

Ansiktslyft för nallen

- kan användaren själv anpassa sitt gränssnitt till WAP-telefonen?

Anna Jacobsson & Camilla Törnquist
VT-2000

Abstract

The purpose of this thesis is to examine whether or not the user of a WAP-phone could create her own individual interface. We do the investigation on the basis of both the technology and the user. Therefore we divide the study into the two corresponding parts; development of a prototype with which the user could customize the interface and an evaluation of that prototype involving interviews. The development of the prototype was successful which proves that it is technically possible to make an adaptable application for a cellular phone. The evaluation of the prototype, as well, showed that it is possible to give the responsibility of the customization to the user.

Handledare:

Erica Wollerfjord, Linq Systems Svenska AB, Göteborg
Agneta Ranerup, Institutionen för Informatik, Göteborg

Förord

Äntligen! Ett fem månader långt examensarbete har nu nått sitt slut och vi vill framföra ett tack till alla dem, som på ett eller annat sätt, hjälpt oss att realisera den här magisteruppsatsen.

Först och främst vill vi tacka alla på det uppdragsgivande företaget **Linq Systems Svenska AB** för ett hjärtligt mottagande och en lärorik tid. Framförallt vill vi tacka **Erica Wollerfjord**, vår handledare från Linq som hela tiden har funnits tillgänglig för diskussion, inspirerat oss med nya idéer och bidragit med värdefull hjälp och handledning.

Vi vill dessutom tacka **de personer vi intervjuat** som genom att medverka har bidragit med väsentlig information till uppsatsen.

Vi vill till sist tacka **Agneta Ranerup**, vår handledare på institutionen för Informatik som bland annat lärt oss hur en vetenskaplig uppsats ska utformas.

Trevlig läsning!

Göteborg, maj 2000

Anna Jacobsson & Camilla Törnquist

1.	INLEDNING	5
1.1.	UPPSATSENS DISPOSTION	6
2.	BAKGRUND	7
2.1.	PRAKTISK RELEVANS.....	7
2.2.	TEORETISK RELEVANS.....	9
3.	TEORI.....	12
3.1.	MOBIL INFORMATIK	12
3.2.	ANPASSNING.....	14
3.2.1.	<i>Behov av anpassning.....</i>	<i>15</i>
3.2.2.	<i>Automatisk och manuell anpassning.....</i>	<i>17</i>
3.2.3.	<i>Föremål för anpassning.....</i>	<i>19</i>
3.2.4.	<i>Utnyttjandegrad av anpassningsmöjligheter</i>	<i>20</i>
3.3.	DESIGN AV GRÄNSSNITT.....	22
3.3.1.	<i>Traditionell design av gränssnitt</i>	<i>22</i>
3.3.2.	<i>Design av gränssnitt på mobila enheter</i>	<i>24</i>
3.4.	TRÅDLÖS KOMMUNIKATION MED HJÄLP AV WAP.....	25
3.4.1.	<i>GSM och WAP</i>	<i>25</i>
3.4.2.	<i>XML, DTD och XSL.....</i>	<i>26</i>
3.4.3.	<i>Exempel.....</i>	<i>27</i>
3.5.	SAMMANFATTNING AV TEORI	30
4.	METOD	31
4.1.	UTVECKLING AV PROTOTYP	32
4.1.1.	<i>Teknikstudier.....</i>	<i>32</i>
4.1.2.	<i>Scenarioplanering.....</i>	<i>32</i>
4.1.3.	<i>Analys, Design och Implementering</i>	<i>33</i>
4.2.	UTVÄRDERING AV PROTOTYP.....	35
4.2.1.	<i>Respondenter.....</i>	<i>35</i>
4.2.2.	<i>Utformning av intervjuer</i>	<i>35</i>
4.3.	SAMMANFATTNING AV METOD	36
5.	RESULTAT	37
5.1.	VÅR PROTOTYP.....	37
5.1.1.	<i>Tekniska begränsningar för utvecklingen av vår prototyp</i>	<i>37</i>
5.1.2.	<i>Scenarier.....</i>	<i>38</i>
5.1.3.	<i>Analys och design</i>	<i>41</i>
5.1.4.	<i>Implementering</i>	<i>43</i>
5.2.	INTERVJUSVAR	47
5.2.1.	<i>Respondenternas bakgrund.....</i>	<i>47</i>
5.2.2.	<i>Respondenternas uppfattningar kring Mobile Organizer.....</i>	<i>48</i>
5.2.3.	<i>Respondenternas åsikter kring vår prototyp.....</i>	<i>49</i>
5.3.	SAMMANFATTNING AV RESULTAT	51
6.	DISKUSSION.....	52
6.1.	REKOMMENDATIONER TILL FORTSATT ARBETE	54
6.2.	SLUTSATS	54
7.	REFERENSER.....	55
	BILAGOR	58

Figurlista

<i>Figur 1: Disposition</i>	6
<i>Figur 2: LinqPortal</i>	8
<i>Figur 3: Linq Mobile Organizer</i>	8
<i>Figur 4: Vår tolkning av interaktionsbegreppet</i>	10
<i>Figur 5: Bakgrund och teorier</i>	11
<i>Figur 6: Modell av IT-användning i mobila förhållanden</i>	13
<i>Figur 7: Reflektivitet och impulsivitet</i>	16
<i>Figur 8: Olika typer av anpassning</i>	17
<i>Figur 9: Beslutsprocess för utnyttjande av anpassningsmöjlighet</i>	21
<i>Figur 10: Förhållanden mellan begreppen SGML, XML, HTML och WML</i>	26
<i>Figur 11: Förhållanden mellan begreppen XML, XSL och DTD</i>	27
<i>Figur 12: Exempel på en DTD</i>	27
<i>Figur 13: Exempel på en XML</i>	28
<i>Figur 14: Exempel på en XSL</i>	29
<i>Figur 15: Resultat av körning av exemplet</i>	29
<i>Figur 16: Vår metod</i>	31
<i>Figur 17: Scenariernas oförutsägbara drivkrafter</i>	38
<i>Figur 18: Objektmodell av vår prototyp</i>	41
<i>Figur 19: Arkitektur över vår prototyp och dess omgivning</i>	42
<i>Figur 20: Gränssnittskomponenten</i>	43
<i>Figur 21: Skärmbild från vår prototyp: Meny</i>	44
<i>Figur 22: Skärmbild från vår prototyp: Meny, där val är gjorda av användaren</i>	44
<i>Figur 23: Skärmbild från vår prototyp: Inbox</i>	45
<i>Figur 24: Skärmbild från vår prototyp: Inbox, användaren har valt ett element som har barn. Därmed expanderar skärmbilden med ett nytt listpar</i>	45
<i>Figur 25: Skärmbild från vår prototyp: Calendar</i>	46
<i>Figur 26: Resultat efter anpassning med vår prototyp i mobiltelefonen</i>	46

1. Inledning

Mobiltelefonen har blivit var mans egendom på rekordkort tid, jämfört med hur lång tid det tog för alla svenska hushåll att skaffa sig fast telefon. Under samma period som användandet av mobiltelefoner explosivt ökat har också utvecklingen när det gäller Internet-teknik tagit ett stort steg framåt. Fram tills för ett par år sedan hade mobiltelefoner och Internet-teknik inte så mycket med varandra att göra medan de nu blir mer och mer integrerade. Den gemensamma faktorn är att den växande grupp människor som blir mer och mer beroende av kombinationen information och kommunikation har ett behov av att ständigt vara uppkopplade på nätet och samtidigt kunna resa och röra sig mer (Dahlbom & Ljungberg, 1999). Ur det här behovet har det vuxit fram en ny mobil livsstil.

En teknik som stöder den här formen av kombinerad information och kommunikation är Wireless Application Protocol (WAP) (Wapforum, 2000). WAP är ett protokoll som är utvecklat av stora telekommunikationsföretag som Ericsson, Nokia, Motorola och Phone.com för att möjliggöra internetbaserad datakommunikation över det mobila telenätet på ett enkelt och enhetligt sätt.

När människor använder ovanstående tekniker växer behovet av att skapa en miljö som är behaglig för användaren, vilket den inte alltid är idag. Det tror vi kan bero på att tekniken ännu är mycket begränsad. Ett exempel på en sådan begränsning är att ytan för informationspresentation på en mobiltelefon inte är större än några få kvadratcentimeter, vi pratar inte ens tum längre!

Situationerna då man använder en mobiltelefon ser också helt annorlunda ut mot de situationer då man sitter framför en stationär dator. Användaren kan inte fästa större delen av sin uppmärksamhet på mobiltelefonen när hon samtidigt rör sig i trafiken eller pratar med andra människor.

Utifrån ovanstående resonemang tycker vi att det skulle vara spännande att se hur man med dagens teknik skulle kunna förbättra omständigheterna för användaren. För oss som systemvetare är det en självklarhet att se till användarens bästa. Mycket av den mentaliteten har vi tagit till oss från vår professor, Bo Dahlbom vid institutionen för Informatik i Göteborg. Vår utgångspunkt för att tillfredsställa användaren är **anpassning**. Är det tekniskt möjligt att anpassa gränssnittet i en mobil applikation? Skulle en van användare av en WAP-applikation ha någon nytta av att kunna anpassa sitt gränssnitt så att det passar just hennes behov? Dessa båda delfrågor leder till magisteruppsatsens vetenskapliga fråga: "**Kan användaren själv anpassa sitt gränssnitt till WAP-telefonen?**"

Syftet med vår studie är att besvara ovanstående fråga utifrån två synvinklar, dels utifrån tekniken och dels utifrån användaren.

För att utreda den tekniska aspekten kommer vi att utveckla en prototyp till ett verktyg för en PC. Tanken är att användaren med hjälp av prototypen ska kunna anpassa sitt gränssnitt till WAP-telefonen. Resultatet av den här delen av utredningen kommer att visa om det är tekniskt möjligt att skapa en sådan prototyp med hjälp av befintlig teknik och vår kompetens.

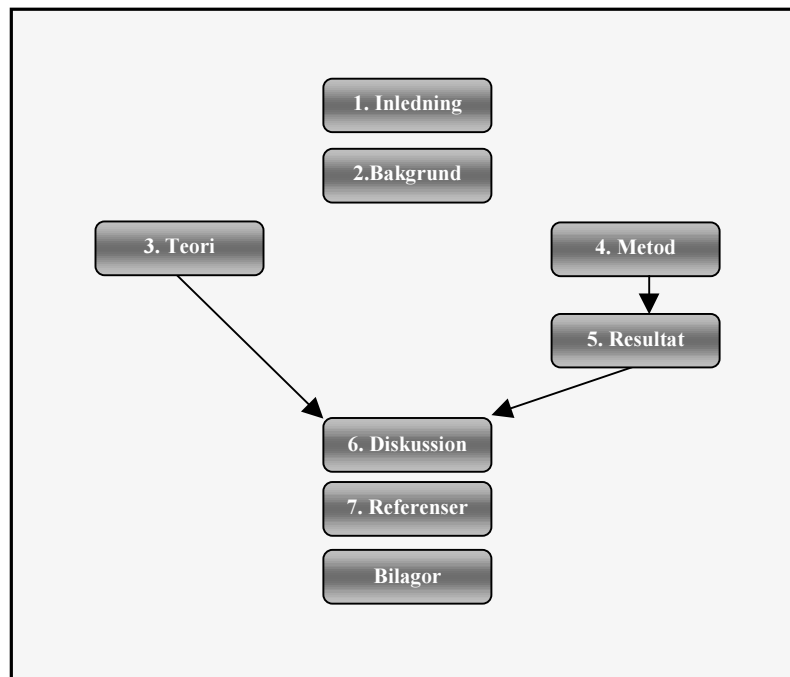
För att utreda den aspekt som innefattar användaren kommer vi att utföra ett antal djupintervjuer med vår prototyp som grund. Vi kommer att grunda intervjufrågorna i teorier kring anpassning och användargränssnittsdesign som vi presenterar i uppsatsen. Genom att jämföra svaren på frågorna med de befintliga teorierna vill vi komma fram till huruvida det är lämpligt att låta användaren sköta sina egna anpassningar i en mobil applikation.

Den här uppsatsen är skriven i samarbete med Linq Systems Svenska AB, i uppsatsen kallat Linq, som är ett produktorienterat konsultföretag som utvecklar företagsportaler.

1.1. Uppsatsens disposition

- I **Inledningen** ger vi en allmän beskrivning av uppsatsen. Vi redogör där för frågeställning och syfte.
- I **Bakgrunden** behandlar vi uppsatsens praktiska och teoretiska relevans.
- I **Teoridelen** presenterar vi begreppen mobil informatik, anpassning och design av gränssnitt och de teorier kring dessa begrepp som uppsatsen bygger på. Dessutom presenterar vi fakta kring ämnet trådlös kommunikation med hjälp av WAP.
- I **Metoddelen** ger vi en beskrivning av hur vi angriper frågeställningen. Vår metod grundar vi i vedertagna metoder kring scenarioplanering, objektorienterad systemutveckling och intervjuteknik.
- I **Resultatet** redogör vi för de två olika grenarna som resultatet består av, dels vår prototyp och dels utvärderingen av prototypen.
- I **Diskussionen** knyter vi ihop hela studien, vi diskuterar kring resultatet med utgångspunkt i de teorier vi presenterat i teoridelen. Här finns också rekommendationer inför fortsatt arbete. Sist presenterar vi vår slutsats i vilken vi sammanfattar resultatet utifrån den diskussion vi fört.
- Samtliga referenser presenterar vi under **Referenser**, de är uppdelade i de olika kategorierna: böcker, artiklar, internetsidor och övrigt.
- **Bilagorna** är bilaga A; Analys och design och bilaga B; Intervjufrågor

Uppsatsens kapitel och hur de bygger på varandra åskådliggörs i figur 1 nedan.



Figur 1: Disposition

2. Bakgrund

Uppgiften som ligger till grund för den här studien har vi arbetat fram tillsammans med Linq. Vi kom överens om att det skulle vara intressant för både oss och för företaget att titta på möjligheterna till anpassning på en av deras mobila applikationer.

Under *praktisk relevans* behandlar vi praktiska grunder till vår uppsats hämtade från Linq. Vi beskriver kortfattat företagets produkter och försöker klargöra var Linq befinner sig idag och vad de har för intresse i vår studie. Det känns viktigt att beskriva Linqs betydelse för oss då deras stöd självklart kommer att påverka oss i vårt arbete.

Teoretisk relevans ger utrymme för diskussion kring begrepp, teorier och uppfattningar som vi under fyra år på det Systemvetenskapliga programmet i Göteborg har samlat på oss. Vi kommer att behandla begreppen informatik och människa-datorinteraktion som båda är bakgrund till vår studie. Dessutom kommer vi att återge vår tolkning av interaktionsbegreppet och utifrån detta och andra väsentliga begrepp avgränsa det område uppsatsen behandlar.

