som reser mellan två av företagets kontor i olika städer. På olika platser under färden finns det olika möjligheter för chefens handdator att ansluta mellan omvärlden. I taxin på väg till flygplatsen finns det bara uppkoppling med begränsad bandbredd tillgänglig därför prioriteras endast viktiga kortare medelanden men när chefen kommer till flygplatsen ansluter handdatorn automatiskt till den ökade bandbredd som flygplatsen WLAN erbjuder. Banavar & Bernstein (2002) betonar betydelsen av att mobila applikationer skall kunna koppla upp sig mot omvärlden och att detta skall ske utan att användaren behöver göra något. Applikationerna skall själva kunna identifiera och kommunicera med olika anslutningstyper och input/output föremål. För att göra detta möjligt har Banavar & Bernstein (2002) identifierat fem olika områden som behöver extra uppmärksamhet:

1. Standardiserade dataformat och protokoll.
2. Skapandet av grundläggande infrastrukturjänster som till exempel funktioner för att upptäcka sensorer i omgivningen.
3. Definiera ansvar för föremål, applikationer och infrastruktur.
4. Kommunikationen mellan användare, applikationer och sensorer.
5. Möjlighet att skapa en omfattande infrastruktur som inte behöver en stor administration.

2.5 Design frågor

Fisher (2001) skriver om behovet av en delad förståelse mellan användare och datorsystem och påpekar att ett av de fundamentala problemen för mjukvarudesign är att

skapa mjukvara för miljontals användare samtidigt som den fungerar som om den var designad speciellt för varje individuell användare. För att lösa detta anser Fisher (2001) att utmaningen är att göra applikationerna kontextmedvetna så att de är medvetna om andra användares handlingar och kan använda detta för att skapa en bättre miljö för den individuella användaren. Applikationerna skall alltså kunna dra slutsatser inte bara av en användares handlingar utan av alla användares handlingar för att på detta sätt skapa ett ökat användarvärde. Detta kan man se exempel på i till exempel e-shops där den potentiella köparen av en produkt kan se vad andra som köpte denna produkt köpte för andra saker. Här drar dock applikationen inga slutsatser utan lämnar det åt användaren däremot använder den information från många olika användare för att öka användarnytan.

2.6 Sociala aspekter

Kontextmedvetna applikationer kan medföra en rad sociala aspekter. Till exempel applikationer som visar var användaren befinner sig, användaren kanske inte vill att hans position skall vara känd. Privatlivet kan påverkas mycket av kontextmedvetna applikationer eftersom de känner till mycket om omvärlden och användaren. Banavar & Bernstein (2002) analyserar sitt exempel (Se 2.4 Behovet av en infrastruktur) för att se vilken påverkan det kan ha på den sociala miljön och drar slutsatsen att all introduktion av ubiquitous computing och därför även kontextmedvetna applikationer innebär introduktionen av sensorer i miljön som oåterkalleligt har en effekt på den sociala miljön oavsett hur små och betydelselösa de är. Sedan kan man givetvis diskutera *hur* mycket de påverkar den sociala miljön. Banavar & Bernstein (2002) nöjer sig med att konstatera att sensorer påverkar - inte hur mycket.

2.7 Utmaningar för ett vetenskapligt tillvägagångssätt inom ett visionärt område

Sättet att forska om ett område som till stora delar består av visioner och framtida möjligheter är en utmaning. Lyytinen & Yoo (2002) noterar att en studie av ubiquitous computing (och därför också kontextmedvetenhet) kräver att minst tre aspekter hanteras:

1. Ubiquitous computing är i ett tidigt stadium av utveckling därför är det viktigt att forskarna håller fast vid vetenskapliga metoder utan att begränsa sin förmåga att vara innovativa.
2. Forskarna behöver finna ett sätt att studera personliga frågor på en global nivå. Till exempel hur Wearable computers interagerar med omgivningen och användaren och hur detta påverkar användarens tillfredsställelse och produktivitet. Detta måste studeras i ett kontext av global diffusion och mobil teknologi.
3. Forskning inom Ubiquitous computing kräver ett överskridande av de traditionella barriärerna mellan det sociala och det tekniska.

Banavar & Bernstein (2002) pekar på problemet med att validera användarerfarenheter eftersom Pervasive computing har möjlighet att fundamentalt ändra sättet människor använder datorer. Olsson (2003) efterlyser ett verktyg för att effektivt kunna testa och utvärdera scenarier som görs möjliga av Pervasive computing och kontextmedvetna applikationer.

2.8 Behovet av utvecklingsverktyg

Ett grundläggande problem med kontextmedvetna applikationer är att de kräver mycket tid och ansträngning för att skapa på grund av behovet av sensorer och problemen med kommunikation med dessa. Dey et al (2001) konstaterar att det ett mycket dåligt utbud av verktyg för att skapa kontextmedvetna applikationer och påpekar att detta hindrar designers i deras skapande. Alltför mycket tid måste läggas på tekniska aspekter som är mindre intressanta ur ett forskningsperspektiv. Därför har Dey et al (2001) bland annat identifierat 6 stycken behov för att designers lättare skall kunna hantera kontext. Dessa sex punkter utvecklas i kapitel 3.

2.9 Sensorer

För att ge applikationer en möjlighet att bli kontextmedvetna behövs det sensorer som applikationen kan kommunicera med. Det finns idag en otrolig mängd olika föremål som på olika sätt kan mäta omgivningen till exempel: GPS-mottagare, övervaknings kameror, termometrar, rörelsesensorer och så vidare. För att de på ett smidigt sätt skall kunna kommunicera med en applikation bör det vara försedda med någon form av informations överförings möjlighet till exempel: Bluetooth, LAN eller WLAN.

Vissa kontextmedvetna applikationer behöver bara en sensor till exempel ett gps-navigations system som bara behöver gps-data. Många kontextmedvetna applikationer behöver dock ha tillgång till så mycket kontextdata som möjligt. Ju mer desto bättre. Att skapa nätverk med mängder av sensorer i är

kostsamt. Dels måste sensorerna köpas in dels måste de ha möjlighet att kommunicera med applikationen. En lösning som man ser ofta är handdatorer med extra utrustning på till exempel Brunnberg och Juhlines (2003) handdator som känner av position med GPS och handdatorns egen lutning med en digital kompass. Se Figur 2. Brunnberg & Juhlines handdator.

Eftersom det kan vara så resurskrävande att skapa/skaffa sensorer till kontextmedvetna applikationer är det intressant att veta vilka miljöer som redan har ett utbrett nätverk av sensorer som en kontextmedveten applikation kan använda sig av.

Några exempel på miljöer som kan vara rika på sensorer:

- Varuhus har kameror, kassatorer, dörrsensorer vissa varuhus har WLAN i hela butiken till exempel COOP Bäckebo.



Figur 2. Brunnberg & Juhlines handdator

- Industrier har en mängd olika sensorer för att övervaka tillverkningen till exempel till industri robotar eller löpande band.
- Fordon, till exempel båtar, flygplan och bilar, har på senare år blivit väl utrustade med en rad olika sensorer.

Enligt Dey et al (2001) är det viktigt att sensorerna är sammankopplade och lätta att kommunicera med. Så kanske inte är fallet i alla sensor rika miljöer till exempel i industrier men i vissa fordon så är det i högsta grad fallet. Det leder oss in på nästa avsnitt.

2.9.1 Sensorer i bilar

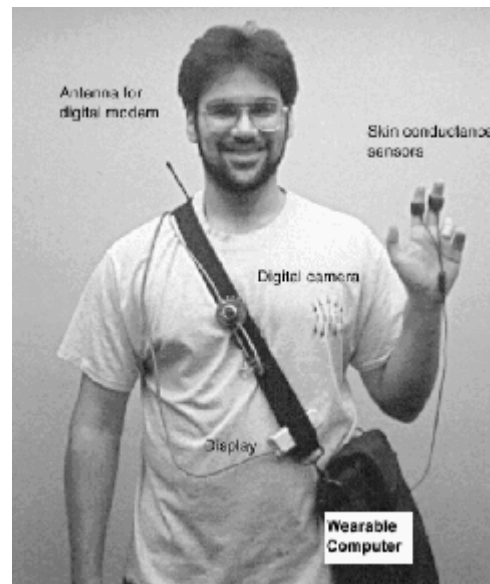
Fler och fler forskare börjar få upp ögonen för ett vardagsföremål som är mycket högteknologiskt och utrustat med en mängd olika sensorer nämligen bilen. Mycket av den forskning som görs om kontextmedvetenhet sker i labbmiljö. Se till exempel Healy & Picards (1998) kamera som antagligen inte känns helt normal och naturlig att använda, se Figur 3. ContextCamera.