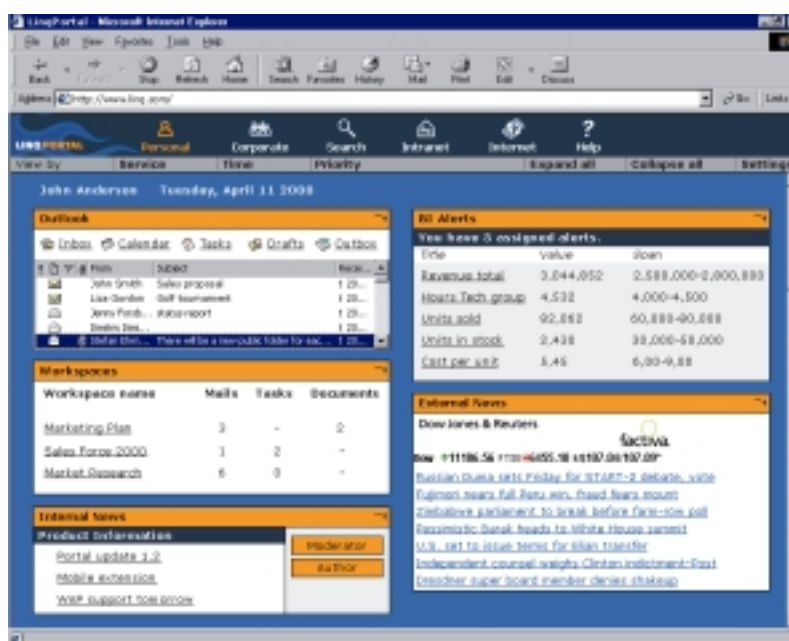
2.1. Praktisk relevans

Linq är ett växande internationellt konsultföretag med drygt 200 anställda som huvudsakligen utvecklar och anpassar företagsportaler (även kallat portal). Den grundläggande idén med en portal är att den ska ge medarbetare i stora företag och organisationer tillgång till information och verktyg som de behöver i sitt dagliga arbete. Linqs huvudsakliga produkt heter *Linq Portal* och innan vi beskriver den mer detaljerat så återger vi vad som är grundläggande för en företagsportal.

Ordet portal står för ingång eller inkörspport, och särskilt stora och pampiga sådana (fritt översatt från www.dictionay.com). Enkelt kan man beskriva en företagsportal som nästa generations intranät. Fördelen med en portal är att all information och arbetsverktyg samlas på en enda plats istället för att finnas utspridda i ett intranät, i olika databaser eller flera olika system. Det kan finnas möjlighet till individualisering av portalen vilket ger användaren hjälp med att sortera informationsbruset och fokusera på det som är viktigt för just henne. För företaget är det ett sätt att hantera sina kärnresurser som t.ex. kan vara de anställdas kunskap och informationsflödet i organisationen. Enligt *Financial Times* beräknas företagsportaler omsätta 125 miljarder kronor redan år 2002. Den amerikanska marknaden för företagsportaler ligger ännu långt före de svenska och europeiska marknaderna, men de senare är på stark frammarsch (*Financial Times*, 2000).

LinqPortal är en företagsportalfamilj och den huvudsakliga idén med LinqPortal är att den samlar och strukturerar informationsmassan som medarbetare i ett företag varje dag översvämmas av. LinqPortal bygger på vanliga produkter från Microsoft Office och Microsoft Backoffice, vilket innebär att företag kan öka värdet på en investering som redan är gjord. Detta tillåter användarna att fortsätta arbeta i en miljö de är vana vid. Det finns fyra olika varianter av LinqPortal: *Basic Edition*, *Corporate Edition*, *Public Edition* och *Business Intelligence Edition*.

Basic Edition är en version som innehåller basfunktioner som behövs för att skapa en portal. De andra tre versionerna innehåller i sin tur färdiga funktioner som är riktade till speciella målgrupper. Corporate Edition riktar sig mot privata företag, Public Edition riktar sig mot offentliga organisationer och Business Edition riktar sig till beslutsfattare inom företag. I figur 2 ser vi ett exempel på hur LinqPortal kan se ut.



Figur 2: LinqPortal

Under sommaren 1999 bedrevs ett sommarprojekt på Linq i samarbete med Viktoriainstitutet och projektet resulterade i utvecklingen av **Linq Mobile Organizer**, i uppsatsen kallat Mobile Organizer. Det är ett klient/server-baserat system som gör funktionaliteten i Microsoft Outlook¹ tillgänglig även för företagets mobila medarbetare. Systemet bygger på WAP där mobiltelefoner fungerar som klienter.

Mobile Organizer ger användaren tillgång till e-post, kalender, kontakter, att göra-lista och all annan information som finns tillgänglig i Microsoft Outlook - direkt i mobiltelefonen. Mobile Organizer utvecklades som ett steg på vägen mot den portal-klient som ska ge de mobila företagsanvändarna tillgång till portaltillämpningarna. Den kommande produkten heter Mobile Extension. Mobile Organizer är precis som portalen baserad på standardkomponenter från Microsoft. Produkten kan användas tillsammans med alla enheter som stöder WAP, t.ex. mobiltelefoner och handdatorer.

Kunskapen och informationen kring LinqPortal och Mobile Organizer har vi inhämtat genom informella samtal med personer på Linq och genom kongresser med Linq².

Beskrivningen av begreppet portal är viktigt för att förstå innebörden i den mobila applikationen Mobile Organizer vilken vi har använt som utgångspunkt i vår studie. Linq har ett intresse i vår studie som har lett oss till att klargöra dels vilka tekniska möjligheter det finns till att göra en mobil applikation anpassningsbar och dels om dylika anpassningsfunktioner skulle användas.

¹ För beskrivning av Microsofts produkt Microsoft Outlook se: <http://www.microsoft.com/office/outlook>

² Kongress i Stockholm, 10/12-99, kongress i Stockholm, 17-18/2-00 samt kongress i Lissabon, 5-9/4-00 vid vilka vi deltog

2.2. Teoretisk relevans

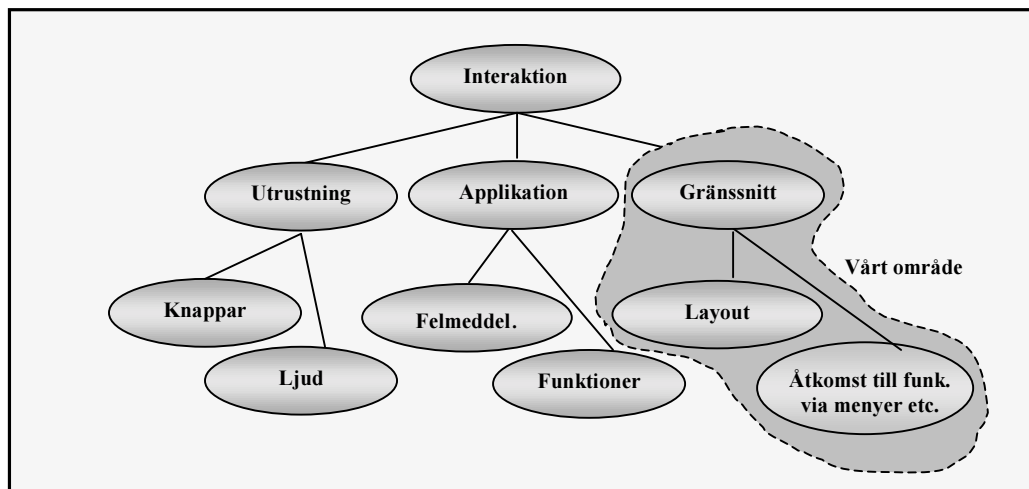
Bo Dahlbom, professor vid institutionen för Informatik, f.d. forskningsledare vid Viktoriainstitutet samt VD för Svenska IT-institutet har under våra fyra år som studenter på det systemvetenskapliga programmet i Göteborg påverkat oss med sina åsikter. I artikeln "Informatics in the next millenium" diskuterar han begreppet **Informatik** som idag namnger den disciplin i Sverige som tidigare hette "administrativ data behandling" (Dahlbom, 1999). Namnbytet indikerar en radikal förändring som bland annat innebär uppkomsten av en rad olika subdiscipliner i Göteborg, t.ex. mobil informatik, interactive learning och knowledge management. Dahlbom spekulerar i samma artikel kring ett namn på disciplinen som skulle kunna förena och stärka de olika subdisciplinerna och tvivlar på att Informatik kan spela den rollen. Om det hade funnits ett namn för "användning av informationsteknologi" hade det föreslagits av Dahlbom, vilket även visar på informatikforskningens fokus i Göteborg som ligger just på *användningen* av informationsteknologi (IT). Användaren ska sättas i centrum och tekniken ska därefter utvecklas och anpassas för att tillgodose användarens behov.

Utvecklingen och bakgrunden till det som vi idag kallar informatik startade redan under andra världskriget. Det var då de första elektroniska datorerna kom till världen (Dahlbom, 1997). Deras huvudsakliga mål var att utföra beräkningar, främst för militären. Under 60-talet kom datorerna att användas som informationssystem. Man insåg datorernas kapacitet när det gällde att registrera och hålla ordning på data. Ambitionen var att automatisera det administrativa arbetet. Nästa steg i utvecklingen kom med introduktionen av PC:n under 80-talet. Med denna kom det grafiska gränssnittet och möjligheten för privatpersoner att äga en dator. I slutet av 80-talet var det nätverk och klient-server teknologi som ändrade fokus för användningen av IT. Det stora genombrottet för IT kom dock med Internet, enligt Dahlbom. IT blev ett verktyg för kommunikation. Ovanstående är en kort introduktion till var vi befinner oss idag, men vart är vi då på väg?

Informationsteknologi har kommit att bli ett medel för att realisera individuella, sociala och organisatoriska mål (Magoulas & Pessi, 1999). Idag frågar vi oss inte om vi ska använda informationsteknologi utan vi frågar snarare hur vi ska använda den. Vi lever i en tid som präglas av "den nya tidens informatik" där viktiga begrepp är nätverkstänkande, IT-användning, kommunikation, knowledge management och prat/tjänste-samhälle. Men det som är mest framträdande i nuläget är enligt Magoulas och Pessi diskussionen kring mobilitet. Den tid då vi hade stora, hierarkiska och stabila organisationer, inom vilka människor arbetade länge, har börjat försvinna. Idag präglas vårt samhälle istället av människor som rör på sig allt mer, både inom och mellan arbeten. Organisationer som bygger på dynamik och nätverk börjar ta form och de skapas för att anpassas till den alltmer föränderliga och turbulenta miljö som växer fram kring organisationerna.

Människa-datorinteraktion (MDI) är den svenska översättningen av Human Computer Interaction (HCI) som är ett forskningsområde vilket fått allt större uppmärksamhet under de senaste 25 åren (Preece, 1994). Från början var ämnet främst förknippat med forskning, men idag är det en betydelsefull del av systemutvecklingsprocessen. En av anledningarna till denna utveckling är att den ökande datorkraften i kombination med de sjunkande priserna har skapat en ny användarkrets som kräver ett enklare bemästrande av datorer. Precis som i bilens och kamerans barndom har det även när det gäller datorerna krävts stor ansträngning från användarens sida för att kunna utnyttja kraften i den nya tekniken (Shneiderman, 1998). Det finns många olika typer av MDI. Några vitt skilda teknikområden vars applikationer kan byggas efter samma grundläggande riktlinjer inom MDI är: PC, handdatorer, maskiner i industrin och WAP-telefoner. Den här uppsatsen behandlar gränssnittsdelen av interaktionen mellan

människa och WAP-telefon. Vi har tolkat termen interaktion som ett paraplybegrepp, där de olika beståndsdelarna är utrustning, applikation och gränssnitt, se figur 4.



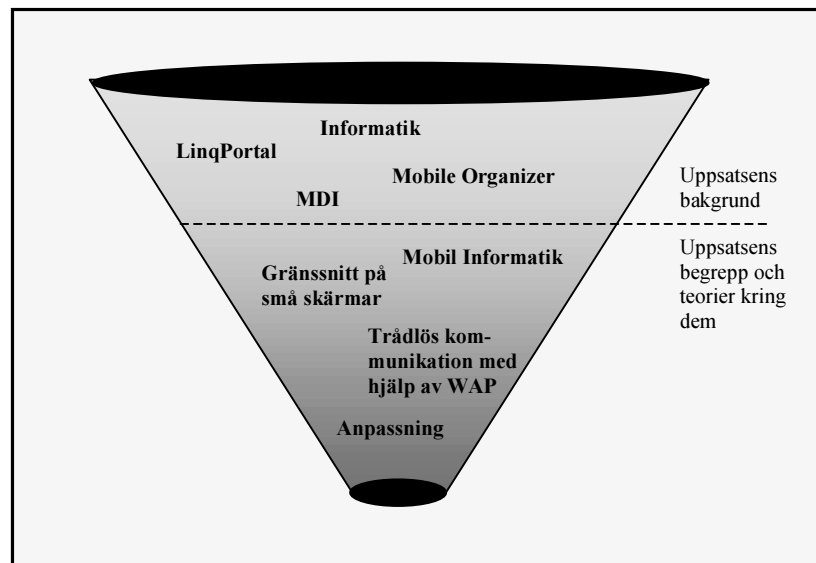
Figur 4: Vår tolkning av interaktionsbegreppet

Med **utrustning** syftar vi på de möjligheter och begränsningar som finns i den teknik som ska erbjuda interaktion med en människa. Exempel på sådana faktorer när det gäller WAP-telefonen är vilka knappar som finns och hur de är placerade. Ett annat exempel är om telefonen har någon funktion för att generera olika ljud, i så fall skulle man kunna bygga in stöd i applikationen som utnyttjar den möjligheten så att telefonen låter på ett speciellt sätt när det kommer ett e-post från en viss person.

Med **applikation** menar vi hur applikationens uppbyggnad och dess funktioner påverkar interaktionen med en människa. För att poängtera att det är viktigt att den applikation som ligger i botten genererar ”bra” data kan man ta exemplet när en applikation skickar felmeddelande nummer 348 till användaren då hon matat in en felaktig e-post-adress istället för att meddela att det inte är en korrekt adress. Det spelar ingen roll hur bra gränssnittet till en sådan applikation är, användaren får i alla fall inget vettigt meddelande som upplyser henne om vad som är fel om applikationen saknar möjlighet att generera den data som behövs.

Med **gränssnitt** vill vi peka på det sätt som applikationen och dess data presenteras på utrustningen, t.ex. på en WAP-telefon, och på hur det påverkar interaktionen med människan. Exempel på det är i vilken ordning data presenteras på skärmen och om data står huller om buller eller i snygga tabeller. I gränssnittet innefattas också hur man kommer åt funktionerna i applikationen, t.ex. genom menyer eller ikoner.

Bakgrunden har behandlat förutsättningar för vår uppsats och leder oss till de begrepp och teorier som är lite mer specifika för vår studie. I figur 5 placerar vi in viktiga begrepp under bakgrund respektive begrepp och teorier.



Figur 5: Bakgrund och teorier

Vi har ritat en tratt i figur 5 för att åskådliggöra vägen fram till våra två huvudbegrepp; anpassning och design av användargränssnitt på små skärmar. De begrepp vi anser vara grundläggande för vårt sätt att tänka och uppfatta världen finns beskrivna här i bakgrunden medan mobil informatik som är länken från det mer generella till våra specifika områden finns beskrivet i kapitel 3 Teori liksom de två huvudbegreppen som nämns ovan.

3. Teori

I uppsatsens teoridel behandlar vi begreppen mobil informatik, anpassningsbarhet, design av gränssnitt och trådlös kommunikation med hjälp av WAP.

Mobil informatik är ett begrepp som leder oss från vår mer generella bakgrund till de övriga begreppen och teorierna som kommer att presenteras. **Anpassning** är ett vitt begrepp som kan ses utifrån många olika aspekter, vi har valt att presentera ett urval av dessa som vi anser relevanta för vår studie. Hur man skapar en bra **design av ett gränssnitt** är naturligtvis också av stor vikt och särskilt då hur man designar gränssnitt på små ytor, i det här underkapitlet presenterar vi grundläggande designprinciper. De mer tekniska begreppen kring **trådlös kommunikation med hjälp av WAP** har vi samlat i det sista underkapitlet.

3.1. Mobil informatik

IT berör alla dimensioner av vårt samhälle, vilket har lett till att ämnet Informatik med sin generella framtoning delats in i en rad subdiscipliner som vi nämnde i kapitel 2 Bakgrund. En av dessa är just mobil informatik som undersöker tjänster och koncept kring mobil IT-användning (Dahlbom & Ljungberg, 1999).

När energi läggs på att skapa datorkraft i mobila enheter skapas en helt ny marknad för informationshantering (Wapforum, 1999). Tillväxten av den trådlösa marknaden har varit fenomenal. Idag finns mer än 200 miljoner användare av mobiltelefoner och denna siffra spås vara en miljard år 2005 (Wapforum, 1999). Marknaden skiljer sig dock avsevärt från den traditionella skrivbordsdatormarknaden och även marknaden för bärbara datorer eftersom det nu vuxit fram nya behov och förväntningar. En mobil enhet bör vara mycket enklare att använda än den enklaste skrivbordsdatorn då användaren oftast har sin huvuduppmärksamhet på någon annan aktivitet än själva användandet av den mobila enheten (Dahlbom & Ljungberg, 1999). Man ska till exempel inte behöva läsa i en manual för att komma på hur det nu var man skickade ett e-post-brev från telefonen och man ska inte ha några svårigheter med att snabbt avsluta det man håller på med om något viktigt skulle inträffa i omgivningen. En applikation kan upplevas som enkel för användaren om gränssnittet är bra och erbjuder snabb och enkel användning av applikationen (Dahlbom & Ljungberg, 1999).

Uppgifterna som användare vill ha utförda efter det att de lämnat sitt skrivbord grundar sig i andra informationsbehov och är av mer grundläggande natur. Användaren vill kanske överblicka sin e-post snarare än att läsa varje specifikt brev eller kanske vill hon snabbt bli uppdaterad om aktuella aktiekurser eller trafikförhållanden. Fokus på IT-användning skiftar således från att vara rent administrativt till att innefatta Internet och mobiltelefoner, vilket medför att arbetsuppgifterna skiftar från att ha fokuserat på pappersarbete till att hantera kundrelationer och från dokumenthantering till försäljning och hantering av tjänster (Dahlbom & Ljungberg, 1999). Genom att titta på mobilitet i sammanhanget arbete så kan man hitta en anledning till varför människor blivit mer och mer mobila. Ser man på moderna organisationer bygger mycket av arbetet på samarbete, människor arbetar i projekt och team. Samarbete leder till ökad användning av IT för att övervinna stora avstånd, t ex med hjälp av e-post, men det leder även till ökad mobilitet för att människor ska kunna mötas fysiskt.

En annan faktor som påverkat mobilitet är den påtagliga trend som inneburit en minskning av tillverkning och en ökning i utförande av tjänster. Utför man tjänster befinner man sig ofta där kunden är medan tillverkning sker i fabriken (Dahlbom & Ljungberg, 1999).

Även själva mobiltelefonerna har lett till ökad mobilitet. Människor är nåbara trots att de inte befinner sig på kontoret (Dahlbom & Ljungberg, 1999).

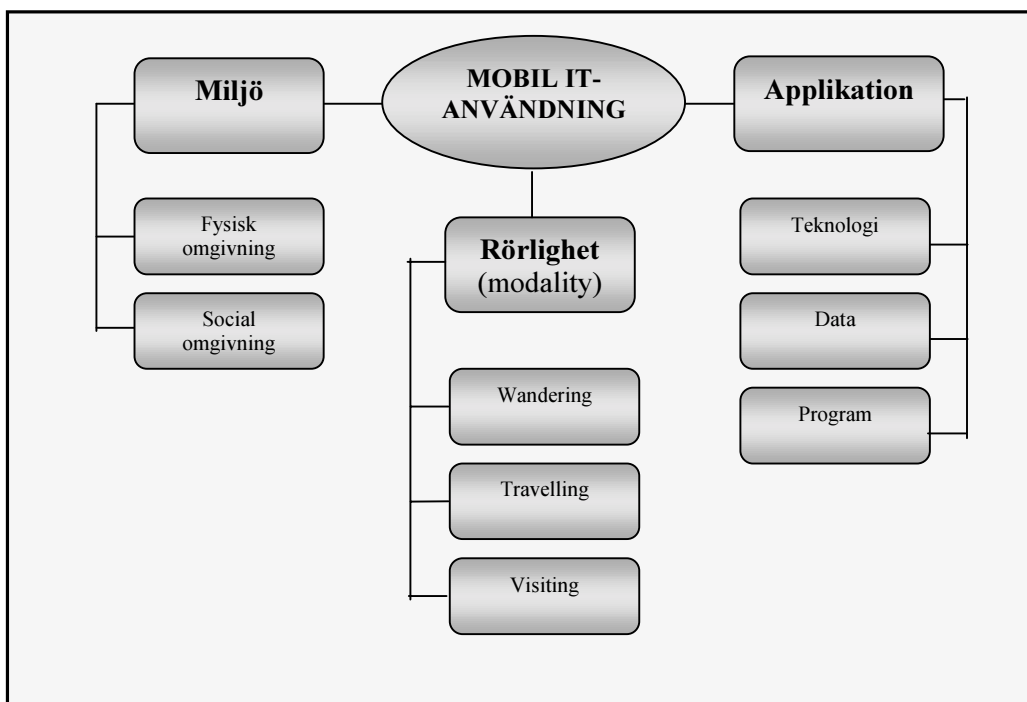
En människa kan vara mobil på mer än ett sätt. Dahlbom och Ljungberg återger tre arketyper (Dahlbom & Ljungberg, 1999):

- 'Wandering' – en aktivitet som karaktäriseras av omfattande lokal mobilitet, t.ex. att röra sig mycket och ofta inom ett stort kontor
- 'Travelling' – en aktivitet som sker när man förflyttar sig med hjälp av ett fordon, t.ex. åker buss eller bil
- 'Visiting' – en aktivitet som äger rum på en plats en viss tid, t.ex. kundbesök

Kristoffersten och Ljungberg återger i en rapport att IT-användningen skiljer sig i mobila gentemot stationära förhållanden (Kristoffersten & Ljungberg, 1998). Därför är de starka influenserna från utvecklingen av stationära applikationer till applikationer på mobila enheter oturlig. Författarna återger ett förslag på en modell av IT-användning i mobila förhållanden som bygger på empiriska studier av mobilt arbete. Modellen presenteras i figur 6. Begreppen miljö och rörlighet används för att förklara det specifika i mobila förhållanden. Modellens syfte är att ge designers ett ramverk av koncept för att kunna förstå och prata om hur människor använder IT i mobila förhållanden. Författarna påpekar att modellen är en designmodell och inte en teori om mänskligt handlande.

Eftersom ett viktigt mål är att hjälpa designers att uppfinna nya koncept och föreställa sig nya metaforer för mobila förhållanden reflekterar modellen hur IT-användning i mobila förhållanden skiljer sig från stationära förhållanden. Modellens startpunkt är Mobil IT-användning. Huvudkomponenterna i Mobil IT-användning är miljö, rörlighet och applikation.

- Miljö – den fysiska och den sociala omgivningen
- Rörlighet – fundamentala mönster av rörelse i en aktivitet.
- Applikation – kombinationen av teknologi, program och data som används.



Figur 6: Modell av IT-användning i mobila förhållanden

De grundläggande riktlinjer som ges inom den mobila informatiken ligger som ett stöd bakom våra resterande teorier kring begreppen anpassning och design av gränssnitt. Att mobila enheter måste vara enklare att använda än den enklaste skrivbordsdatorn styrker vår idé om anpassning från användaren. Det blir enklare att använda en mobil enhet vars applikation man själv har personifierat efter egna arbetsuppgifter och önskemål.

Ser man på modellen som visas i figur 6 på sidan 13 finns det en rad faktorer som spelar in vid mobil IT-användning. Dessa är som nämnts miljö, rörlighet och applikation med sina respektive beståndsdelar. För vår studie är det väsentligt att fundera kring dessa faktorer i sammanhanget anpassning. Nedan kopplar vi begreppen miljö och rörlighet till anpassning.

Man skulle kunna tänka sig att **miljön** som består av den fysiska och sociala miljön skulle kunna påverka användaren annorlunda om applikationen var anpassningsbar. Ett exempel är att du har en mobil applikation till din WAP-telefon som du använder ofta men där du bara är intresserad av en specifik funktion. Det är många steg du måste genomföra för att komma till funktionen och det är krångligt att navigera sig fram, speciellt då du bara kan använda en hand eftersom du står på bussen och dessutom försöker prata med kompiserna bredvid samtidigt. Då kan man föreställa sig att en anpassning av applikationen där det är möjligt att själv välja vilken startside man vill ha skulle vara lämplig. Det första du möts av när du går in i applikationen är således den sidan där din huvudfunktion finns. Den *fysiska miljön* som innebär att du står på en buss och bara kan använda en hand och den *sociala omgivningen* som innebär att du samspråkar med din kompis blir plötsligt ett mindre bekymmer.

Även **rörlighet** som nämns i modellen i figur 6 skulle kunna påverka IT-användningen annorlunda med integration av anpassning. Här kan man tänka sig att applikationen anpassas efter på vilket sätt man är mobil. För att exemplifiera kan man föreställa sig en applikation för WAP-telefonen som innehåller en e-post-funktion där du kan läsa och skriva e-post, men även genom en signal bli uppdaterad om att du har ny e-post. Det skulle troligen inte vara önskvärt att det började pipa när du sitter på ett kundbesök, ”*visiting*” men du tar gärna emot en signal om ny e-post när du är ute och vandrar på kontoret, ”*wandering*”. Man skulle även kunna föreställa sig att när du är ute och reser på tåget, ”*travelling*”, och blir signalerad om ny e-post är du endast intresserad av se nya meddelanden. Du har således gjort en anpassning av applikationen som säger att du bara vill se oläst e-post i din telefon.

3.2. Anpassning

I det här underkapitlet kommer vi att gå in djupare i teorier kring anpassning. Vi kommer att behandla behov av anpassning där vi påvisar att behov växer fram ur individuella skillnader och varierande arbetsuppgifter. Därefter kommer vi att presentera två olika typer av anpassning där vi utifrån befintliga grupperingar ur litteraturen delar in anpassning i två typer; manuell och automatisk. Vidare diskuteras föremål för anpassning där föremålen utgörs av funktionalitet och användargränssnitt. Sist behandlas utnyttjandegraden av anpassningsmöjligheter. Men innan vi behandlar dessa aspekter ger vi en liten introduktion till anpassning.

I engelskan finns det ett flertal ord som används för att beteckna den process genom vilken en användare själv anpassar sin applikation efter sina egna behov. Exempel på dessa ord är *adaption*, *customization*, *personalization* och *individualization*. Det finns en viss nyansskillnad i betydelsen av orden men det ord vars mening passar oss bäst är *customization*, vilket översatt till svenska betyder ”anpassning till individuella eller personliga specifikationer” (Dictionary.com, 2000). Målet med anpassning till individuella eller personliga specifikationer är givetvis att applikationen ska bli mer användbar (Oppermann, 1994). Begreppet användbarhet omnämns med många olika definitioner i forskningslitteraturen. Allwood ser användbarhet som något iterativt, där olika egenskaper i användningssituationen bestämmer användbarheten (Allwood, 1998). Fyra

faktorer som Allwood anser påverka användbarhet är anpassning, användarvänlighet, användaracceptans och användarkompetens.

Anpassning i det här sammanhanget betyder att applikationen överhuvudtaget stöder de arbetsuppgifter som en användare har. Allwood har således en lite snävare betydelse av anpassning i det här sammanhanget än vi har i den övriga uppsatsen.

Användarvänlighet innefattar ett antal olika aspekter. En viktig aspekt är att applikationen är åtkomlig. Om det inte finns tillgång till programmet går det inte att använda. En annan elementär aspekt är att applikationen måste vara förenlig med hur användaren fungerar mentalt. Information som en användare behöver hålla i huvudet vid ett interaktionstillfälle får inte överskrida den mängd information som en användare klarar av. Även Shneiderman har en princip som säger att man inte ska överbelasta korttidsminnet (Shneiderman, 1998). Allwood talar vidare om individualisering som kan jämföras med anpassningsbarhet (Allwood, 1998). Eftersom användare är olika är det större chans att de interagerar med applikationen på ett tillfredsställande sätt om individualisering tillåts. Det är dock viktigt att individualiseringen i någon mån är kontrollerad så att användaren inte tillåts att skapa en kaotisk miljö.

Användaracceptans innebär att användaren är välvilligt inställda till och har hög motivation att använda programmet. Om motivation saknas är det lätt hänt att programmet inte används eller att det används felaktigt.

Att användaren har tillräckligt med förståelse och tillräckliga färdigheter för att kunna samspela med datorn på ett effektivt sätt innefattas i begreppet **användarkompetens**. För att uppnå detta krävs en effektiv utbildning på systemet, underförstått att användaren har kompetens kring de arbetsuppgifter som hon vill lösa med hjälp av systemet.

När Allwood talar om att skapa en användbar applikation nämner han anpassning. Han kallar det dock för individualisering, men det är ett tecken på att anpassning kan hjälpa oss att uppnå användbarhet vilket Oppermann anser vara målet med anpassning (Oppermann, 1994).

3.2.1. Behov av anpassning

Vid utvecklandet av i princip alla applikationer kännetecknas målgruppen av en heterogen grupp av människor. De har olika förkunskaper, individuella egenskaper och de är kanske intresserade av att lösa olika arbetsuppgifter med hjälp av applikationen i fråga. Eftersom det inte är realiserbart att utveckla en applikation för varje individ kan en lösning på problemet vara anpassningsbara gränssnitt. Det ger användaren en möjlighet att anpassa applikationen efter egna behov och önskemål.

Benyon och Murray återger i en artikel två huvudsakliga skäl till att göra en applikation anpassningsbar (Benyon & Murray, 1993):

- Det finns stora individuella skillnader mellan användare
- De uppgifter användarna vill lösa med hjälp av applikationen varierar

Mellan datoranvändare finns det stora **individuella skillnader**. Individuella egenskaper som diskuteras av Allwood är analogiskt tänkande, kunskapsnivå, kognitiv stil och motivation (Allwood, 1991).

Analogiskt tänkande innebär att tänkandet till stor del är beroende av individens förkunskaper. I sitt tänkande utgår individen från vad han eller hon redan vet och kan och det är vanligt att individen utnyttjar kunskaper från ett område inom ett annat. Detta kan ske genom att individen ser ett objekt i termer av ett annat, tex. kan en dator betraktas som en skrivmaskin.

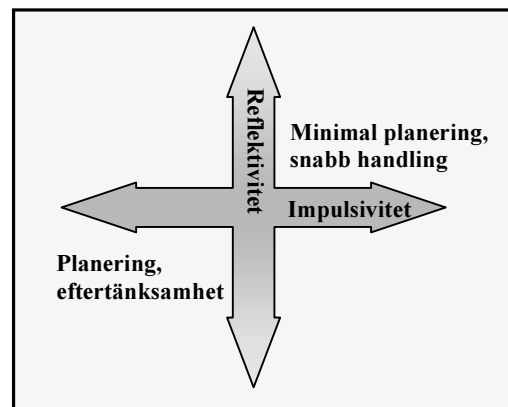
Eftersom analogiskt tänkande beror på individens förkunskaper så skiljer sig användare åt med avseende på vilka effekterna blir av deras analogiska tänkande. Allwood återger att man rimligen kan tänka sig att en nybörjare skulle kunna få negativa effekter av analogiskt tänkande genom att

utnyttja ursprungsdomäner som ligger utanför datorvärlden. Experter i sin tur skulle kunna få negativa effekter genom för mycket kunskap om andra program. Ett exempel är om man arbetar med en ny version av en programvara som avsevärt skiljer sig från den gamla, men ändå hela tiden gör jämförelser mellan de båda.