I de moderna bilarna har forskaren eller designern tillgång till ett välutrustat nätverk av sensorer *och* en miljö som känns väldigt vardaglig för de flesta. Detta gör att eventuella systemtester kan göras under mer naturliga former än vad som är fallet för vissa andra kontextmedvetna applikationer. Bilen är ett vardagsföremål både för privatpersoner och företag och en viktig hörnsten i vårt samhälle.

Detta gör bilen intressant ur en mängd olika perspektiv bland annat det kontextmedvetna eftersom bilen har så många användare *och* en mängd sensorer. Då en mängd olika människor använder bilar blir målgruppen för kontextmedvetna applikationer för bilen stor.

Bilen har också intressant aspekter ur ett annat perspektiv. Banavar & Bernstein (2002) skriver ”Any introduction of a ubiquitous computing implies the introduction of sensors, which irrevocably have an impact on the social structure, no matter how unobtrusive they seem to be.”. I bilen finns sensorerna redan på plats vilket minskar påverkan på den sociala strukturen jämfört med till exempel en kamera.

Det finns också nackdelar med att använda bilen som testbänk. Det krävs att designern har tillgång till en ny (dyr) bil. Dessutom är det inte bra om vem som helst att ändra i bilens system eftersom det kan hantera aspekter som är kritiska för körningens säkerhet. Detta gör att det behövs ett nära



Figur 3. ContextCamera



Figur 4. SAAB 93

samarbete med biltillverkaren eller biltillverkarens samarbetspartners. Många nya bilar som tillverkas idag är väl utrustade med informations system, se Figur 5. SAAB 93 Instrumentpanel. Systemen känner till mycket om bilens status. Till exempel en SAAB 93s system kan känna mer än 20 olika variabler för bilen, se Figur 4. SAAB 93. Några som kan vara intressanta ur ett kontext perspektiv är:

- GPS-position
- Navigationssystem med kartor
- Bilens hastighet
- Bränsleförbrukning
- Bränslenivå
- Temperatur
- Driver workload (ett värde på hur pass upptagen föraren är av körningen)
- Trafikmeddelanden (TrafficMessageChannel)
- Varningslampor för fel på bilen



Figur 5. SAAB 93 Instrumentpanel

För en komplett lista av alla sensorer se Appendix 9.2.

SAAB 93 är den första bilen som levereras med BlueTooth vilket avsevärt underlättar kommunikationen med till exempel mobiltelefoner och handdatorer i bilen. Det är nödvändigt att det finns någon form av möjlighet till kommunikation mellan bilen och kontextmedvetna applikationer. Om så inte är fallet tappar bilen mycket av sitt värde som en miljö för kontextmedvetna applikationer eftersom möjligheterna att enkelt använda bilens sensorer blir mycket begränsade.

3 Behov för förenklad kontexthantering

För behovet av utvecklingsverktyg (se 2.8) har Dey et al (2001) identifierat flera viktiga punkter som jag tittar närmare på i detta kapitel. Om dessa behov uppfylls blir det enklare att skapa kontextmedvetna applikationer eftersom skaparna kan koncentrera sig på själva applikationen utan att behöva utveckla en massa andra lösningar för tekniska problem, till exempel hur sensorer skall kommunicera med applikationen.

Dey et als (2001) behov:

1. Separering av problem
2. Kontext tolkning
3. Genomskinlig, distribuerad kommunikation
4. Konstant tillgång till kontextdata
5. Kontext lagring och historia
6. Resurs upptäckande

3.1 Separering av problem

En av anledningarna till varför kontextmedvetenhet inte används oftare i applikationer är att det inte finns något enkelt eller vanligt sätt att insamla och hantera kontextdata. De flesta kontextmedvetna applikationer som finns hanterar kontextdatan på sitt eget sätt. De har alla löst insamlandet och hantering på de sätt som passade just dom bäst.

Vissa applikationer har sensorerna för kontext insamlande direkt inkopplade. Detta leder till att skaparna får skriva en massa kod som hanterar sensor inputen på det protokollspråk som sensorn använder. Detta leder till två problem:

Det blir jobbigt för skaparna att både hantera kommunikation med sensorer och dessutom använda kontexten på ett vettigt sätt. Det ökar kravet på skaparens kompetens som kodare.

Det andra problemet är att det inte är riktigt förenligt med god mjukvaruskapande praxis. Kod skall helst vara återanvändbar och det blir svårt att göra om varje kontextmedveten applikation använder sitt egna kontextinsamlingsätt.

Lösningen på dessa problem är att skapa en mera standardiserad applikation för inläsningen av kontextdata, Dey et al (2001) kallar den för *Widget*. Widgets återfinns till exempel i kommunikationen med den vanliga datormusen. Hur mycket svårare skulle programmering vara om alla program hanterade mus-input på olika sätt?

Denna Widget skulle hantera lågnivå input från sensorer och kunna presentera informationen på ett mera lättförståeligt sätt för de högre nivåerna i applikationen.

3.2 Kontext tolkning

Det finns ett behov av att utöka fråge- och svarsmechanismerna för att tillåta applikationer att hämta kontext från distribuerade datorer. Ofta handlar det om att kombinera

information från flera olika sensorer för att få fram ett meningsfullt kontext. Detta kan vara något så enkelt som att översätta ett smart card Id till dess ägares namn. Men också så komplexa uppgifter som att identifiera en person med hjälp av ett fotografi. Till exempel: En applikation vill bli meddelad om när ett informellt möte mellan två personer inträffar. På lägsta nivån måste platsinformation tolkas för att avgöra var de olika användarna befinner sig och deras identiteter måste vara kända. På nästa nivå bedöms ljudnivåer och scheman för att kunna avgöra om ett möte äger rum eller om de bara passerar förbi varandra.

Ur ett design perspektiv måste dessa olika nivåer vara transparenta. För att stödja denna transparens måste kontextinformationen ofta tolkas innan den kan användas av en applikation.

3.3 Genomskinlig, distribuerad kommunikation

I traditionella system kommer inputen från föremål anslutna direkt till datorn till exempel mus och tangentbord. I en kontextmedveten applikation kan sensorerna (kontext insamlarna) vara spridda över större avstånd till exempel över en mässlokal. Detta gör det svårt att ansluta alla sensorerna direkt till samma dator. Detta skapar ett behov av ett genomskinligt och distribuerat kommunikationsnätverk. Med ett sådant kan mycket designarbete sparas. Utan ett sådant nätverk måste designern skapa ett eget nätverk mellan de olika datorer som sensorerna är kopplade till och se till så att det skickar rätt kontextinformation på rätt sätt vid rätt tillfälle.

3.4 Konstant tillgång till kontextdata.

Dey et al anser att kontextmedvetna applikationer hela tiden skall ha tillgång till kontextdata. Applikationen skall inte behöva slå på en sensor för att få kontextdata. Dessutom bör andra applikationer kunna komma åt kontextdatan samtidigt. Detta leder till att komponenten (ofta en sensor) måste kunna agera självständigt. Detta förenklar för programmeraren som inte behöver instansiera, underhålla och hålla reda på komponenter/funktioner som hanterar kontextdatan. Till exempel telefonkopplingsfunktionen i Active Badge projektet (Want et al,1992). När ett samtal tas emot försöker systemet att koppla det till telefonen närmast mottagaren (mobiltelefoner var inte så vanliga 92). Systemet kunde inte lokalisera användaren om Active Badge servern inte var instansierad och Active Badge servern kunde inte instansieras om den användes av andra applikationer. Ett enkelt sätt att lösa konstant tillgång till kontextdata är att sensorn skickar information kontinuerligt eller när något inträffar.

3.5 Kontext lagring och historia

Ett behov som sammanhänger med konstant tillgång till kontextdata är behovet av att ha tillgång till historisk data för att till exempel kunna förutsäga trender. Kontextdatan skall alltså inte bara användas direkt utan den bör även lagras, i till exempel en databas, för att

kunna utnyttjas senare. Detta är något som Pascoe, Ryan & Morse (1998) använder sig av i sitt zoologiska system som lagrar användarens anteckningar tillsammans med tiden och platsen de gjordes på så att zoologen kan studera dessa när han återvänder från fältet.