Kunskapsnivån kring datordomänen har stor betydelse eftersom det finns väsentliga skillnader hos nybörjare och experter. En nybörjares förståelse för datordomänen kännetecknas i allmänhet av att vara ofullständig, dåligt integrerad, delvis felaktig och diffus. Däremot kan deras förståelse för arbetsuppgiften vara mycket bra. Variationen av förståelsen hos en grupp nybörjare kan vara ganska stor medan en grupp experter ofta har likvärdig förståelse. Det har visat sig att experter har en större förmåga att se helheten i en uppgift medan en nybörjare riskerar att drunkna i en massa detaljer.

Kognitiv stil handlar om allmänna tendenser i en individs tänkande. Individer har visat sig vara olika vad gäller dimensionen impulsivitet och reflektivitet. Det leder till att vissa individer är planerande och eftertänksamma och därmed ogärna vill råka ut för effekter av en datorinteraktion innan de tror sig veta vad som kommer att hända. Andra individer kan i sin tur handla snabbt med minimal planering och är därmed villiga att omedelbart ta konsekvenserna av sitt handlande. Se figur 7.

Motivationen varierar av olika anledningar hos olika individer. Det kan bero på skillnader i tidigare erfarenheter och aktuell livssituation. Motivation kan om den är för hög eller för låg påverka en prestation negativt. I lagom dos kan dock motivation underlätta för en individ att lära sig något datorrelaterat då motivationen påverkar användarens uppmärksamhet, villighet att försöka själv, villighet att försöka igen efter misslyckande och ansträngning att försöka ta sig ur problemsituationer.



Figur 7: Reflektivitet och impulsivitet

Den andra punkten som Benyon och Murray uppger vara ett huvudsakligt skäl till att göra applikationer anpassningsbara är att det finns en stor **variation av uppgifterna** en användare vill lösa med hjälp av en applikation. Ett datasystems främsta syfte är att hjälpa människor med deras arbetsuppgifter. Oftast utvecklas ett system för att ge stöd åt en rad olika uppgifter. Hur en användare sedan använder applikationen för att utföra sina arbetsuppgifter är individuellt och föränderligt (Oppermann, 1994). Användningsmönstren ser således olika ut för olika användare vid olika tidpunkter.

Utifrån ovanstående resonemang kring att ingen användare är den andra lik och att de använder samma applikation till att lösa olika arbetsuppgifter kan man dra en del slutsatser kring anpassning. Det verkar rimligt att anta att möjlighet till anpassning skulle öka sannolikheten för överensstämmelse mellan applikationen och användaren eftersom individuella behov skulle kunna tillfredsställas bättre. Dessutom verkar det finnas ett behov av att anpassa applikationen med avseende på vilken arbetsuppgift som ska lösas. Den enda som kan anpassa applikationen efter detta behov är användaren eftersom hon är den enda som vet vilken arbetsuppgift hon vill lösa vid varje specifikt tillfälle.

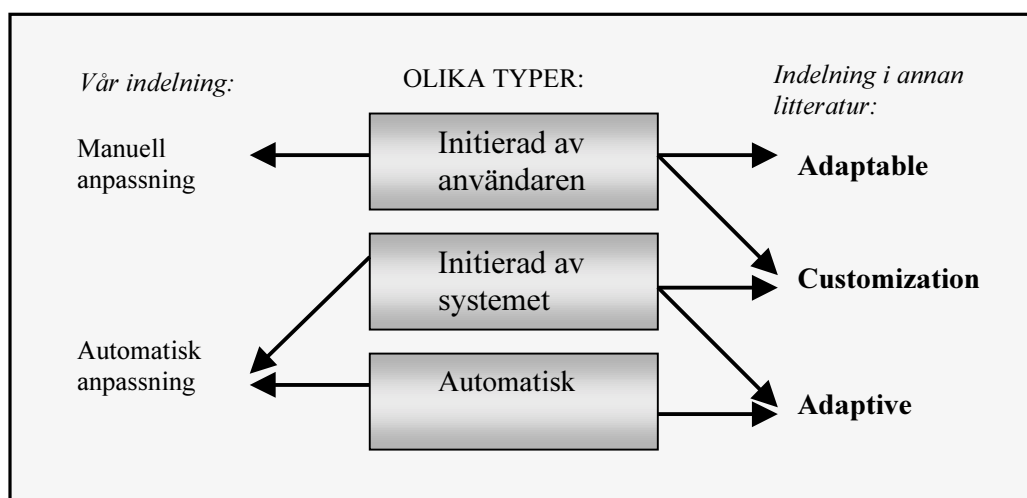
Johansson, Lind och Wahlsten undersöker i sitt examensarbete om den enskilde användaren är tillfredsställd med ett gränssnitt som är utformat för flera användare (Johansson, Lind & Wahlsten, 1997). För att besvara den frågan har de utfört en fallstudie på Norrbottens läns

Landstings Vård Administrativa System (VAS). Studien som omfattade tolv användare av VAS visade att användarna inte var *helt* tillfredsställda med gränssnittet till VAS. Det som de tillfrågade användarna var mest missnöjda med var ojämna svarstider vilka upplevdes stressande och frustrerande. Dessutom tyckte ungefär hälften av användarna att skärmbilden var ologisk och rörig.

Det Johansson, Lind och Wahlsten kom fram till var att en användare sällan blir riktigt nöjd med ett gränssnitt som är utvecklat för flera användare. Eftersom deras studie grundade sig på ett specifikt system och antalet respondenter endast uppgick till tolv personer så finns risken att de mönster de såg var slumpmässiga, men trots det tycker vi att uppfattningen från just dessa användare tydligt stöder idéerna kring anpassning. Långa svarstider från systemet kan minskas genom att låta användaren skapa sina egna menyer och navigeringsmönster och därmed slipper hon krångliga vägar för att nå specifika funktioner. Genom att låta användaren organisera sina egna skärmbilder inom kontrollerade gränser skulle även det kunna leda till en mer tillfredsställd användare.

3.2.2. Automatisk och manuell anpassning

Begreppet anpassning kan delas in i tre olika typer; efterfrågad av användaren, initierad av systemet och automatisk (Benyon & Murray, 1993). Benyon och Murray anser att de två första typerna är en form av det engelska uttrycket customization. Samtidigt omtalas i engelsk litteratur begreppen adaptable och adaptive (Oppermann, 1994; Constantine & Lockwood, 1999) som vi väljer att översätta med manuell anpassning respektive automatisk anpassning. Vid automatisk anpassning är initiativtagaren till anpassningen systemet. För vissa applikationer ges endast förslag till anpassning, för andra genomförs de helt utan användarens medtycke. Manuell anpassning initieras och genomförs av användaren. Figur 8 ger en överblick över begreppen.



Figur 8: Olika typer av anpassning

Automatisk anpassning innebär att det är systemet som initierar och/eller genomför en anpassning. Initieringen kan baseras på så enkla mekanismer som sist-använd- eller mest-använda-listor men även på sofistikerade slutledningsmotorer som försöker analysera vad användaren gör och tillhandahåller automatiska genvägar för att åstadkomma detta (Constantine & Lockwood, 1999).

Från början av 90-talet har mycket arbete lagts på att göra system mer flexibla (Oppermann, 1994). Resultat som är märkbara idag är ordbehandlingsprogram som anpassar sig efter användaren. En användare av Microsoft Word 2000 kan tex. råka ut för att menyerna anpassar sig

efter vilka kommandon hon använde senast. Ett annat exempel är att användaren föreslås använda brevguiden om hon inleder ett dokument på ett sätt som Word anser vara ett brev.

Fördelar med automatisk anpassning är således att anpassning sker efter varje individs arbetsmönster. En användare behöver inte lära sig hur en anpassning genomförs vilket leder till hög utnyttjandegrad (Oppermann, 1994).

Ett problem med automatisk anpassning är att kontrollen över systemet tas från användaren (Constantine & Lockwood, 1999). En helig princip vid utveckling av användargränssnitt har länge varit idén om att användaren ska kontroll över systemet (Shneiderman, 1998). Om en mjukvara uppträder på ett sätt som tar kontrollen från användaren så minskar användbarheten (Constantine & Lockwood, 1999). Det är även dyra utvecklingskostnader för automatiskt anpassningsbara system. Utan att gå in på hur det i detalj fungerar kan nämnas att avancerad och tidskrävande programmering krävs för att utveckla den här typen av system.

Manuell anpassning innebär att användaren själv är den som tar initiativet till anpassningen. Systemet tillhandahåller således något sorts verktyg som användaren kan använda för att genomföra anpassningen (Oppermann, 1994). Det är väsentligt att de verktyg som används för att göra en anpassning är enkla att använda. Annars är det lätt att användaren bortser från möjligheten att anpassa systemet. Det är också viktigt att anpassningsverktyget är avskilt från den övriga applikationen på ett sådant sätt att felanvändning av verktyget undviks.

Det är farligt om anpassningar som kan skapa en ostrukturerad och oanvändbar miljö tillåts (Allwood, 1998). Det är inte meningen att allt ansvar för att designa komfortabla gränssnitt ska föras över från systemutvecklaren till slutanvändaren (Oppermann, 1994). Vidare ska utvecklare inte ta för givet att möjligheten till anpassning utnyttjas. Oppermann refererar till en studie³ som visar på att nybörjare nästan aldrig använder anpassningsfinesser och att experter använder dem i begränsad skala. Fördelar är att användaren alltid har kontroll över systemet och anpassningarna (Constantine & Lockwood, 1999). Användaren sätter sin egen personliga prägel på applikationen vilket kan framkalla en positiv inställning till applikationen.

Man skulle kunna tänka sig att det inte är slutanvändaren som gör anpassningen men att det ändå kallas manuell anpassning. McIlhagga har skrivit en uppsats som delvis handlar om att skapa en anpassningsbar applikation där en administratör kan utföra anpassningar för att skapa olika användarprofiler (McIlhagga, 1998). Han beskriver en nivåstruktur där den anpassningsbara applikationen vilar på en grund som utgörs av de tekniska begränsningarna. Ovanför applikationen men innan användaren finns ett lager som han kallar applikationssemantik. Med applikationssemantik menas här att betydelsen av applikationens funktioner inte ska ändras även om åtkomsten av applikationens funktioner och därmed det totala beteendet ändras.

Malcolms idé om en anpassningsbar applikation bygger på att det under designen av en standardapplikation skapas en avgränsad komponent som ska styra anpassningen. Den administratör som ska installera applikationen i en bestämd miljö kan sedan använda sig av denna komponent för att anpassa gränssnittet efter sina användares behov. Vårt arbete bygger på samma grundtanke som Malcolms; att man ska kunna anpassa gränssnittet, men vi flyttar möjligheten att anpassa funktionaliteten till användaren och överlåter därmed ansvaret för hennes profil på den enskilde individen. I enlighet med vår modell i figur 4 på sidan 10 så skiljer även Malcolm på interaktionen med applikationen och interaktionen med gränssnittet.

Manuell anpassning är det som kommer att stå i fokus för vår studie. Som vi kort nämnde i kapitel 2 Bakgrund finns det möjlighet att individualisera LinqPortal:en och det Linq är intresserade av att veta är om det även skulle vara fördelaktigt att erbjuda en liknande möjlighet och låta användaren anpassa den mobila versionen av portalen på eget initiativ. Under nästa

³ En empirisk studie av Jörgensen och Sauer, 1990

rubrik förklarar vi mer ingående vad som kan vara föremål för anpassning och hur anpassning kan realiseras.

3.2.3. Föremål för anpassning

Reinhard Oppermann som är professor på ”Human-Computer Interactions Institute” i Tyskland har anpassning som ett av sina huvudforskningsämnen. Oppermann säger i en av sina böcker att anpassning av en applikation kan ske med avseende på funktionalitet eller med avseende på användargränssnitt (Oppermann, 1994). Vid anpassning av funktionalitet är både funktionernas omfattning och beteende intressanta medan anpassning av gränssnitt innefattar förändring i hur information presenteras och hur inmatning ska ske. Vi kommer i resterande del av det här underkapitlet att förklara dessa föremål för anpassning utifrån det Oppermann säger i sin bok.

När det gäller **anpassning av funktionalitet** utreder vi begreppen total funktionalitet och aktuell funktionalitet. Vidare behandlar vi makron och programtillägg som båda tillåter användaren att lägga till ny funktionalitet i en applikation. Dessutom hanteras defaultvärden och triggers då de är exempel på hur användaren via olika inställningar kan påverka applikationen.

Den *totala funktionaliteten* är i ett program summan av alla de varierande funktionerna som i princip är tillgängliga via användargränssnittet eller olika kommandon. All funktionalitet i den totala funktionaliteten behöver inte vara direkt åtkomlig för användaren. Den funktionalitet som en användare har direkt åtkomst till vid en viss tidpunkt kallas *aktuell funktionalitet*. Den aktuella funktionaliteten är en föränderlig delmängd av den totala funktionaliteten. Att låta en användare modifiera aktuell funktionalitet är ett viktigt steg mot individualisering. Direktåtkomst innebär att funktionen kan nås direkt genom att använda den för applikationen specifika aktiveringstekniken. Exempel på aktiveringstekniker är att göra menyval, klicka på ikoner eller trycka på en tangent.

Det finns ett flertal alternativ till hur man kan anpassa den aktuella funktionaliteten. Man kan erbjuda olika menyer där funktionsuppsättningen varierar. Man kan gå ett steg längre genom att låta användaren definiera sin egen meny. Det skulle kunna leda till att uppgift- och användarspecifik anpassning uppnås. Ovanstående är exempel på anpassning vad gäller applikationsnivå. Anpassning skulle även kunna ske på dokumentnivå. I Word för Windows tillåts användaren att definiera aktuell funktionalitet för specifika dokument genom det som kallas mallar (eng. templates).

Genom *makron* kan användaren lägga till ny funktionalitet. Ett makro är en sekvens av användarhandlingar som kompileras under ett namn och som sedan känns igen av systemet som en ny funktion. Makron som är direkt åtkomliga är således en del av den aktuella funktionaliteten. Vad som kan vara viktigt att nämna när det gäller makron är att de inte tillför någon fundamentalt ny funktionalitet. Allt som kan utföras med ett makro kan även uppnås genom att enskilt använda de funktioner som redan finns. Styrkan ligger istället i att effektivisera användningen av en sekvens av funktioner som används ofta.

Genom att integrera externa komponenter eller så kallat *programtillägg* kan man utöka funktionaliteten i en applikation. I det enklare fallet kan det handla om makrofiler som kopieras in i en speciell katalog. Ett mer avancerat exempel är separata program som installeras för att sedan kunna tillhandahålla funktionalitet till kanske flera andra applikationer.

Vid anropet av en funktion är det vanligt att olika parametrar styr hur kommandot ska falla ut. Vanligtvis har dessa parametrar *defaultvärden*. Ofta går dessa värden att ändra vilket ger upphov till en viktig aspekt på anpassbarhet. Anledningen till att man använder defaultvärden är för att underlätta för användaren. I många fall är det inte önskvärt att fylla i ett flertal parametrar vid användningen av en viss funktion. För att förändring av defaultvärden ska klassificeras som en anpassning krävs att inställningarna är bestående. En vanlig dialogruta är den som aktiveras när en utskrift är begärd från tex. Word. Där är defaultvärdet för antalet utskriftskopior alltid 1. Detta är oberoende på hur många kopior som skrevs ut vid senaste tillfället. Förfarandet anses således inte som anpassning.

En viktig källa till anpassning är sk. *triggers*. En trigger utförs av ett villkor-respons-par som innebär att när ett visst villkor är uppfyllt eller en viss händelse inträffar så utlöses automatiskt en funktion. Det är inte så vanligt att en applikation stöder triggers av så generell natur att användaren fritt kan ange både villkor och respons för en trigger. Vanligare är att det finns fördefinierade villkor och responser där användaren kan koppla ihop dem efter egna önskemål.

Anpassning av användargränssnitt genomförs huvudsakligen genom att ändra *åtkomst av funktionalitet* eller *layout*. Genom att anpassa åtkomst av funktionalitet kan användaren få en applikation som innehåller den funktionalitet som är intressant för henne och som tillhandahåller högprioriterad funktionalitet enklare än övrig. Att möjliggöra anpassningar av layouten är vanligt. Anpassningar av den typen ger användaren en känsla av att hon kan sätta sin personliga prägel på en applikation. Nedan diskuteras de båda tillvägagångssätten för anpassning av gränssnittet vidare.

Standardvägen till att möjliggöra *åtkomst av funktionalitet* i dagens PC-mjukvara är genom en meny. Vad gäller menyer är det ganska vanligt med anpassning. Förutom möjlighet till omstrukturering tillåter vissa applikationer användaren att själv välja vilka funktioner som ska ingå i menyn. Vidare brukar användare kunna ändra menyalternativets namn så att det passar hennes uppfattning om alternativet. Att lägga till egenskapade makron som menyalternativ är inte heller ovanligt.

Även tangentkommandon, eller kortkommandon, brukar ge tillgång till de vanligaste funktionerna. Anpassning av kortkommandon är möjligt i många applikationer. Man kan ofta skapa nya kortkommandon till befintliga funktioner. När ett eget makro skapas är det inte ovanligt att ett kortkommando kopplas till detta.

För ett antal år sedan kom även knapprader som består av klickbara knappar eller symboler (ikoner) vilket är ett tredje sätt att nå olika funktioner. Knapprader presenteras ofta i egna fönster vilket leder till att de kan flyttas runt och ändras i storlek. Av användaren godtycklig funktionalitet kan representeras i knappraden.

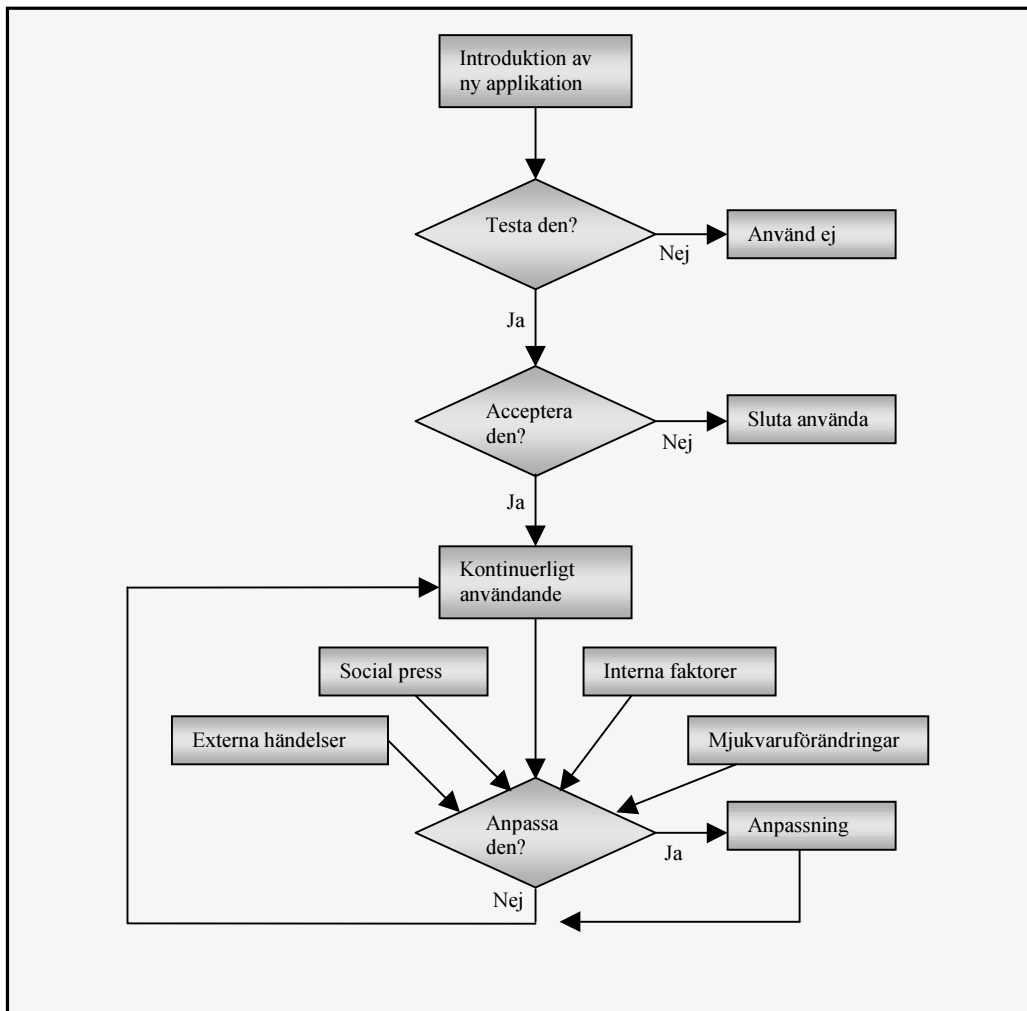
Alternativ för att kunna anpassa *layout* är mycket vanligt. Att kunna göra individuella val vad gäller estetik och design uppskattas ofta. Anpassning av layout kan innebära att utseende på fönster, knappar och formulär ändras. Ett annat område är möjligheten att kunna anpassa format för hur data ska visas som tex. datum, tid och vikt. Man kan även ändra layout genom att välja att inte visa vissa fält som tex. statusrader.

Som vi nämnt tidigare är det anpassning av gränssnitt som är av intresse för oss. Vi har valt att inte titta på anpassning av funktioner i denna studie i överensstämmelse med vår avgränsning. Figur 4 på sidan 10 i bakgrunden visar hur vi tolkar interaktionsbegreppet genom tre möjliga interaktionsvägar mot ett system. De tre är utrustning, applikation och gränssnitt och vi definierar redan där att det är gränssnittsdelen vi är intresserade av. Det känns rimligt att studera anpassning kring hur funktionalitet kan göras åtkomlig eftersom mobiltelefoner kräver en annan sorts design och tankegång när det gäller gränssnittet. Det skulle vara önskvärt att användaren själv kan anpassa hur och i vilken ordning hon ska få tillgång till vissa funktioner. Ovanstående utesluter dock inte att det skulle vara intressant att undersöka anpassning av funktionalitet vid ett annat tillfälle.

3.2.4. Utnyttjandegrad av anpassningsmöjligheter

Applikationer utvecklas allt oftare till att stödja anpassningsbarhet, detta trots bristfällig kunskap kring om användaren verkligen utnyttjar möjligheterna (Mackay, 1991). Mackays studie syftar till att beskriva beslutsprocessen som användare går igenom när de bestämmer om de vill anpassa en mjukvara eller inte och identifiera faktorerna som influerar deras beslut. Mackay definierar anpassningsbar mjukvara som mjukvara som innehåller mekanismer som tillåter användaren att anpassa sin mjukvara utan att skriva kod och där förändringarna sträcker sig över flera sektioner.

Enligt tidigare kategoriseringar i vår studie behandlar således Mackays undersökning manuell anpassning. Figur 9 illustrerar beslutsprocessen för en användare då hon konfronteras med en för henne ny applikation som är anpassningsbar.



Figur 9: Beslutsprocess för utnyttjande av anpassningsmöjlighet

Först och främst måste användaren bestämma sig för om hon överhuvudtaget vill pröva applikationen. Att användaren väljer att pröva applikationen behöver dock inte leda till kontinuerlig användning, för det krävs mer övertygelse om att applikationen är bra och värd att använda. Om användaren accepterar applikationen är nästa steg att besluta huruvida anpassning ska utnyttjas. Det finns en rad faktorer som påverkar detta beslut. De kan grupperas i; externa händelser, social press, mjukvaruförändringar och interna faktorer. Externa händelser innefattar arbetsförändringar, kontorsförflyttningar, personalförändringar osv. som tvingar användare till att stanna upp i sitt vanliga arbete och tänka på hur de organiserar sin tid och användning av olika applikationer. Social press utgörs av avbrott från kollegor som vill dela med sig av erfarenheter eller tillhandahålla hjälp eller av användare som ser något de blir intresserade av. Mjukvaruförändringar kan vara uppgraderingar, programtillägg, nya versioner mm., vilket kan leda till att användaren anpassar applikationer. Till sist kan man finna anledningar för en användare att anpassa i interna faktorer vilka kan vara tristess, oväntad tillgång till tid eller insikt om att något inte fungerar som det borde (Mackay, 1991).

Mackays studie resulterar i ett antal, vad han kallar triggers och barriärer till anpassning. Nedan sammanfattas några av de mest framträdande av dessa:

Triggers – utlösande faktorer för användning av anpassningsmöjligheter:

- När en användare upptäcker att hon gör något om och om igen och väljer att automatisera denna process
- När ett system förändras och en användare önskar att systemet ska bete sig som innan förändringen
- När användaren känner behov av att förhindra händelser som är störande eller långsamma i systemet
- När en användare observerar något som en annan användare gör och sedan önskar göra detsamma
- När användaren är nyfiken och vill utforska applikationen för att den är ny

Barriärer – bromsande faktorer för användning av anpassningsmöjligheter:

- Tidsbrist
- Brister i kunskap om hur anpassning genomförs
- Avsaknad av intresse
- En generell känsla av att problemet i fråga inte är värt att fixa

Mackays studie uppmanar utvecklare till att tänka igenom utveckling av anpassningsbara applikationer. Det är bevisligen så att många användare ignorerar möjligheten till anpassning av en eller annan anledning. Därmed är det förmodligen klokt att ta de triggers och barriärer som finns och sätta in dem i det aktuella sammanhanget och överväga för- och nackdelar innan man börjar utveckla för anpassning.

3.3. Design av gränssnitt

Det vi menar med gränssnitt i uppsatsen är användargränssnittet. Det finns ytterligare typer av gränssnitt, t.ex. systemgränssnitt eller gränssnitt mellan olika objekt i ett system, men det är gränssnittet mot användaren vi behandlar här.

Design betyder konstnärlig formgivning (Svenk-Engelskt Lexikon, 2000). När det handlar om design av mjukvara så ligger fokus på formgivningen snarare än på den konstnärliga biten. Formgivningen innebär då mer än att bara ge en estetiskt tilltalande form, det handlar om att designa för användbarhet.

Med ordkombinationen att designa ett gränssnitt menar vi alltså att skapa ett bra gränssnitt mellan människa och dator.

Det har inte skrivits mycket om hur man bör gå tillväga för att designa ett bra gränssnitt på en liten yta som t ex på en telefon. Det finns inte heller många källor att hänvisa till som beskriver vad som är ett bra gränssnitt på en mobil enhet, däremot finns det många som pekar på att det är viktigt att gränssnitten på mobila enheter är bra (Kuutti 1998; Björk 2000).

Det finns mycket skrivet kring ämnet gränssnitt när det gäller traditionell design av PC-applikationer så därför börjar vi i den änden och drar sedan paralleller till design av gränssnitt på små skärmar.

3.3.1. Traditionell design av gränssnitt

Trots att ämnet MDI är så aktuellt vilket vi beskrev i bakgrunden så är det förvånansvärt att det fortfarande finns och utvecklas system med så dåligt designade gränssnitt. Ett välkänt exempel på ett gammalt inarbetat system med mycket funktionalitet och ett gränssnitt under all kritik är det

svenska bibliotekssystemet GUNDA. När GUNDA byggdes på 1970-talet (Gravas, 2000) fanns det inga verktyg för att göra grafiska gränssnitt. Fokus låg på funktionaliteten som under åren byggts ut och nu är välutvecklad. Det är dock extremt svårt för nybörjare att utnyttja ens en bråkdel av den funktionaliteten eftersom systemet är helt textbaserat.

Ett annat exempel är den allmänna hysterin kring ny teknologi som t.ex. webbteknik och WAP-teknik som har lett till en mängd krångliga interna applikationer. Vid utvecklandet av sådana applikationer kommer återigen tekniken i fokus men av en annan anledning den här gången, det finns nämligen inga användare av den nya tekniken innan den är utvecklad. Eftersom det inte finns några användare att testa prototyper mot så hamnar fokus lätt på den andra biten av utvecklandet, nämligen vad som kan göras med den här tekniken.

Dessa båda exempel representerar varsin typ av system:

- Den äldre typen av system som byggdes under 80-talet eller tidigare då det inte fanns några verktyg för att göra grafiska gränssnitt. Den här typen har ofta byggts på under åren och är fyllda med funktionalitet men ser oerhört tråkiga ut. De är ofta textbaserade och det finns precis som i GUNDA en mängd koder för alla funktioner vilka det krävs övning i att lära sig. Eftersom man – om man har utbildning – kan använda systemet till så mycket så kan det ta emot att börja om från början och försöka bygga på ett grafiskt gränssnitt.
- När det gäller nyutvecklade system som innehåller den allra nyaste och fräckaste tekniken finns det ofta ingen typisk användare att testa på eller designa för eftersom tekniken ännu inte används på marknaden. Detta gör att tekniken hamnar i fokus och man vill vänta med att utveckla ett ordentligt gränssnitt tills man vet att man har en gångbar produkt.

Det finns många författare som har presenterat olika principer för hur man designar ett bra gränssnitt, Ben Shneiderman är en av dem. Han har utformat åtta gyllene grundregler för hur ett användaranpassat gränssnitt bör designas (Shneiderman, 1998):

- Sträva efter *konsistens*. I liknande situationer bör terminologi, menyer, hjälptexter och kommandon vara konsistenta.
- Låt systemet *växa med användaren*. Skapa genvägar och kortkommandon för vana användare.
- Ge informativ *feedback*. Varje operation mot systemet bör ge någon form av återkoppling, vanliga operationer behöver inte ge så mycket respons som ovanliga eller större operationer.
- Designa *dialoger som hänger ihop*. Sekvenser av aktiviteter bör organiseras i grupper som har en början, en mitten och ett slut.
- Ge enkel *felhantering*. Systemet bör designas så att användaren inte gör fel, om det ändå skulle hända bör systemet ge ett meddelande som beskriver felet på ett sådant sätt att användaren inte gör om det. Det får inte heller skrämman upp användaren i onödan.
- Tillåt användaren att *ångra sig*. Det tar bort oro från användaren och stöder det nyfikna utforskandet.
- Låt användaren vara *initiativtagare* istället för att vara den som svarar på datorns aktion. Användare vill känna att de kontrollerar systemet, systemet ska inte överraska.
- *Undvik att överbelasta korttidsminnet*. Följ tumregeln att en människa kan komma ihåg sju plus minus två bitar av information samtidigt.

Shneiderman har också skrivit rekommendationer för hur man bäst organiserar en skärmbild, några av de rekommendationerna är rena tolkningar av de grundläggande principerna, men ett par av dem skiljer sig så mycket att de är värda att omnämnas:

- *Kompatibilitet* mellan presentation och inmatning av data. Presentationsformatet bör överrensstämma med inmatningsformatet.
- *Flexibilitet* för användarkontroll av presentationen av data. Användare bör kunna få data presenterade på ett sätt som är anpassat efter deras specifika behov och känsla.