3.6 Resurs upptäckande

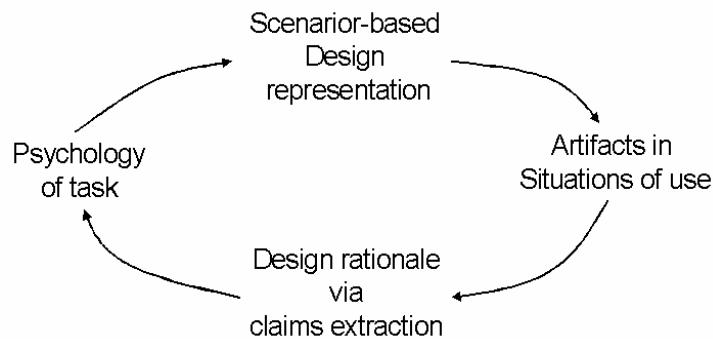
För att skall kunna kommunicera på ett meningsfullt sätt med en sensor måste applikationen veta vilken sorts kontextdata som sensorn kan bidra med och hur de skall kommunicera. För distribuerade sensorer så betyder det här att applikationen måste veta minst värnhamn och port till datorn som sensorn är ansluten till. För att kunna upptäcka det här behövs någon form av resurs upptäckande. Med denna resurs upptäckarfunktion skulle en applikation kunna fråga efter en viss sorts kontextdata och få ett relevant svar om och var kontextdatan finns från resurs upptäckarfunktionen.

3.7 Problem med kontext hantering

Dey et al (2001) skriver om flera olika problem vid skapandet av kontextmedvetna applikationer. Bland annat ett som är högst väsentligt för denna uppsats. På grund av det endast finns ett fåtal design och konceptuella verktyg är det lätt att designers väljer att använda kontextdata baserat på vilken hård eller mjukvara som finns tillgänglig. Detta sätt att närma sig problemet är kanske inte det allra bästa. Begränsningar hos sensorerna kan följa med upp i applikationsnivå och hindra eller begränsa applikationens fortsatta utveckling och evolution. Viktigare är att när valet av sensorer styr utvecklingen kan detta begränsa skaparen i dennes valmöjligheter och innovation.

4 Metod

4.1 Uppgift-artefakt cykeln



Figur 6. Uppgift-artefakt cykeln

Uppgift-artefakt cykeln är en metod som använder ett cykliskt sätt att arbeta, se Figur 6. Uppgift-artefakt cykeln. Den får sitt namn från konceptet att en uppgift är någonting som implicit sätter behov för utvecklandet av artefakter för att stödja uppgiften. Carrol, Kellog & Bossom (1991) argumenterar för cykeln genom att för sticka hål på två teknologiska myter. För det första myten att vår vetenskapliga förståelse av naturen systematiskt används för att skapa ny teknologi. För det andra att teknologiska innovationer beror på heroiska bedrifter av enskilda individer. Myterna skjuts i sank av ett exempel på hur ångmaskinen uppfanns. De flesta tror att James Watt byggde den första ångmaskinen 1775 efter att han studerat hur ånga sprutade ur en té-kittel. I verkligheten hade Watt i många år jobbat med att reparera ångmaskiner även om de var av en annorlunda och enklare design. Dessutom kan man spåra grundtanken till ångmaskinen ända tillbaka till 1200talets Kina. Watt hade alltså redan studerat enklare ångmaskiner innan han ”uppfann” sin variant och det tog honom 10år att bygga en acceptabelt fungerande sådan. Det rörde sig alltså inte om en blixtpå inspiration utan av en lång tid provade och studier av hur tidigare artefakter fungerade. Poängen som Carrol, Kellog & Bossom (1991) gör med detta är att tekniken utvecklas och omutvecklas hela tiden och att det inte är någon skillnad med den tekniska frammarschen inom människa-data interaktion. Uppgift-artefakt cykeln använder sig av detta sätt att arbeta. I ett projekt kan skaparna starta var som helst i cykeln.

Cykelns delar (se Figur 6. Uppgift-artefakt cykeln):

- Design rationale är en detaljerad beskrivning av en artefakts historia och mening.
- Scenario-based design är en uppgiftsbaserad teknik för visualiseringen av en artefakt *innan* den skapats.

- Psychology of task kritiseras av Carrol, Kellog & Bossom (1991) som en något vag punkt. Den skall beskriva detaljerade klassificeringar av mänskliga och artefakt domäner.
- Artefakten är helt enkelt föremålet som skall studeras/utvecklas. Till exempel ett ordbehandlingsprogram.

Det viktiga med task-artefakt cykeln är förståelsen av att teknologiska landvinningar inte uppstår av sig själva. Utvecklingen drivs framåt i små eller ibland stora steg av tidigare artefakter och ideer.

I uppgift-artefakt cykeln ingår skapandet prototyper in som en viktig del i främst Scenario-based design. Om prototypen är tillräckligt väl utvecklad kan den också tjäna som en artefakt på vilken ytterligare studier kan grundas, nya artefakter skapas också vidare vilket sluter cykeln. Jag har i projektet inte vandrat runt cirkeln utan vi började med en mycket kort design rationale och har därefter jobbat oss fram till att skapa en välutvecklad prototyp som förhoppningsvis kan tjäna som en artefakt för forskningen.

4.2 Prototyp som metod

Som en del av uppgift-artefakt cykeln finns skapandet av prototyper vilket också är den del av cykeln som vi främst har inriktat oss på.

Preece et al (2002) skriver att när du hör ordet prototyp kanske du tänker på något som till exempel en skalenlig modell av en bro eller en byggnad eller ett program som kraschar var femte minut. Men en prototyp kan också vara en eller flera teckningar av hur ett program skall se ut eller en video simulering av en uppgift och så vidare. Faktum är att en prototyp kan vara allt från en teckning av ett objekt till ett avancerat program.

Vad är nyttan av en prototyp? Prototypen är ett användbart föremål för att öka förståelsen både för den som skapar den såväl som omvärlden. Schön (1983) skriver att många designers anser att prototyp skapandet är en viktig del av i design processen. Prototyper svarar på frågor och stöder designers i deras val mellan olika alternativ. Därför stöder de en mängd olika skapande processer som till exempel om det är möjligt att genomföra en idé, för att klargöra kravspecifikationer, för att genomföra användartest och utvärdering eller för att undersöka vilken design som är kompatibel med resten av systemet. Preece et al (2002) delar upp prototyper i två olika kategorier, low-fidelity prototyp och high-fidelity prototyp.

4.2.1 Low-fidelity prototype

En low-fidelity prototyp liknar inte den färdiga produkten särskilt mycket. Den använder ofta material som är väldigt olika den färdiga produkten. Till exempel en serie skisser som beskriver ett användarscenario. Low-fidelity prototyper är användbara eftersom de tenderar att vara billiga, snabba och enkla att modifiera. De är inte avsedda för att integreras i den färdiga produkten utan skall endast användas för att utforska möjligheter.

4.2.2 High-fidelity prototyp

Dessa använder material som liknar det man kan finna i den färdiga produkten. Till exempel kod istället för skisser. Delar av high-fidelity prototyper kan också återanvändas i den färdiga produkten till exempel delar av koden i en prototyp som ingår i det färdiga programmet. High-fidelity prototyper har dock flera nackdelar som Retting (1994) tar upp:

- De tar lång tid att bygga
- Prototyp testaren tenderar att hänga upp sig på detaljer istället för att se helheten.
- Utvecklare kan vara motvilliga att ändra något de jobbat med i timmar.
- En mjukvaruprototyp kan sätta för höga förväntningar.
- En enda bugg i prototypen kan omöjliggöra testning.

Dessa nackdelar kan dock uppvägas av att high-fidelity prototyper är så pass lika den färdiga produkten vilket leder till att den är enklare att dra slutsatser av den. Dessutom kan det finnas mycket återanvändbart material i en high-fidelity prototyp.

4.3 Metod val

Idealet för metoden hade varit att konstruera en fullt fungerande prototyp och sedan prova den i dess rätta miljö med en mindre etnografisk studie, till exempel den så populära "Quick and Dirty" metoden. Detta är tyvärr inte möjligt under den begränsade tid som finns tillgänglig. Jag har ingen möjlighet att prova prototypen i dess rätta miljö, dvs en SAAB 93, förens till hösten. Det är däremot möjligt att prova prototypen i en simulerad miljö där en stationär dator med bluetooth skickar simulerad kontextdata till handdatorn. Detta sätt har ringa värde för mig eftersom etnografi handlar om att studera fenomen i deras *naturliga* miljö. Att låta användare prova prototypen i en simulerad miljö för att skaffa etnografisk data skulle troligen inte ge ett resultat som har några större likheter med hur ett resultat av en etnografisk undersökning av prototypen i dess rätta miljö skulle ge. En etnografisk undersökning av prototypen i en simulerad miljö skulle kunna ge mycket information om vi har lyckats skapa ett underhållande spel. Det viktiga och intressanta ur forskningsperspektiv är dock hur vi har lyckats med användningen av kontext informationen och detta är mycket svårt att göra i en simulerad miljö. Det finns en mängd olika funktioner i prototypen, som till exempel de negativa effekterna om föraren av fordonet överstiger hastighetsgränsen, som måste upplevas i dess rätta miljö för att kunna studeras.