Enligt traditionell design bör man följa Shneidermans principer, flera andra kända författare har refererat till honom, en av dessa är Ian Sommerville som har sammanfattat Shneidermans principer i fem egna punkter (Sommerville, 1995).

3.3.2. Design av gränssnitt på mobila enheter

Marknaden för mobila enheter och mobila applikationer är ännu i sin barndom och den typiske användaren av sådana produkter har varit en van användare av en stationär dator. Även utvecklarna av den här typen av nya produkter är vana användare av stationära datorer, det har naturligtvis påverkat utseendet på produkterna.

Den typiska mobila enheten har ärvt den inarbetade ”skrivbordsmetaforen”, många av de mobila applikationerna är miniatyrversioner av traditionella kontorsapplikationer (Dahlbom & Ljungberg, 1999). Dahlbom och Ljungberg tror att skrivbordsmetaforen kommer att överleva så länge som mobila IT-användare växlar mellan stationärt och mobilt IT-användande.

Norman behandlade redan 1988 hur gränssnitt mot mindre skärmar som används vid mobilt arbete bör utformas (Norman, 1988). Norman anser att det mobila användandet inte får kräva för mycket av användarens koncentration på skärmen eftersom det är svårt att lokalisera visuella objekt på en så liten yta. Av samma anledning anser Kawachiya och Ishikawa att pekdon inte hör ihop med mobila enheter, då man måste ha visuell kontakt när man väljer och klickar på objekt (Kawachiya & Ishikawa, 1998). Vidare anser Norman att det är viktigt att utveckla alternativa interaktionskanaler till det visuella användargränssnittet då den mobila enheten inte alltid är i användarens fokus (Norman, 1988).

Phone.com är ett företag som utvecklar mjukvara för att kunna erbjuda internetbaserade tjänster till en marknad för trådlösa telefoner (Phone.com, 2000). Deras utvecklingsteam har skrivit riktlinjer för utvecklandet av användarvänliga applikationer på telefoner (Phone.com Application Development Team, 1999):

- *Navigationen* bör vara *intuitiv*, det ska vara ett naturligt flöde. Användare tänker i termer av hur ska jag lösa mitt problem, inte i termer av hur ska jag använda verktyget för att lösa mitt problem. Verktyget ska vara genomskinligt för användaren.
- Applikationen bör vara *konsistent*.
- *Huvudaktiviteter* bör kräva *få operationer*, de aktiviteter som användaren utför ofta bör inte kräva att man letar sig genom flera nivåer i en hierarki.
- Om man ställs inför ett *nytt problem* bör man kunna komma fram till lösningen *intuitivt*. När användaren vill utföra en aktivitet som hon inte utfört tidigare bör hon kunna komma fram till hur det ska göras utan att ta hjälp av någon extern manual.
- *Ovanliga aktiviteter* kan kräva *manual* men ska sedan vara *lätta att komma ihåg*. Om användaren vill göra något som hon inte brukar göra så behöver det inte vara självklart hur det ska gå till, hon kan behöva slå upp det i en manual, men när hon väl har kollat upp hur man gör så ska hon inte behöva gå tillbaka till manualen nästa gång.

Det går att se att dessa riktlinjer har likheter med Shneidermans. Både Phone.com och Shneiderman är inne på samma spår när de talar om vikten av att sträva efter konsistens genom hela applikationen så att användaren känner igen sig och vet vad hon gör.

Vikten av att navigationen är intuitiv styrks också av Shneiderman som uttrycker det som att dialogerna bör hänga ihop. De båda källorna är överens om att det ska finnas en sekvens med ett naturligt flöde så att användaren hela tiden vet hur hon ska komma vidare och inte lämnas i någon avkrok av systemet från vilken hon varken kan komma vidare eller hitta tillbaka.

Phone.com's tre sista punkter kan alla sammanfattas i Shneidermans punkt om att inte överbelasta korttidsminnet. Att Phone.com pekar så mycket på dessa riktlinjer visar att det spelar ännu större roll när skärmen är liten. Eftersom användaren inte kan ha överblick över mer än fyra till fem rader samtidigt så måste designern verkligen tänka på att visa relevant data och få nivåer i eventuella hierarkier.

Shneiderman anser att systemet bör designas så att det kan växa med användaren och motsvara både novisens och expertens behov. Man kan här dra en parallell till Phone.com's riktlinje om att ovanligare aktiviteter kan kräva att användaren tittar i manualen för att lära sig hur man gör men sedan bör det vara enkelt att komma ihåg finessen till nästa gång. Novisen som antagligen inte behöver utföra de mest ovanliga aktiviteterna från början slipper då en massa störande tilläggsinformation om något som ändå aldrig kommer att beröra henne.

Vid sidan om dessa riktlinjer har Phone.com också utvecklat en metod för att underlätta för utvecklare av WAP-applikationer att skapa bra och användarvänliga applikationer, den innebär att man börjar med att bilda sig en uppfattning om vilka typer av användare det finns och sedan bygger man ett antal olika profiler. När man har profilerna klara för sig går man vidare och försöker bryta ner de olika aktiviteterna användarna vill kunna utföra i mindre uppgifter som enligt principen om intuitiv navigation följer på varandra i en naturlig sekvens. Sedan gör man en prioritering mellan dessa uppgifter och kan därefter klassificera dem som delar av huvudaktiviteter eller mer ovanliga aktiviteter. Nu kan man göra ett utkast till gränssnitt som sedan ska testas.

3.4. Trådlös kommunikation med hjälp av WAP

I det här stycket beskriver vi de grundläggande tekniker som den praktiska delen av vårt resultat bygger på. I det första underkapitlet ger vi en översiktlig beskrivning av grundteknikerna GSM och WAP. Sedan förklarar vi innebörden av de tre förkortningarna; XML, DTD och XSL. Avslutningsvis presenterar vi ett exempel för att knyta de olika teknikerna till varandra.

3.4.1. GSM och WAP

Idag är GSM⁴-nätet det dominerande mobiltelenätet i Europa och Asien med täckning i över 200 länder (GSM Data Knowledge Site, 1999). De flesta människor kopplar fortfarande ihop användandet av en mobiltelefon med röstöverföring. GSM som är ett *digitalt* nätverk har dock mer kapacitet än så, man kan använda nätverket för överföring av data. Eftersom GSM-nätet använder radiofrekvenser utgör det en *trådlös* plattform vilket är en grundförutsättning för att ha mobil kommunikation. Hastigheten i GSM-nätet är betydligt mindre än i ett markbundet nätverk vilket ställer helt andra krav på de protokoll som sköter överföringen, det finns inte utrymme för "onödig" information.

GSM är inte den enda påtänkta tekniken för den här typen av dataöverföring även om det länge har varit den enda som funnits i praktiken. Exempel på kommande nätverk med högre hastigheter är General Packet Radio Service (GPRS) och Universal Mobile Telecommunications System (UMTS), dessa bygger på en paketbaserad teknik (Hjelm, 2000). Fördelen med det är att

⁴ Global System for Mobile communications

man kan debitera kunden per datapaket, alltså mängd data istället för per tidsperiod vilket innebär att kunden ständigt kan vara uppkopplad och slipper uppkopplingstiden.

Wireless Application Protocol (WAP) är det protokoll som används för presentation och transport av information för mobila enheter som mobiltelefoner eller PDA:er (Personal Digital Assistant). Idag sker det alltså via GSM-nätet men inom en snar framtid kommer vi att få se det ske via ett GSM-nät som är utbyggt med t ex GPRS eller UMTS.

WAP-specifikationerna är öppna standarder framtagna i WAP Forum, en samarbetsorganisation där de drygt 200 medlemmarna tillsammans täcker upp större delen av världsmarknaden för trådlös data- och röstöverföring (Wapforum, 1999a). Ericsson, Nokia Motorola och Phone.com var initiativtagarna till WAP.

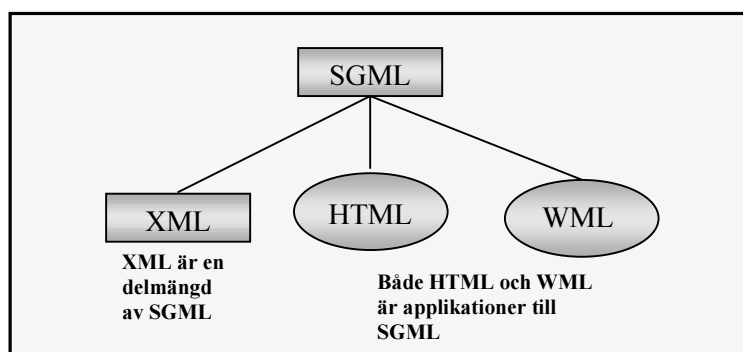
Specifikationerna för WAP (Wapforum, 1999b) inkluderar:

1. Wireless Markup Language (WML), språket som används för att skapa sidor som ska visas på en WAP-terminal (t.ex. en mobiltelefon eller en handdator). WML är liksom HTML en applikation av Standard Generalized Markup Language (SGML) se figur 10.
2. WMLScript, ett scriptspråk som liknar JavaScript.
3. En serie av telefonrelaterade funktioner som att ringa, lägga till namn i adressboken eller skicka SMS-text
4. En protokollstack som hanterar sessioner (motsvaras av HTTP), transaktioner (motsvaras av TCP/IP) och det fysiska lagret (motsvaras av t ex GSM) av kommunikation med en WAP-server eller gateway.

3.4.2. XML, DTD och XSL

Dessa tre begrepp beskriver vi för att skapa en förståelse för hur vi tänker bygga vår prototyp. Förklaringarna till dessa förkortningar har vi hämtat från Whatis?com som är en kunskapsbas kring informationsteknologi (Whatis?com, 2000).

XML betyder Extensible Markup Language och är ett flexibelt sätt att skapa vanliga informationsformat och att dela både dessa format och data på webben, intranät eller någon annanstans. Om man kommer överens om ett standardiserat språk så kan man skicka intelligenta förfrågningar kring ämnet till människor man aldrig träffat och ändå förstå varandra. Precis som i HTML innehåller XML fördefinierade taggar för att beskriva innehållet på en webbsida. Skillnaden mot HTML är att taggarna är obegränsat många och man kan definiera egna taggar som är ämnesspecifika. XML är egentligen en delmängd av SGML, standarden för att skapa dokumentstruktur, se figur 10.



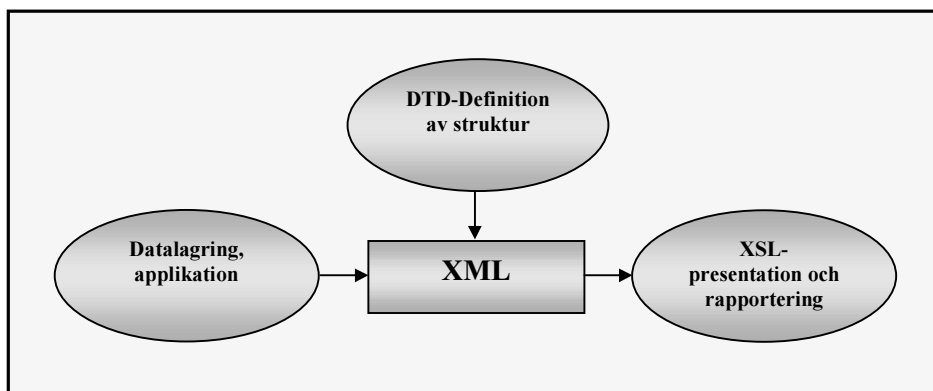
Figur 10: Förhållandet mellan begreppen SGML, XML, HTML och WML

SGML betyder Standard Generalized Markup Language. Ett SGML-dokument är definierat av en valfri Document Type Definition (DTD) som bestämmer informationens struktur.

En **DTD** är en specifik definition som hör ihop med ett dokument och förklarar vilka taggar som beskriver formatet i det dokumentet och hur de ska utföras. Det betyder att en enda SGML-kompilator kan generera en mängd olika typer av dokument, den kollar bara upp vilken DTD som gäller och visar det riktiga dokumentet.

XSL betyder Extensible Stylesheet Language och det är således ett språk för att skapa stilmallar (eng style sheets). Dessa mallar beskriver hur data som skickas på formen XML ska presenteras för användaren. Man tänka sig att XML fungerar som HTML med den skillnaden att det står ren data mellan taggarna istället för text eller grafik. Eftersom taggarna är egendefinierade för att passa ämnet t.ex. <boklista> så kan man med hjälp av XSL beskriva precis vilka taggar som ska presenteras på ett visst sätt. En XSL kan användas för att definiera stilen för ett enda XML-dokument eller så kan många XML-dokument använda samma XSL.

I figur 11 knyter vi dessa tre förkortningar till varandra.



Figur 11: Förhållandet mellan begreppen XML, XSL och DTD

Spindeln i nätet är XML-filen som innehåller ren data enligt en viss struktur men ingen information om hur den ska presenteras. XML:en genereras vanligen av en applikation eller så finns den lagrad t.ex. i en databas. Definitionen av strukturen enligt vilken XML:en är uppbyggd finns i en DTD och information om hur data från XML:en ska presenteras finns i en XSL.

3.4.3. Exempel

Nu är alla begrepp utredda och för att ge en inblick i hur vi ska använda dessa tekniker så visar vi här ett litet exempel innehållandes kodsnuttar i XML, DTD, XSL och WML.

Vi vill skapa ett bibliotekssystem, för att motsvara ett riktigt bibliotek bygger vi det kring begreppen boklista som är tänkt att vara en lista över alla böcker som finns i biblioteket och bok som ska beskriva alla data vi vill ha om en bok. Man kan naturligtvis fylla på med en lista över alla artiklar, en lista över alla cd-skivor etc.

Den första kodsnutten, som visas i figur 12, innehåller den DTD som definierar strukturen på informationen.

```

<?xml version="1.0" ?>
<!ELEMENT BIBLIOTEK (BOKLISTA)>
<!ELEMENT BOKLISTA (BOK)>
<!ELEMENT BOK (TITEL, FÖRFATTARE)>
<!ELEMENT TITEL(#PCDATA)>
<!ELEMENT FÖRFATTARE(FÖRNAMN, EFTERNAMN)>
<!ELEMENT FÖRNAMN(#PCDATA)>
<!ELEMENT EFTERNAMN(#PCDATA)>

```

Figur12: Exempel på en DTD

En DTD är uppbyggd av element som kan vara föräldrar till inget, ett eller flera andra element. De element som står inom parentes efter ett elements namn är dess barn, om det bara står #PCDATA där innebär det att det är ett element som endast innehåller ren data. Ett element kan också innehålla attribut som id eller något annat man vill att det ska innehålla.

Nästa kodsnuitt, som återges i figur 13, är en XML-fil som innehåller de data vi är intresserade av att visa, den kan också innehålla mycket mer data än det vi vill visa. Den här XML:en innehåller en boklista med endast två böcker i. Enligt den DTD som finns här ovanför ligger bok-elementet ”inuti” boklista-elementet.

```

<?xml version="1.0"?>
<BIBLIOTEK>
  <BOKLISTA>
    <BOK>
      <TITEL>Mio min Mio</TITEL>
      <FÖRFATTARE>
        <FÖRNAMN>Astrid</FÖRNAMN>
        <EFTERNAMN>Lindgren</EFTERNAMN>
      </FÖRFATTARE>
    </BOK>
    <BOK>
      <TITEL>Hugo och Josefin</TITEL>
      <FÖRFATTARE>
        <FÖRNAMN>Maria</FÖRNAMN>
        <EFTERNAMN>Gripe</EFTERNAMN>
      </FÖRFATTARE>
    </BOK>
  </BOKLISTA>
</BIBLIOTEK>

```

Figur 13: Exempel på en XML

Nu har vi data i en XML-fil och vi vet att den är strukturerad så som den ska vara för att mottagaren av data ska kunna tolka den rätt eftersom den följer den DTD som hör ihop med XML-dokumentet.

Nästa steg blir att skapa en XSL-fil som ska innehålla information om hur det data som finns i XML:en ska presenteras. Vi har gjort ett mycket enkelt exempel som skapar en sida med en listning av alla böcker som för närvarande finns i bibliotekets register. Se figur 14.

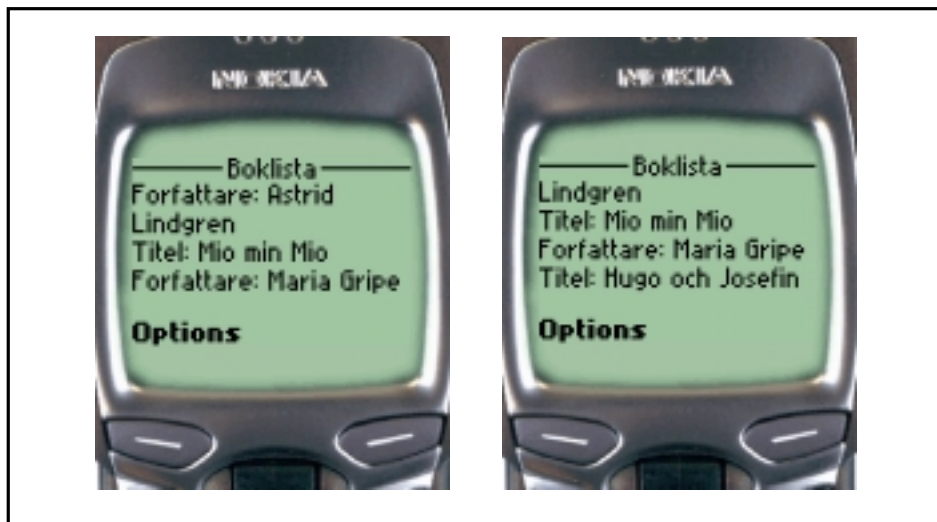
```

<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/TR/WD-xsl"
  xmlns="http://www.w3.org/TR/REC-html40"
  result-ns="">
  <xsl:template match="/">
    <wml>
      <card id="card1" title="Boklista">
        <p><xsl:apply-templates select="/BIBLIOTEK/BOKLISTA/BOK"/></p>
      </card>
    </wml>
  </xsl:template>
  <xsl:template match="/BIBLIOTEK/BOKLISTA/BOK">
    <xsl:apply-templates select="FÖRFATTARE"/>
    <xsl:apply-templates select="TITEL"/>
  </xsl:template>
  <xsl:template match="FÖRFATTARE">
    Författare: <xsl:value-of/><br/>
  </xsl:template>
  <xsl:template match="TITEL">
    Titel: <xsl:value-of/><br/>
  </xsl:template>
</xsl:stylesheet>

```

Figur 14: Exempel på en XSL

Genom att lägga ovanstående kodfiler på en Microsoft Internet Information Server (IIS) är det med hjälp av Active Server Page (ASP) möjligt att köra filerna i en mobiltelefon som stöder WAP. En ASP är en HTML-sida som innehåller ett eller flera script (små inbäddade program) som utförs på en Microsoft Web Server innan sidan skickas till användaren. Vår ASP till det här exemplet samordnar de olika filerna och gör det möjligt att genom en XSL se data från en XML i mobiltelefonen. Resultatet av att surfa till ASP:n visas i figur 15.



Figur15: Resultat från körning av exemplet

Det som sker vid en körning av exemplet är att alla bibliotekets böcker med deras författare och titlar visas i displayen på mobiltelefonen. I vårt exempel fanns det endast två böcker. I figur 15 ses resultatet i två skärmdumpar eftersom det behövdes en skrollning för att se hela resultatet.

3.5. Sammanfattning av teori

Vissa delar av de presenterade teorierna och begreppen är mer betydelsefulla för syftet med vår uppsats än andra, vi tänkte därför peka på vilka delar det är så att läsaren kan ha dem färska i minnet inför de resterande delarna av uppsatsen.

I underkapitel 3.1 Mobil informatik fokuserar vi på att samhället övergår från ett traditionellt industrisamhälle till ett tjänstesamhälle vilket är en grundförutsättning för att mobiltelefonen och WAP-telefonen ska bli lika allmänt spridda arbetsredskap som PC:n är idag. Det är också viktigt att förstå att WAP-telefonen kan komma att användas på olika sätt beroende på hur den mobila situationen ser ut är och då syftar vi på Dahlboms och Ljungbergs tre begrepp wandering, travelling, visiting (Dahlbom & Ljungberg, 1999).

Utifrån ovanstående tror vi att det finns behov av att anpassa applikationer till individuella eller personliga specifikationer som grundar sig i individuella skillnader och varierande arbetsuppgifter. Den typ av anpassning som vi fokuserar på är den manuella, dvs. den är både initierad och genomförd av användaren. När man lägger över kontrollen av anpassningen på användaren är det viktigt att inte ta för givet att användaren utnyttjar möjligheterna till anpassning, det måste med andra ord se bra ut redan från början. Vårt föremål för anpassning är användargränssnittet där man huvudsakligen ändrar åtkomst av funktionalitet och layout.

När vi kommer till designen av gränssnitt så är vi naturligtvis mest intresserade av design på små skärmar och det som då sammanfattningsvis är viktigast att tänka på är att hela applikationen ska vara konsistent, att navigationen ska vara intuitiv och att korttidsminnet inte ska överbelastas.

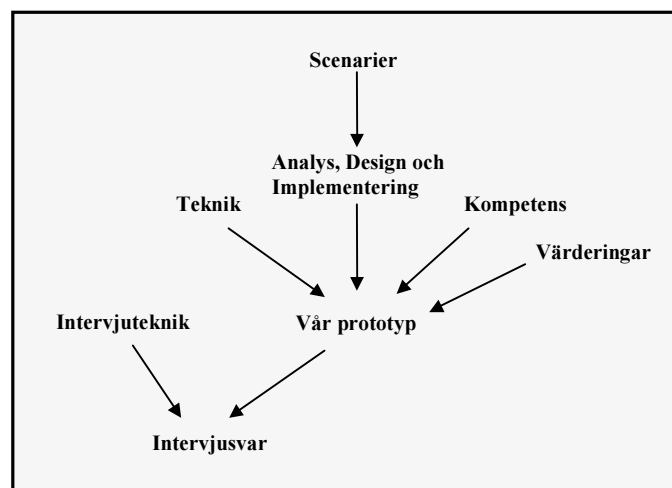
I den tekniska delen av våra teorier presenterade vi de grundläggande begrepp som man bör ha någon form av insikt i för att kunna följa alla delarna av vårt resultat.

4. Metod

Ett vanligt angreppssätt vid forskning inom ämnet informatik är att börja med en etnografisk studie av hur människor/framtida användare betar sig i vissa situationer med fokus på ett eventuellt användande av IT (Dahlbom & Ljungberg, 1999; Hammersley & Atkinson, 1993). Därefter utvecklar man prototyper som testas på användarna tills de är nöjda, de båda grundläggande faserna kan sedan följas av implementering, utvärdering och test.

Det finns två anledningar till att en etnografisk studie inte är aktuell i vårt fall. Dels sträcker sig en välgjord etnografisk studie över en längre tidsperiod än vi har möjlighet till (6 mån – 2 år) och dels finns det idag inga vana användare av WAP-tjänster vilket gör att det är svårt att få fram relevanta uppgifter från framtida användare. Det går t ex inte att fråga en människa, som aldrig använt en WAP-telefon för att läsa e-post på, om hon skulle vilja kunna organisera om utseendet på hur e-posten visas i telefonen. Vi har därför ett annat angreppssätt som innebär att vi med scenarioplanering som grund utvecklar vår första prototyp direkt utan någon initial kontakt med de framtida användarna. Därefter utvärderar vi prototypen och idéerna kring den genom ett antal djupintervjuer.

För att utreda frågan om användaren själv kan skapa sitt eget gränssnitt för WAP-telefonen utifrån tekniken och användaren anpassar vi metoden efter de båda resultat vi förväntar oss att få. Resultaten är prototypen och svaren från utvärderingen. Se figur 16.



Figur16: Vår metod

Utvecklingen av en prototyp kommer att ge oss den tekniska delen av svaret på magisteruppsatsens vetenskapliga fråga. För att komma fram till prototypen ska vi lära oss tekniken, utforma scenarier, genomföra analys, design och implementering, vi ska tillföra vår ökande kompetens och våra värderingar.

Den andra delen av svaret på uppsatsens vetenskapliga fråga får vi genom att göra en **utvärdering** av vår prototyp. Eftersom vi kommer att bygga en fristående prototyp som ska representera hur ett externt anpassningsverktyg skulle kunna se ut, behöver vi en referensapplikation som respondenterna kan utgå ifrån. Det är naturligt för oss att välja Mobile Organizer eftersom vår studie utförs i samarbete med Linq. I utgångsläget av vår studie finns det dock inga användare av Mobile Organizer vilket innebär att vi måste vänta med intervjuerna till slutet av vår studie. Relevansen till vår referensapplikation grundar vi delvis i Mackay's teori

kring beslutsprocessen för utnyttjandegrad av anpassningsmöjligheter som vi återgav i underkapitel 3.2.4. För att avgöra om man ska utnyttja möjligheten till anpassning måste man först ha accepterat applikationen och börjat använda den kontinuerligt. Eftersom våra respondenter har kommit till det stadiet bedömer vi att intervjuer med dem, kring anpassning av en mobil applikation, kan ge rättvisa svar.

4.1. Utveckling av prototyp

4.1.1. Teknikstudier

Vi börjar med att lära oss och testa den teknik vi har för avsikt att använda. Teknikstudierna sker med två olika angreppspunkter, det ena är att vi ska undersöka vilka begränsningar och möjligheter det finns med den plattform vi valt; en telefon av typen Nokia 7110. Syftet med den undersökningen är att vi ska veta inom vilket spektrum vi kan låta användaren anpassa sitt eget gränssnitt när det gäller rent tekniska förutsättningar i telefonen och dess utvecklingsmiljö. Den andra angreppspunkten är att lära oss den teknik vi väljer att använda oss av för att utveckla prototypen med hjälp av vilket användaren ska kunna utföra sina individuella anpassningar för telefonen från sin PC. Exempel på sådan teknik finns beskriven i underkapitel 3.4 Trådlös kommunikation med hjälp av WAP.

För att uppnå det resultat vi vill ha från den här delen av vår studie måste vi koppla ihop flera olika tekniker och miljöer. Det ger oss en ganska komplex arbetssituation och för att reda ut den börjar vi med att i möjligaste mån testa teknikerna var för sig innan vi försöker att koppla ihop dem i hela kedjan. Vi börjar med att skapa oss ett litet grepp om alla delarna och sedan fördjupa våra kunskaper i varje del efter hand som det tillstöter problem.

4.1.2. Scenarioplanering

Vi använder oss av scenarioplanering för att beskriva målgruppen för vårt verktyg, dvs. hur ser den typiske användaren av verktyget ut? Genom att utföra scenarioplanering hoppas vi komma fram till fyra vitt skilda, extrema användare av den mobila portalen och utifrån det behov vi föreställer oss att de skulle ha av vårt verktyg kan vi sedan genomföra analysen av vårt verktyg.

Scenarier är berättelser om möjliga framtider. Dessa berättelser kan hjälpa beslutsfattare att se problem och möjligheter ur nya perspektiv (Dahlbom & Ljungberg, 1999; Fickas et al., 1994; Wilkinson, 1998). Scenarioplanering är ett kvalitativt verktyg som följer systematiska faser, det första steget är att isolera ett beslut eller en fråga, nästa steg är att identifiera de drivande krafterna och därefter avgöra vilka av dem som är oförutsägbara och osäkra, dessa kan man experimentera med i scenarierna (Schwartz, 2000). Sedan väljer man ut två eller tre av dessa faktorer och sätter dem längs korsande axlar. Nu kan man i varje kvadrant bygga ett scenario, alla med olika förutsättningar. En utförligare redogörelse för vad scenarioplanering innebär finns i en artikel på Global Business Network's hemsida (Schwartz, 2000).

Vi börjar med att beskriva hur den tänkte användaren i vårt projekt ser ut, även om vi i vårt resonemang inledningsvis vänder oss mot den tänkte vane användaren av WAP-tjänster så kan det ändå finnas en mängd olika typer av sådana användare. För att skapa en bra uppfattning om hur en framtida användare skulle kunna använda sig av ett nytt system kan olika scenarier men också processen under vilken de skapas vara till stor hjälp (Shneiderman, 1998), därför utformar vi fyra olika scenarier, eller med andra ord fyra olika världar där vi beskriver den typiske användaren i vardagliga situationer. Dessa scenarier ska sedan ligga till grund för analysen av vårt verktyg och för en diskussion kring sambanden mellan vårt tekniska resultat och vår utvärdering.

4.1.3. Analys, Design och Implementering

När vi har skapat våra scenarier och bildat oss en uppfattning om vilka tekniker vi ska använda kan vi parallellt med vidareutbildning inom de olika teknikområdena utföra en enkel analys och design av den prototyp som vi har för avsikt att utveckla.

Analysen och designen utför vi med hjälp av metoden Objektorienterad Analys och Design (OOA&D) som är framtagen av Mathiassen m fl (Mathiassen et al., 1998). Vi har valt den metoden eftersom den är den mest heltäckande objektorienterade metoden som vi känner till, dessutom har vi erfarenhet av att arbeta med OOA&D sedan tidigare projekt vilket gör att vi vet hur vi ska använda de olika delarna av metoden och kan jämföra med hur vi gjort tidigare.

OOA&D är en samansatt metod där många delar härstammar från andra metoder. Metoden är uppbyggd kring sex metodologiska principer, dessa principer utgör grunden för analys och design och bidrar till att knyta ihop allt till en enda metod. Dessa principer är:

- Objekt som samlande begrepp
- En öppen metod
- En situationsbestämd strategi
- Återanvänd förebilder, mönster och komponenter
- En bra design har inga väsentliga svagheter
- En kortfattad dokumentation av hög kvalitet

Under analysen används *objektbegreppet* för att skapa en förståelse för samband mellan vilka delar ur verkligheten systemet ska hantera och hur systemet ska användas. Under designarbetet bygger man vidare på tankegången att all funktionalitet kretsar kring objekt och får då ett enhetligt sätt att tänka på systemet. Det här sättet att använda samma begrepp rakt genom hela utvecklingen är en av de stora skillnaderna mot tidigare inte objektorienterade analys- och designmetoder.

Analys och design kan aldrig utföras på ren rutin, det fordras kreativitet och personliga avvägningar. OOA&D är en systematisk och fullständig metod för att beskriva analys och design. Men samtidigt är metoden *öppen* på ett sådant sätt att den lätt kan anpassas till lokala traditioner och kompletteras med andra metodinslag. Det centrala i metoden är principer och begrepp, vilka tekniker och notationer man sedan använder är av mindre betydelse.

Var i ett projekt man behöver lägga mest kapacitet beror alltid på vilken typ av system projektet syftar till att bygga, vilken typ av relation man har till användarna samt vilka svårigheter man stöter på under projektets gång. Allt detta kan sammanfattas i att det beror på situationen. OOA&D erbjuder olika tekniker för att utforma en konkret och *situationsbestämd* strategi för analys och design.