På grund av detta har jag valt att basera min metod på konstruktionen av en prototyp och litteraturstudier. Min förhoppning är att denna prototyp till hösten skall kunnas provas och användas i sin rätta miljö och att detta kan bidra till forskningen om kontextmedvetenhet.

Eftersom kravspecifikationen från våra uppdragsgivare specificerade en prototyp med mycket hög funktionalitet var det ett enkelt val att göra en high-fidelity prototyp men eftersom den prototypen behövde så pass hög funktionalitet gjorde vi först en low-fidelity prototyp så att vi skulle få en bättre överblick och ökat samförstånd mellan de inblandade.

4.4 Bakgrund till metoden

Jag kom kort innan jul i kontakt med Viktoriainstitutets telematikgrupp och blev erbjuden att delta i ett av deras projekt. Projektet ställde en rad olika frågor inom området kontextmedvetenhet och var därför en bra start till en uppsats om kontextmedvetenhet. Detta projekt är skapat som ett underprojekt till ett samarbete mellan Viktoriainstitutet, Saab Automobile, Mecel och Vodafone.

Storprojektets bakgrund:

”Inom både forskning och näringsliv har stor uppmärksamhet riktats mot de tekniska utmaningar som är förenade med att integrera kommunikations- och informationsteknik med bilteknik. Idag finns det således ett flertal fungerande plattformar och tillämpningar för telematiktjänster. Dessa initiativ har fötts och genomförts i ljuset av optimistiska prognoser om framtida tillväxt inom telematikområdet. Analysfirman McKinsey förutspår exempelvis en marknad på 40 miljarder dollar vid 2010. Det saknas dock kunskap bland biltillverkare, telecomleverantörer och teleoperatörer om hur telematiktjänster ska utvecklas för att generera användarnytta. Utan denna kunskap kommer sannolikt inte förhoppningarna om en hållbar tillväxt inom telematikområdet att infrias.”

Storprojektet syftar till att:

”...identifiera och analysera hur innovativa telematiktjänster för bilen kan integreras och komma till nytta i olika användningssammanhang. Projektet kommer att inriktas på fem speciella användargrupper: tjänstebilsinnehavare, nya körkortsinnehavare, kvinnor, baksätesspassagerare och 'fleet operators'.”

Storprojektet har identifierat ett antal nyckeltjänster, bedömda som särskilt intressanta för utveckling och utvärdering under våren och hösten 2003. En av dessa tjänster är kontextmedvetna spel på handhållna datorer för medpassagerare, vilket avser att inte enbart underhålla passagerare utan också att göra det på ett sätt som skapar medvetenhet om den kontext en bil i rörelse färdas i, på ett sätt som adderar mervärde till resan i sig, likväl som till spelet. Det ska bli roligare att vara passagerare under en bilresa. Det underprojekt som jag deltog i inriktar sig på – PDA Gaming Using Vehicle and Location Data – och riktar sig främst till kategorin baksätesspassagerare, även om naturligtvis medpassagerare fram även ingår i målgruppen.

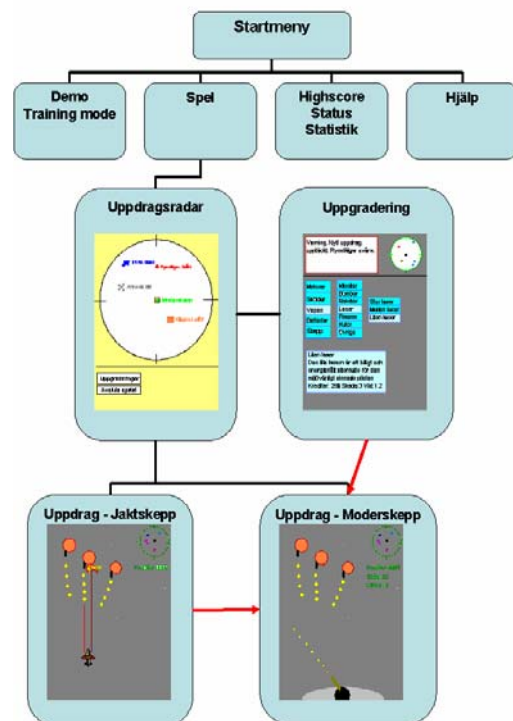
Tjänstespecifikationen för projektet som tagits fram av storprojektet innehöll följande: Det skall vara ett spel som stimulerar medvetenhet om hur bilen presterar och om omgivningen (kontexten) som bilen färdas igenom. Spelet skall också uppmuntra säker och miljövänlig körning dessutom skall det vara underhållande att spela. Spelet skall kommunicera med bilen via Bluetooth. För att få omgivningsdata skall spelet innehålla en kopia av vägverkets databas, detta ändrades senare till TeleAtlas vägkarta eftersom vägverket ansåg att deras databas var för komplex.

5 Resultat

I detta kapitel beskrivs hur metoden realiserades och resultatet av detta. För att förstå kapitlet rätt är det viktigt att läsaren förstår vad jag menar med ”användningsmiljö”. Med detta menar jag den miljö som applikationen befinner sig i och använder sig av. Fokus är alltså på applikationens användande av miljön inte en persons användande av applikationen.

5.1 Utvecklingsarbetet

Vi började i mitten av januari med en flera dagar lång diskussion om hur vi skulle göra spelet för att bli så underhållande och kontextmedvetet som möjligt, därefter ritades objektmodeller och användarscenarion skrevs. Halvvägs genom objektmodeller och användarscenarion dök en skiss upp som tycktes förena det bästa av båda metoderna. Denna low-fidelity prototyp har sedan kommit till stor användning under hela arbetet genom att den sammanfattade vår gemensamma syn på hur den färdiga produkten borde se ut och fungera. Skissen satte vi upp på en anslagstavla mittemellan våra arbetsstationer så att den alltid var synlig. Se Figur 7. Grundskissen. Därefter påbörjade vi arbetet med high-fidelity prototypen. Eftersom det redan var bestämt att prototypen skulle köras på ett pocket-pc system så var det enklaste valet att koda i C++ med hjälp av Visual Studio .NET eftersom det redan innehåller färdiga funktioner för att porta över program till pocket-pc. Vi valde också att använda oss av GapiDraw som plattform vilket avsevärt förenklade skapandet av grafiken. Arbetet med high-fidelity prototypen var en mycket föränderlig process där funktioner lades till och togs bort allteftersom prototypen växte fram. Vi använde till exempel först textfiler för att lagra information men gick sedan över till xml, slutligen kom vi fram till att en databas skulle bli bästa lösningen eftersom vi hade så stora mängder information att hantera därför slutade vi använda textfiler och xml. Arbetet med prototypen tog, inte oväntat, mer tid än vi hade räknat med precis som Retting (1994) ansett.



Figur 7. Grundskissen

5.2 CABdriver

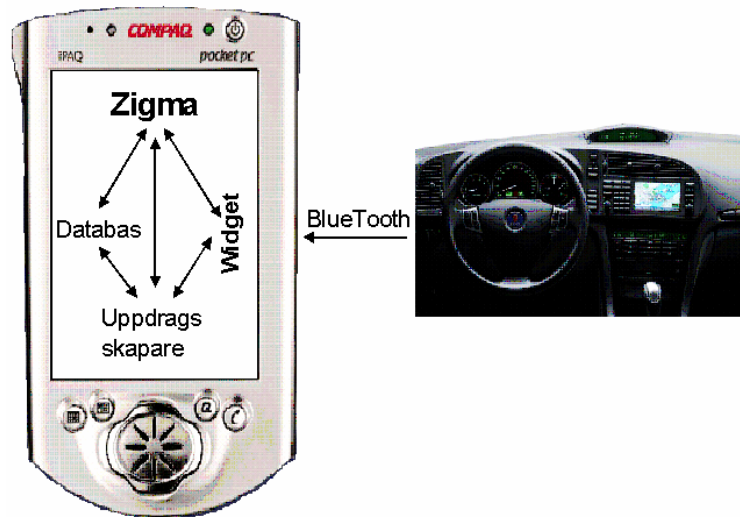
CABdriver är namnet på vår prototyp, i denna del av kapitlet beskrivs CABdrivers teknik och funktioner närmare.

5.2.1 Teknisk beskrivning

CABdriver är skapat i C++ och använder sig av GapiDraw för att förenkla skapandet av grafiken. Nästan ingen information är hårdkodad i spelet utan den hämtas från en Sybase databas med hjälp av ODBC. Användandet av en databas gör att det blir lätt och snabbt att hämta rätt information. Något som annars kunde bli svårt med tanke på att de

vägar kartor från TeleAtlas vi använder är mycket omfattande i storlek och CABdriver ofta behöver fråga databasen om lokala förhållanden, till exempel hastighetsgränser. Valet av Sybase som databas beror på att de erbjuder en bra databas som lätt kan portas till en pocket pc.

CABdriver, widgeten, uppdragsskaparen och databasen ligger på en handdator som sedan kommunicerar med bilen med hjälp av Bluetooth. Se Figur 8. Dataflöden.



Figur 8. Dataflöden

5.2.2 CABdriver i korthet

CABdriver är ett "Combat Game" enligt Crawford (1997) eftersom det handlar om att placera sig på ett sådant sätt att spelaren kan skjuta datorn men datorn inte kan skjuta spelare. Det finns många typer av Combat Games, som underkategori klassas spelet som en "Space shooter" vilket är den vanligaste och populäraste kategorin inom Combat Games. Som andra exempel på Space Shooter kan nämnas de legendariska spelen Space Invaders och Asteroids av Atari. Nu följer en kort beskrivning av spelet, därefter skall jag redogöra mer omfattande för spelets olika delar.

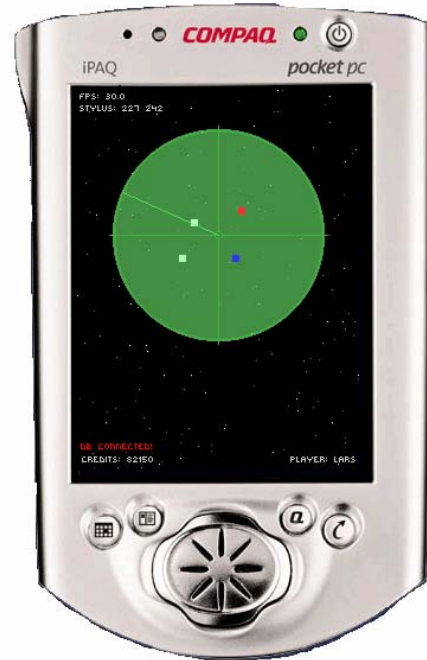
Spelaren flyger ett rymdskepp i olika uppdrag och försöker förstöra så många fiender som möjligt utan att själv bli förstörd. Uppdragen skapas med kontextdata som grund, de är alltså baserade på "verkligheten" runt bilen. Uppdragen väljs från en radarskärm och sedan genomförs de. Vyn är 2d sett ovanifrån. Spelfältet (skärmen) scroller framåt i vertikalled med en konstant hastighet, spelaren kan röra sig fritt inom spelfältet men inte utanför det. Spelarens rymdskepp pekar alltid framåt i spelfältets scrollerriktning även om spelaren färdas åt sidorna eller bakåt. För varje fiende som förstörs erhåller spelaren

krediter. Spelaren kan även få krediter för speciella händelser. Krediterna kan spelaren sedan använda mellan uppdragen för att reparera eller förbättra sitt rymdskepp. Det finns också ett moderskepp som är spelarens hemmabas. Moderskeppet färdas, på samma sätt som bilen, genom en rymd som är baserad på ”verkligheten” runt bilen. Moderskeppet kan under vissa omständigheter komma under attack och då får spelaren försvara det på ungefär samma sätt som ett vanligt uppdrag. CABdriver gränssnitt består i huvudsak av 3 delar som beskrivs mer omfattande nedan

5.2.2.1 Uppdragsradarn

Uppdragsradarn visar en ”radar” bild av vilka uppdrag som finns runt bilen inom 2 kilometers radie. Vad denna radarbild innehåller beror på vilken kontextdata som uppdragsskaparen väljer att använda sig av och baserar uppdragen på.

Radarn innehåller också en kort förklaring om uppdragets natur till exempel att uppdraget baseras på ett fornminnes märke vid namn ”Ingelinge Hög” och att spelaren kan förvänta sig att uppdraget är lätt. Moderskeppet ligger centrerat i radarn. Spelaren väljer ett uppdrag genom att klicka på det. När spelaren har valt ett uppdrag startar det. Om moderskeppet kolliderar med ett uppdrag på radarn får spelaren välja om han vill försvara moderskeppet eller inte. Om spelaren väljer att försvara moderskeppet förflyttas han till ”Moderskepps försvar”, om spelaren väljer att inte försvara det får denne ett kredit avdrag. Radarbilderna förflyttar sig hela tiden i förhållande till bilens hastighet och position. Se Figur 9. Uppdragsradarn.



Figur 9. Uppdragsradarn

5.2.2.2 Uppdrag – Jaktskepp

När uppdraget startar börjar striden. Spelaren flyger ett rymdskepp och blir anfallen av fiender som försöker förstöra spelaren på olika sätt. Spelet styrs med knapparna på handdatoren. Vyn är 2d sett ovanifrån. Spelfältet (skärmen) scrollar framåt i vertikal riktning med en konstant hastighet, spelaren kan röra sig fritt inom spelfältet. Spelarens rymdskepp pekar alltid framåt i spelfältets scrollriktning även om spelaren färdas åt sidorna eller bakåt. Fiender förstörs

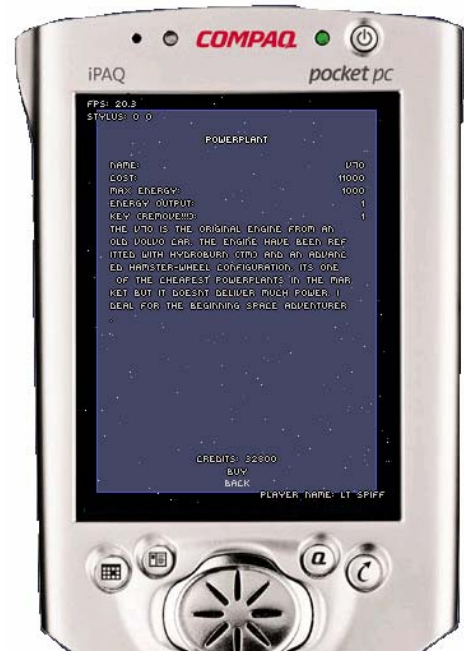


Figur 10. Uppdrag jaktskepp

genom att spelaren skjuter dem med sitt rymdskepps vapen. För varje fiende som förstörs erhåller spelaren krediter. Spelaren kan även få krediter för speciella händelser. Radarn visas i en förminskad version under uppdraget så att spelaren kan se om moderskeppet riskerar att hamna under attack. Om moderskeppet hamnar under attack visas ett varningsmeddelande på skärmen. Uppdraget avslutas efter en viss tidsrymd, efter ett visst antal förstörda fiender, om spelaren blir förstörd av fienden, om spelaren väljer att avbryta uppdraget till exempel för att skydda moderskeppet. Ett uppdrag varar i genomsnitt en eller ett par minuter. Efter ett framgångsrikt avslutat uppdrag erhålls en kredit belöning och uppdraget försvinner från radarn. Efter att uppdraget avslutas återvänder spelarens skepp automatiskt till moderskeppet. Se Figur 10. Uppdrag jaktskepp.

5.2.2.3 Uppgradering

Menyn är endast åtkomlig när spelaren är i Mission control läge. I denna meny kan spelaren använda sina intjänade krediter för att köpa eller sälja förbättringar till sitt jaktskepp och moderskepp. Exempelvis större kanoner, missiler, sköldar, motorer, batterier mm. Se Figur 11. Uppgradering.



Figur 11. Uppgradering

5.2.3 Framtida utvecklingsarbete på CABdriver

Själva spelet och databasen är färdiga och fungerar på handdatorn. Widgeten fungerar på pc men inte handdatorn. Kommunikationen med hjälp av Bluetooth mellan bil och handdator finns inte än. Eftersom den testdata vi har haft tillgång till varit mycket begränsad och endast simulerad har vi inte kunnat skapa någon uppdragsskapare. Uppdragsskaparen skapar uppdrag åt spelaren med hjälp av kontextdata och är en viktig detalj för att spelet skall kunna bli kontextmedvetet.

5.3 Användarscenario

Här presenteras ett tänkbart användarscenario

Det är första dagen på sommarlovet och 13 årige Kalle och hans mamma och pappa skall åka till deras sommarställe i Havstensund, 15 mil norr om Göteborg. Familjen sätter sig i sin SAAB 93 och Kalle startar upp CABdriver på sin handdator.

Pappa har bråttom och kör i 65 på 50 vägen i deras kvarter. Sakta ner! Skriker Kalle eftersom Kalle har sett på uppdragsradarn att hans moderskepp är jagat av polismonster och han har lärt sig att polismonster bara dyker upp när det går för fort. Kalles pappa saktar ned och polismonstren försvinner.

Kalle är lite osäker på hur de skall köra så han frågar pappa om vilken riktning de skall köra i och väljer därefter ett uppdrag som verkar ligga åt det hållet för att minska chansen för en eventuell krock mellan moderskeppet och uppdraget.

Kalle spelar igenom uppdraget och upptäcker sedan att de närmar sig McDonalds Bäckebo. Kalle beslutar sig för att vänta med ett nytt uppdrag till dess att McDonalds uppdraget dyker upp, han vet nämligen att det brukar dyka upp ett bra uppdrag vid McDonalds restauranger. Han passar också på att tjata lite på mamma att det skall stanna och äta på McDonalds men det blir ett blankt nej. McDonalds uppdraget dyker upp på radarn och kalle spelar igenom det och tjänar en massa krediter. Kalle sparar krediterna eftersom han hoppas att det skall finnas någon ny bra uppgradering att köpa i Havstensund som inte finns i Göteborg.

På radarn ser Kalle att de närmar sig ett fornminnes-uppdrag. Eftersom han är väldigt förtjust i historia börjar han genast att spana efter vad det kan vara för något samtidigt som han kollar på radarn för att se vilken position det har i förhållande till bilen. Efter ett tag tonar Kungälv's fästning majestätiskt upp sig vid sidan av vägen och Kalle njuter av synen och spelar sedan uppdraget.

Kalle börjar spela ett nytt uppdrag. Snart kommer familjen in i Kungälv och det blir lite rörigt för föraren i en korsning. CABdriver registrerar att föraren har mycket att göra och troligen inte vill bli störd. Därför förändras uppdraget så att Kalle måste vara mer koncentrerad på det och inte har tid att störa föraren. När föraren inte är lika upptagen så återgår CABdriver till normal läget.

Senare ser Kalle på radarn att de närmar sig en järnvägs-korsning så han ber pappa att sakta in så att han kan titta på tågen. Kalles pappa blir mer uppmärksam på sin omgivning eftersom han inte vill bli överraskad av något tåg på den obehagade järnvägs-korsningen. Efter ytterligare ett antal uppdrag är de framme i Havstensund. Kalle är nöjd med resan och han fick se Kungälv's fästning också. Kalles föräldrar är också nöjda eftersom Kalle har hållit sig lugn och inte stört dem. Kalles pappa som körde är dock lite irriterad över att han inte fick köra för fort för Kalle.

5.4 Metodkritik

Att skapa prototypen tog enormt mycket tid, en grov uppskattning hamnar på mer än 1000 arbetstimmar ($160\text{tim/månad} * 3\text{månader} * 2\text{personer} = 960 + \text{ytterligare arbete av andra projektinblandade}$) frågan är om det verkligen behövdes skapas en så pass avancerad prototyp för att komma fram till slutsatsen.

Prototypen överensstämmer inte med Dey et al (2001) åsikt att kontextmedvetna applikationer inte skall skapas med tillgången på sensorer som utgångspunkt.

6 Diskussion

6.1 Vad är CABdriver?

Min definition av kontextmedvetenhet säger att ” En kontextmedveten applikation är en applikation som är medveten om delar av omgivningen och kan reagera på detta.”

CABdriver är medveten om omvärlden genom bilens sensorer och TeleAtlas vägkarta.

Den reagerar på input från omvärlden på ett flertal olika sätt. Om kontexten förändras förändras också CABdriver. Därför är CABdriver en kontextmedveten applikation.

Vilken sorts kontextmedveten applikation är då CABdriver? Dey et al (2001) har delat in kontextmedvetna applikationer i tre klasser:

1. Presentation av information och tjänster.
2. Automatiskt utförande av tjänster
3. Lagring av kontext information för senare användning

CABdriver är främst en applikation av klass 1 och 2.

Klass 1 främst genom att CABdriver presenterar kontext information för användaren genom att ge denne en uppfattning om hur det ser ut i en 2 km radie runt bilen.

Klass 2 genom att CABdriver ändrar sig beroende på kontext förändringar till exempel genom att spelaren blir jagad av polis monster om föraren kör för fort.

CABdriver lagrar ingen kontext information som används senare av CABdriver, det enda som lagras är uppgifter om spelaren som till exempel antal krediter. Däremot lagras widgeten kontextinformation den används dock inte av CABdriver.

6.2 Utmaningar för kontextmedvetenhet

Jag skall i detta avsnitt diskutera hur moderna bilar generellt och SAAB 93 och CABdriver specifikt relaterar och kan bidra till forskningen om hur Olssons (2003) åtta utmaningarna skall lösas.

Olssons (2003) åtta utmaningar:

1. Begreppet kontext
2. Kontextmedvetenhet är baserad på mer än positionsdata ANVÄND
3. Användbart gränssnitt ANVÄND
4. Behovet av en infrastruktur
5. Design frågor
6. Sociala aspekter
7. Utmaningar för ett vetenskapligt tillvägagångssätt inom ett visionärt område
8. Behovet av utvecklingsverktyg, denna utmaning finns noggrant beskriven i stycke 6.3

Begreppet kontext

Begreppet kontext har diskuterats mycket och en rad olika definitioner har föreslagits. Gemensamt för forskningsvärlden tycks vara behovet av en definition som alla accepterar och använder. Med hjälp av en gemensam definition skulle det också vara möjligt att med större precision utreda huruvida en applikation är kontextmedveten eller ej. Är till exempel en luftkonditioneringsanläggning kontextmedveten eller, i andra ändan av skalan, ett navigationssystem för bilen. Denna uppsats bidrag till en gemensam definition är inte mer än efterlysandet av just denna gemensamma definition.

Kontextmedvetenhet är baserad på mer än positionsdata

Schmidt et al (1998) drar slutsatsen att ju mer information en kontextmedveten applikation har om kontexten desto större blir användarnyttan. Jag anser att Schmidt har delvis rätt men att det finns någon form av övre gräns för användbar kontextinformation. Vissa typer av kontextinformation kan visa sig helt onödig eftersom användaren omöjligt kan ha nytta av denna. CABdriver ligger i framtiden bra positionerad för att utreda vilka sensorer i bilen som kan komma användaren till nytta. Det är i SAAB 93 möjligt att överföra information till en handdator om hur långt det är kvar till nästa 5000mil service. Detta är ett värde som användaren enkelt kan räkna ut själv genom att titta på milräknaren men det behövs egentligen inte eftersom bilen själv tänder en lampa som meddelar om att det är dags för service. Är detta kontextdata som är intressant för användaren att ha tillgång till i sin kontextmedvetna applikation? Med CABdriver finns det möjlighet att försöka utreda vilka av bilens sensorer som kan användas på ett bra sätt.

Användbart gränssnitt

Ett användbart gränssnitt är naturligtvis viktigt och är en stor utmaning inom ubiquitous computing eftersom många av dessa föremål har eller ger möjlighet till annorlunda former av gränssnitt. Brunnberg & Juhlin's (2003) innovativa baksätets spel bevisar att det är möjligt att skapa ett underhållande spel för baksätets passagerare med hjälp av kontextdata och deras annorlunda gränssnitt men de pekar också på de svårigheter som kan finnas med att välja ut ett, för spelaren, intressant kontext.

Moderna bilar som miljö för kontextmedvetna applikationer ställer också höga krav på att gränssnittet är användbart. Applikationer som riktar sig mot föraren av bilen måste till exempel ta hänsyn till att denne kan ha svårt att använda ögon och händer till annat än körningen. Även för baksätets passagerare ställs nya krav på gränssnittet till exempel kan ljus och ljudförhållanden i en bil varierar mycket mer än i en stationär miljö och det kan vara obekvämt att ha ett helt tangentbord i bilen.

CABdriver använder sig enbart av traditionella och beprövade former av kommunikation mellan handdator och användare, det vill säga knapparna på handdatorn. Däremot är kommunikationen mellan omgivningen (kontexten) och handdatorn mera otraditionell. Här kommer den mycket intressanta frågan in om hur kontextinformationen kan förmedlas på bästa sätt till användaren. Eftersom CABdriver är ett spel så kan all kontextinformation användas åtminstone som någon form av slumpgenerator men detta är inte ett meningsfullt sätt att använda kontextinformationen på eftersom man då lika gärna kan använda en vanlig slumpgenerator. Det är viktigt att utreda hur den kontextinformation som designern har bedömt som värdefull kan presenteras för användaren på ett sådant sätt att det ökar användarvärdet. Detta är en av kärnfrågorna

som CABdriver förhoppningsvis kan hjälpa till att svara på då det till hösten finns goda möjligheter att prova på hur användaren reagerar och hanterar olika typer av kontextinformation som CABdriver presenterar.

Ta exemplet från användarscenariot där användaren (Kalle) blir jagad av polismonster på grund av att bilens förare kör för fort. Hur kommer Kalle att tolka detta? Inser han att monstren kommer på grund av förarens hastighet och upplever han detta som något negativt?

Behovet av en infrastruktur

Behovet av en global infrastruktur för att stödja kontextmedvetna applikationer som Banavar & Bernstein (2002) beskriver skulle möjliggöra en rad nya kontextmedvetna funktioner som till exempel automatiskt överförande från en nätverkstjänst till en annan beroende på tillgång av bandbredd och behov. Widgeten i CABdriver fyller till viss mån detta behov men *bara* i bilen. CABdriver ger inte heller någon möjlighet att närmare undersöka hur skapandet av en sådan infrastruktur kan underlättas.

Design frågor

Fisher (2001) skriver att det är av viktigt för kontextmedvetna (och andra) applikationer att känna till och kunna dra slutsatser inte bara av en användares handlingar utan av alla. På detta sätt anser Fisher (2003) att användarnyttan skulle kunna ökas. Eftersom CABdriver är ett Singelplayer spel utan möjlighet till uppkoppling till exempel Internet så finns det ingen möjlighet för applikationen att dra slutsatser om andra användare. Detta gör att CABdriver inte kan användas för att undersöka denna design fråga.

Sociala aspekter

CABdriver har goda möjligheter att i framtiden kunna svara på en del olika frågor som finns runt kontext användning i bilen. Eftersom CABdriver är ett av de första (Se även Brunnberg & Juhlin 2003) kontextmedvetna spelen för bilar har det potentialen att ligga till grund för en undersökning av de sociala aspekterna för kontextmedvetna applikationer i bilar.

De sociala aspekterna av kontextmedvetna applikationer kan vara större än de för ”traditionella” program eftersom de interagerar på ett helt annat sätt än omgivningen. Ta till exempel användarscenariot där Kalle ber pappa att sakta ner till laglig fart eftersom det är negativt för Kalles spelande om pappa kör för fort. Hur de aktiva (Kalle) och passiva (Pappa) användarna att reagera på en sådan händelse? Det är möjligt att Kalles pappa saktar ned och då har CABdriver uppfyllt ett viktigt syfte. Det finns också en rad andra möjligheter på hur användarna kommer att reagera. Kalle kanske tycker att polismonstren är roliga att spela mot och uppmanar pappa att gasa. Kalles pappa kanske inte tillåter Kalle att spela eftersom det är så irriterande att han tjarar om hastigheten hela tiden.

Det är också viktigt enligt projekt specifikationen att användaren inte bara vänder sig inåt mot spelet utan även utåt mot omgivningen och på det sättet får en större behållning av själva resan än att bara ha suttit och tittat på en skärm hela tiden, detta skall också förhoppningsvis leda till en större social interaktion mellan användaren och de andra i bilen. Hur uppnår man detta? Vi försöker i CABdriver göra det genom att spelaren kan straffas om han inte vet vad som kommer att hända längre fram på vägen till exempel en

kurva som sätter hans skepp på kollisionskurs med ett uppdrag. Samtidigt försöker vi stimulera spelaren genom att sätta intressanta uppdrag på intressanta platser, som exempelvis borgar, men vi har inte hittat något bra sätt att belöna spelaren för att han tittar ut, än så länge bestraffas bara spelaren om han inte är medveten om omgivningen runt bilen men även detta kan vara en form av belöning enligt Hopson (2001). Det finns en stor mängd obesvarade frågor om hur kontexten skall kunna användas på bästa sätt, en felaktigt använd kontext kan få negativa följder. Eftersom CABdriver inte är ett nyttoprogram för användaren utan snarare ett nöjesprogram är det inte bara viktigt hur kontexten används utan även att den används på ett, för användaren, underhållande sätt. Om CABdriver inte är underhållande finns det inga större anledningar för användaren att använda det. Det är troligt att när CABdriver är färdigt kommer det att testköras i sin rätta miljö och en studie av CABdriver och dess sociala effekter kommer att genomföras. Resultatet av denna studie kan huvudsakligen indikera två saker enligt Olsson (2003). Ett, att artefakten (CABdriver) behöver designas bättre med färre begränsningar – alltså en misslyckad design – men även ett misslyckande kan, två, hjälpa oss att förstå de sociala aspekterna och därför hjälpa till med en ökad förståelse för kontextmedvetna applikationer, särskilt i bilar.

Utmaningar för ett vetenskapligt tillvägagångssätt inom ett visionärt område

Olsson (2003) konstaterar att forskning om kontextmedvetenhet är svår att bedriva eftersom det ofta handlar om visioner och framtida scenarier. Avancerade kontextmedvetna applikationer är än så länge ovanliga i vårt samhälle vilket begränsar den vetenskapliga erfarenheten och möjligheten att dra slutsatser av tidigare applikationer. Kanske kommer de framtida studierna av CABdriver bidra med någon liten del i utvecklingen av nya verktyg för att studera kontextmedvetna applikationer med.

6.3 Behov för förenklad kontexthantering

Det finns ett stort behov av utvecklingsverktyg för kontextmedvetna applikationer. Dey et al (2001) har närmare dokumenterat vilka faktorer som behöver uppfyllas för att skapa ett sådant verktyg. Jag skall nedan utreda hur SAAB 93 och widgeten uppfyller dessa faktorer.

Dey et al (2001) behov:

1. Separering av problem
2. Kontext tolkning
3. Genomskinlig, distribuerad kommunikation
4. Konstant tillgång till kontextdata
5. Kontext lagring och historia
6. Resurs upptäckande

Genomskinlig och distribuerad data uppfylls till viss mån av att bilens system håller reda på all data och den lätt kan fås i ett enkelt format. Problemet är att bilens system behöver någon form av program för att sända och ta emot data via Bluetooth. Bilen sänder alltså

inte automatiskt rätt data. Mecel utvecklar dock just en sådan applikation för SAAB 93 så att detta problem är förhoppningsvis löst inom den närmaste framtiden.

Konstant tillgång till kontextdata uppfylls så länge bilen är igång, så länge tändningen är tillslagen så sänder bilen data. Det uppfylls inte om bilen är avstängd eller om användaren lämnar bilen mer än ett par meter, Bluetooth har en mycket begränsad räckvidd. Eftersom bilen i sig är en klart definierad och isolerad miljö som man tydligt lämnar när bilen lämnas är detta inte något problem.

Kontext tolkning uppfylls till stor del naturligt av bilens eget system. Till exempel så tolkar bilen själv bränsleförbrukningen till liter per mil. All input som bilen får från sina sensorer översätts till siffervärden och är på det sättet tolkade. Kontext tolkning i bilen är enkelt och naturligt till skillnad från till exempel en videokamera som ger ett mycket mera svårtolkad information för en applikation.

Separering av problem, Kontext lagring och historia och Resurs upptäckande uppfylls inte av bilens eget system här behövs det nya applikationer och funktioner. Det är här vår prototyp CABdriver och widgeten kommer in.

Separering av problem behövs så att skaparna av kontextmedvetna applikationer inte behöver skriva egna applikationer för varje sensor de behöver kommunicera med. Lösningen på detta problem kallar Dey et al (2001) för en "widget" och den sköter kommunikationen mellan sensorer och kontextmedvetna applikationer. Det går att göra en grov jämförelse mellan en widget och ett operativsystem om man antar att användaren är en kontextmedveten applikation och själva datorn är en sensor. Widgeten är alltså en sorts mellanlager. Vi har till CABdriver utvecklat en fristående widget som kommunicerar och hämtar kontextdata från bilen och sedan skickar den till CABdriver. Vår widget blev framgångsrik och skall till hösten användas i en kurs i Mobil tjänster och telematik programmet på IT-universitetet.

Kontext lagring och historia behövs så att kontextdatan kan användas vid ett senare tillfälle. Vår widget sköter även denna uppgift genom att skriva kontextdatan till en databas. Eftersom vi använder oss av ODBC behöver det inte vara en specifik databas utan det går bra med de flesta.

Resurs upptäckande går inte riktigt att applicera på miljön i bilen eftersom alla sensorerna alltid är "upptäckta" och i kommunikation med widgeten. Dock, om en ny sensor skulle installeras i bilens miljö skulle den inte upptäckas utan den skulle behöva hårdkodas in i widgeten.

Av de 6 punkter som Dey et al (2001) satt upp för att förenkla skapandet av kontextmedvetna applikationer uppfyller SAAB 93 och vår widget 5 av dem. Med kunskap om bilens olika sensorer och tillgång till widgeten är det lättare att skapa kontextmedvetna applikationer, skaparna behöver inte skaffa en mängd olika sensorer och skriva protokoll för att kommunicera med dem. De kan istället koncentrera sig på själva applikationen. Det är viktigt att tänka på att punkterna bara är uppfyllda i en SAAB

93 miljö de är alltså inte sanna i någon annan miljö dessutom är inte widgeten helt klar men kommer att bli det snart förhoppningsvis. Detta begränsar användbarheten och mångsidigheten av lösningarna. Det är teoretiskt möjligt att skapa en widget som kan hantera input från alla tänkbara sensorer i alla tänkbara kontext men den skulle bli så otroligt omfattande och komplex att det nog är mycket svårt.

Det finns säkerligen en mängd olika andra miljöer än fordon som är välutrustade med olika typer av sensorer som lämpar sig för kontextmedvetna applikationer men att identifiera dessa tillåter inte den begränsade tid jag har på mig.

Dey et al (2001) skriver att designers *inte* bör utgå från tillgången av sensorer när de skapar kontextmedvetna applikationer eftersom det kan begränsa innovation och utveckling. Detta är inte vad vi har gjort med prototypen utan vi använde istället de tillgängliga sensorerna som utgångspunkt för CABdriver. Detta är också nyttan med CABdriver; att framtida designers och forskare har tillgång till ett färdigt nät av sensorer. Att detta strider mot Dey et al (2001) åsikter behöver dock inte vara ett problem om man ser bilen som en komplett, tydligt avgränsad miljö där det finns ett antal sensorer som designern kan välja och vraka emellan. Dessutom är det fortfarande möjligt för designern att lägga till nya sensorer om han vill. Även om dessa inte stöds av widgeten är det fortfarande möjligt att göra det även om det kan kräva en del arbete.

7 Slutsats

Syftet med denna uppsats är att besvara frågeställningen: *Hur lämpar sig moderna bilar som användningsmiljö för kontextmedvetna applikationer?* Med användningsmiljö så menas miljön som applikationen befinner sig i och använder sig av och inte användarens miljö. Till exempel så är Windows användningsmiljö en PC.

För att svara på detta skapades en prototyp, CABdriver. Denna visar att det är möjligt att skapa denna kontextmedvetna applikation som använder sig av en mängd olika kontextdata utan att lägga ned särskilt mycket tid på själva hämtningen av kontexten till programmet. I diskussionen visas att moderna bilar löser vissa av utmaningarna för kontextmedvetna applikationer. Särskilt de behov av utvecklingsverktyg som Dey et al (2001) efterlyser. Slutsatsen blir att moderna bilar med dess rikedom på sensorer kan förenkla skapandet av kontextmedvetna applikationer för dem. Eftersom många av problemen med kontextmedvetna applikationer härrör från tillgång till sensorer anser jag att moderna bilar är en lämplig miljö för kontextmedvetna applikationer eftersom de löser många av dessa problem.

8 Referenser

Andersen consulting. (2002) Mobile Multimedia study, European Commission Directorate-General Information Society

Banavar, G. & Bernstein, A. (2002) Software infrastructure and design challenges for ubiquitous computing applications. Communications of the ACM. Vol. 45, No. 12, pp. 92-96.

Benerecetti, M., Bouquet, P. & Bonifacio, M. (2001). Distributed Context-Aware Systems. Human-Computer Interaktion, Volume 16, pp. 213-228.

Brunnberg, L. & Juhlin, O. (2003). Movement and Spatiality in a Gaming Situation – Boosting Mobile Computer Games with the Highway Experience. Proceedings of Interact 2003, Zürich, Switzerland.

Carroll, J. M. Kellogg, W. A. & Rosson, M. B. (1991) The task-artifact cycle. Designing interaction: psychology at the human-computer interface. Cambridge University Press p 74-102

Crawford, C. (1997). The art of computer game design. Webresurs <http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html>

Dey, A. K., Gregory, D. A. & Salber, D. (2001). A conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applikations. Human-Computer Interaction, Volume 16, pp. 229-241.

Easterby-Smith, M., Thorpe, R. & Lowe, A. (1991). Management research an introduction. London: United Kingdom: SAGE Publications Ltd

Fisher, G. (2001). Articulating the task at hand and making information relevant to it. Human-computer interaction, Volume 16, pp. 243-256.

Grudin, J. (2001) Desituating Action: Digital Representation of Context. Human-Computer Interaction, Volume 16, pp. 269-286.

Hopson, J. (2001) Behavioral Game Design. Webresurs http://www.gamasutra.com/features/20010427/hopson_02.htm##

Hughes, J., King, V., Rodden, T. & Andersen, H. (1994) Moving out from the control room: Ethnography in system design. In proceedings of computer supported cooperative work CSCW, vol. 1:429-439 läst

Healy, J. & Picard, R. W. (1998). StartleCam: A cybernetic wearable camera. Proceedings of the 2nd international symposium on Wearable Computers (ISWC 98). Los Alamitos, CA: IEEE

Olsson, C. M. (2003). Taking the next step in context-aware applications. In Proceedings of IRIS26, Helsinki, Finland.

Pascoe, J., Ryan, N. S. & Morse, D. R. (1998). Human-computer-giraffe interaction: HCI in the field. Proceedings of the workshop on Human Computer Interaction with Mobile Devices. GIST Technical Report G98-1. Glasgow, Scotland: University of Glasgow.

Preece, J., Rogers, Y. & Sharp, H. (2002) Interaction Design. USA. John Wiley & Sons, Inc. s239-278

Retting, M. (1994) Prototyping for tiny fingers. Communications of the ACM, 37(4), 21-27.

Schmidt, A., Beigl, M., Gellersen, H.-W. (1998) There is more to context than location. Proc. of the Intl. Workshop on Interactive Applications of Mobile Computing (IMC98), Rostock, Germany.

Schmidt, A., Takaluoma, A. & Mäntyjärvi, J. (2000). Context-aware telephony over WAP. Personal Technologies, 4, 225-229

Schön, D. (1983) The reflective practitioner: How professionals think in action. New York: Basic Books

Weiser, M. (1991). The computer for the 21st century. Scientific American, September 1991, 94-104

Want, R., Hopper, A., Falcao, V. & Gibbons, J. (1992). The Active Badge location system. ACM Transactions on Information Systems, 10(1), 91-102



9 Appendix

9.1 Projektdeltagare

Lars Nyström från Chalmers Electro: Prototyputveckling.
Mark Lagerström från informatik: Prototyputveckling och denna uppsats.
Carl Magnus Olsson från Telematik gruppen:Handledning för prototypen.
Magnus Anderson från informatik. Hjälp med widgeten (en del av prototypen).
Håkan Johansson från Mecel: Hjälp med widgeten (en del av prototypen).
Ola Henfridsson från Telematikgruppen. Ansvarig för storprojektet.

9.2 Sensorer i SAAB 93

Område	Namn	Parametrar	Kommentar
Car specific	PowerMode	Key position (off, acc, run, crank)	Startnyckelns position
	NightPanel	On/Off	Instrumentbrädans belysning
	Transmission gear selector	Gear, valid	Växel
Navigation	GPS	Longitude, Latitude, Fix, Speed, Heading, Time	Innehåller flera värden, dessa kan kombineras för att få flera olika data som till exempel plats eller hastighet
	Gyro	Yaw rate	Känner av när bilen svänger
	Number of satellites	Antal	Hur många satelliter GPSen har kontakt med
MOST	Radio	Hz	Via Radion kan TMC data fås. TMC (Traffic Control Message) Sänds riktstäckande och innehåller trafikinformation.
	CD	Track number	
	Audio	Volume	

Other	Driver work-load	1,2,3	Ett värde på hur upptagen föraren är, baserat på hastighet, acc/deacc mm.
Basic	Odometer	Km	
	Vehicle speed	Km/h	
	Fuel tank	Liter	
Bonus	Fuel consumption	(momentan)	
	Engine RPM	Varv/minut	
Bilens direkta omgivningen	Temperatur	Grader Celcius	
	Hel- och halvljus	På/Av	
	Vindrutetorkare	På/Av	
Bilens interna status	Mil till nästa service	Mil	
	Varningslampor	På/Av	Ett flertal olika lampor, till exempel: spolarvätska, dörrar, blinkers, anti-spinn /sladd, säkerhetsbälten.