Ett grundläggande sätt att försäkra sig om kvalitet och förnuftig ekonomi i analys och design är *återanvändning*. Genom objekt- och komponentbegreppet inspirerar OOA&D till att återanvända. Man kan också använda sig av förebilder, t ex ett annat känt system inom samma område. Ett annat tillvägagångssätt är att använda sig av mönster som är ett mer abstrakt och förenklat begrepp, som en mall.

Under designen bör man betona begriplighet, flexibilitet och användbarhet. Begriplighet eftersom designen ska användas som grund för beslut och som kommunikationsmedel för fortsatt arbete. Flexibilitet eftersom all systemutveckling sker i en turbulent miljö, omgivningen och tekniken utvecklas och förändras hela tiden. Användbarhet eftersom systemets framgång står och faller med hur väl det passar in i den organisatoriska och tekniska omgivningen. Arbetet med designen är färdigt när man skapat en design *utan väsentliga osäkerheter*.

Dokumentationen ska användas som referens av dem som utvecklar systemet under det fortsatta arbetet. Dokumentationen är också ett styrningsverktyg och ett avtalsverktyg. Den ska

skapa ett sammanhang under hela systemutvecklingsprocessen. Dokumentet ska hålla hög kvalitet och vara kortfattat för att man snabbt ska kunna skaffa sig en överblick. Dessutom får inte designdokumentet vara så detaljerat att det tar bort den kreativitet som finns hos dem som ska sköta implementeringen.

De fyra huvudaktiviteterna som metoden består av utförs alltid iterativt, där överväganden från en aktivitet ger upphov till nya överväganden från en annan. Den första aktiviteten är **analys av problemområdet**. Här görs en avgränsning av objektsystemet och dess dynamiska egenskaper beskrivs. Objektorienterad analys fokuserar på frågorna: vad ska systemet handla om och vad ska systemet användas till? För att besvara dessa frågor definierar man vilka klasser som ska ingå i systemet, vilken strukturen mellan klasserna ska vara och vilka olika beteendemönster varje klass kan ingå i.

Den andra aktiviteten är **analys av användningsområdet**. Här ska man fastslå de användningsmässiga kraven. Den viktiga frågan är: hur ska systemet användas? För att svara på det beskriver man kraven på användningen av datasystemet, dess funktioner och dess gränssnitt. Kraven testas i det typiska fallet genom att man utformar olika prototyper.

Den tredje huvudaktiviteten är **design av arkitekturen**. Här jämförs kraven på systemet med vad som är möjligt att uppnå på den valda tekniska plattformen. Först utförs en prioritering av de kriterier som ska gälla för arkitekturen, sedan struktureras datasystemets klasser i komponenter och därefter fokuserar man på de dynamiska relationerna mellan datasystemets objekt.

Den fjärde aktiviteten är **design av komponenterna**. Här ska de valda komponenterna designas. Ett vanligt sätt att dela in systemet är i en modellkomponent, en funktionskomponent och en gränssnittskomponent. Analysarbetet överförs till designen som byggs ut med hur realiseringen av analysen ser ut. Designen ligger sedan till grund för implementeringen.

Vi börjar med att *analysera problemområdet*, det vill säga den del av verksamheten som ska stödjas av datasystemet. Resultatet från den här analysen kommer att bli en sammanhängande modell av objektsystemet med dess klasser och deras inbördes struktur.

Under *analysen av användningsområdet*, det vill säga den organisation som administrerar, övervakar eller styr problemområdet, ska vi komma fram till vilka krav som finns på prototypen när det gäller användningen, funktionerna och gränssnittet. Det här presenterar vi i form av vilka användningsmönster som prototypens klasser ska kunna hantera. För att fördjupa vår insikt i vilka möjliga användningsområden som finns så använder vi oss här av metoden scenarioplanering som finns beskriven i stycke 4.1.2 Scenarioplanering.

Under *designen av arkitekturen* vill vi åstadkomma en strukturering av prototypens komponenter i en överskådlig och begriplig arkitektur. Metoden OOA&D säger här att man kan lägga upp en struktur även för systemets processer, hur de samordnas och var det finns flaskhalsar. Det gör vi inte eftersom vår prototyp inte är så omfattande och eftersom vi inte har tillräckligt mycket erfarenhet för att se sådana samband i ett så tidigt skede.

Designen av komponenterna ska resultera i beskrivningar av alla ingående komponenter, vi kommer att dela in prototypen i en modellkomponent vars klasser representerar objekt ur verkligheten och en gränssnittskomponent vars klasser representerar skärmbilder. Metoden OOA&D förespråkar även en funktionskomponent som kan innehålla klasser som endast är bärare av funktioner, det är en inkapsling som vi inte ger oss in på eftersom prototypens omfattning inte är så stor. Vi väljer att lägga all funktionalitet i våra modellobjekt.

4.2. Utvärdering av prototyp

4.2.1. Respondenter

I vår undersökning har det ingått fem respondenter. Vi har valt dessa utifrån ett kriterium:

De ska använda eller ha en förståelse för vad Mobile Organizer innebär. Vi vill att den förutsättningen ska vara uppfylld för att vi ska kunna ställa frågor om vad en möjlighet till anpassning av produkten skulle innebära.

Fyra av dem är anställda på Linq medan den femte arbetar för ett annat datakonsultföretag. De fyra som arbetar på Linq har alla erfarenheter kring Mobile Organizer och använder den i större eller mindre utsträckning. Vår femte respondent har varken sett eller använt Mobile Organizer men utifrån grundliga kunskaper i Microsoft Outlook, WAP-teknologi och erfarenhet av både utveckling och användning av andra mobila applikationer kan respondenten föreställa sig Mobile Organizer på ett vad vi tror ett rättvist sätt.

Anledningen till att undersökningen inte innefattar fler respondenter är att användarna av Mobile Organizer är väldigt få och det är respondenter från denna målgrupp som intresserar oss. Vi vill inte att en övervägande del av våra respondenter ska vara icke-användare av Mobile Organizer och väljer därför att inte söka efter fler sådana respondenter.

Vi har gjort försök att få fler användare av Mobile Organizer att ställa upp på intervjuer men av olika anledningar har vi inte lyckats. En anledning har varit att många av de tillfrågade har använt Mobile Organizer för kort tid för att kunna tillföra studien något.

4.2.2. Utformning av intervjuer

Vi kommer att utföra fem djupintervjuer vilket bl.a. innebär att vi kommer att kunna dra vissa generella slutsatser från de svar vi får. Respondenterna kommer dock inte att kunna stå för någon statistisk representativitet. Det vi är ute efter är snarare ett så brett spektrum av åsikter som möjligt på det här stadiet i utvecklingen. Våra djupintervjuer har en låg grad av struktur vilket innebär att vi under intervjuens gång anpassar de fördefinierade basfrågor som återges i bilaga B efter situationen (Easterby-Smith, Torpe & Lowe, 1997).

Vi kommer utföra intervjuerna genom att en av oss kommer att ställa frågor medan den andre antecknar det svaret som respondenten ger. Vi tror att det ökar riktigheten i anteckningarna om en av oss till fullo kan koncentrera sig på vad respondenten säger och anteckna detta. Dessutom tror vi att det underlättar för respondenten om det hela tiden bara är av oss som han/hon behöver prata med.

Vi har delat in intervjuerna i tre moment (se bilaga B). Det första momentet innebär att vi ställer några frågor kring individen och hans eller hennes bakgrund. Det gör vi för att få en uppfattning av vad det är för typ av person. Områden som behandlas är som nämnts bakgrund, men även teknikintresse och yrke.

I det andra momentet behandlar vi hur respondenterna använder Mobile Organizer. Vi är här intresserade av att ställa frågor som grundar sig i de begrepp och teorier vi presenterar i uppsatsen och i detta andra moment grundar vi frågorna i de designprinciper som återfinns underkapitel 3.3 Design av gränssnitt.

När vi går in i det tredje momentet har vi således fått en uppfattning om hur respondenterna använder Mobile Organizer och då presenterar vi vår prototyp och våra idéer kring den. När respondenterna fått en uppfattning om vad vår prototyp innebär ställer vi hypotetiska frågor om hur användningen av den skulle påverka dem. Vid utformning av dessa frågor utgår vi från anpassningsteorier.

Resultatet vi förväntar oss från intervjuerna är en sammanställning av de åsikter om vår prototyp och våra idéer kring den som respondenterna har. Eftersom vi kommer att utveckla prototypen innan intervjuerna är svaren vi får intressanta för en eventuell vidareutveckling av idéerna.

4.3. Sammanfattning av metod

Den första delen av metoden behandlar den tekniska aspekten på utredningen. Den syftar till att klargöra vilka anpassningsmöjligheter som finns dels utifrån begränsningarna i telefonen och dels utifrån den teknik som finns. För att ta reda på detta **utvecklar** vi en prototyp som kommer att spegla våra värderingar och vår kompetens inom området. Innan vi gör det måste vi dock lära oss den teknik vi ska använda. Först efter att vi har utforskat den kan vi påbörja den egentliga prototyputvecklingen. Vi tar till en början hjälp av scenarioplanering för att utforma fyra olika scenarier som ska ge oss en uppfattning om de tänkta användarna. Därefter genomför vi i sann systemutvecklingsanda analys, design och implementering av den prototyp som sedan ska ligga till grund för utvärderingen av våra idéer kring de begrepp och teorier som finns presenterade i kapitel 3 Teori.

I den andra delen av metoden beskriver vi hur vi går tillväga när vi **utvärderar** prototypen. Vi genomför ett antal djupintervjuer för att få en bred bild av hur respondenterna arbetar, hur de använder sig av Mobile Organizer och hur de skulle kunna använda vår prototyp. Vi genomför intervjuerna med grund i de teorier och begrepp som presenterats tidigare i uppsatsen.

5. Resultat

Målet med studien är att svara på frågan ”Kan användaren själv anpassa sitt gränssnitt till WAP-telefonen?”. Arbetet för att nå detta mål har genomförts i två steg som vi grundar i de två delfrågorna: ”Är det tekniskt möjligt att anpassa gränssnittet i en mobil applikation” och ”Skulle en van användare av en WAP-applikation ha någon nytta av att kunna anpassa sitt gränssnitt så att det passar just hennes behov”. Steg ett innefattar utveckling och implementering av en prototyp och steg två utgör utvärdering av prototypen.

Vägen fram till vår prototyp gick genom att först titta på de tekniska begränsningarna i telefonen, därefter utvecklade vi scenarier för att skapa oss en bild av en eventuell användare, dessa låg sedan till grund för nästa del som var att utföra en analys- och designfas. Genom den skaffade vi oss både en helhetsbild av systemet och mer detaljerade planer på delarna. Allt detta resulterade sedan i implementeringen av prototypen.

Steg två presenteras i underkapitel 5.2 Utvärdering. Där kommer vi således att återge resultatet vi fick från de intervjuer vi utförde. Resultatet kommer att presenteras i tre delar: respondenternas bakgrund, respondenternas uppfattningar kring Mobile Organizer och respondenternas åsikter kring vår prototyp. Vi ville undersöka respondenternas bakgrund för att se huruvida den påverkar deras senare svar. Vi ville dessutom få en uppfattning om hur respondenterna använder Mobile Organizer idag eftersom vi tyckte att det borde ge oss tankar kring om vår idé med en anpassningsmöjlighet skulle påverka deras användning idag positivt eller negativt. Till sist ville vi utvärdera vår prototyp och våra idéer kring den. Resultatet av utvärderingen är således en sammanfattning av respondenternas svar på frågor från ovanstående tre områden.

5.1. Vår prototyp

5.1.1. Tekniska begränsningar för utvecklingen av vår prototyp

Den av oss valda plattformen inkluderar en mobiltelefon, av typen Nokia 7110. Eftersom interaktionsbegreppen utrustning, applikation och gränssnitt från figur 4 på sidan 10 ligger mycket nära varandra kan det vara svårt att särskilja dem och eftersom utrustningen, som i vårt fall är en Nokia 7110, ligger till grund för vad vår prototyp ska kunna göra för anpassningar så behöver vi här redogöra för vilka begränsningar vi kommit fram till, dels i telefonen och dels i tekniken som används runt den. Detta trots att vi har avgränsat oss från att behandla utrustningen. Vi har kommit fram till följande begränsningar:

- Telefonens display kan inte visa färger, den har inte heller så bra grafik att bilder ger ett nämnvärt förtydligande.
- Storleken på displayen är väldigt liten. Det leder till att det inte går att göra klick-knappar för funktioner man alltid vill ha till hands, dessa funktioner måste läggas i en särskild optionsmeny vilket tar dem ett klick från den skärmbild man vill påverka. Menyerna kan inte heller göras speciellt långa innan man tappar överblicken när skärmen är så liten som den är.
- Det saknas tangentbord eller motsvarande enkla möjlighet att mata in text i telefonen. Det gör att det är ett tidsödande moment att försöka skriva ett helt meddelande.
- Det saknas stöd för push-teknik⁵ både i telefonen och i de första WAP-specifikationerna vilka fortfarande är de som gäller.

⁵ Push-teknik genererar information och skickar ut den till användaren utan att hon efterfrågat den.

- Enligt Nokia saknas stöd för vissa WML-taggar som; tab, imgfieldset, em, strong, i, b, u, big, small (Nokia mobile phones, 1999). Dessa begränsningar påverkar utseendet på gränssnittet i telefonen så att man t ex inte kan stryka under text eller göra den kursiv.
- En stor begränsning är det befintliga GSM-nätets svagheter när det gäller debitering. För att användningen ska komma igång måste det vara enklare och gå snabbare. Idag är det alldeles för dyrt att vara ständigt uppkopplad eftersom användaren debiteras per tidsenhet och inte per dataenhet som laddas ner. Därför är det ytterst få som är uppkopplade hela tiden eftersom så mycket tid går åt till återuppkoppling.
- Data måste överföras i relativt små portioner till användaren eftersom bandbredden är så liten.

5.1.2. Scenarier

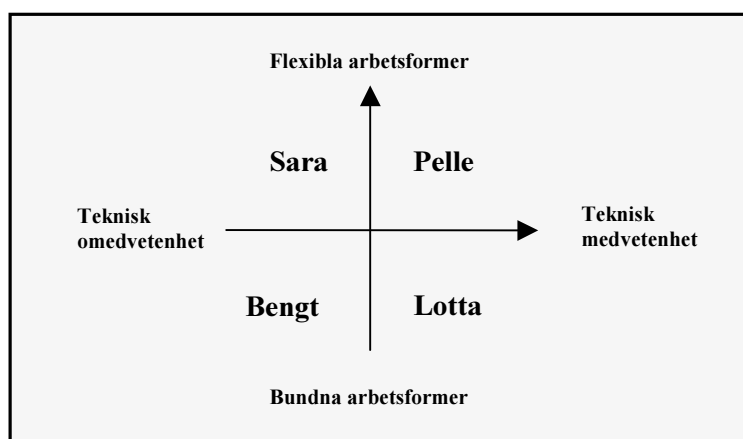
Vi valde att använda oss av scenarioplanering för att beskriva möjliga användare till vårt verktyg. Som utgångspunkt för de scenarier vi ville bygga hade vi således frågan: ”Vem är användaren av vårt verktyg om två år samt hur kommer han eller hon att använda verktyget?”. Anledningen till att vi satte tidpunkten i vårt antagande till två år framåt i tiden var att vi tror att WAP eller liknande teknologi då är mer allmänt känt och används i större utsträckning än idag.

Några av de drivande krafterna vi valde mellan inför skapandet av scenarierna var; samhällets teknikutveckling, flexibla anställningsformer, den mobila livsstilens genomslagskraft och trender i samhället.

Av dessa var de mest oförutsägbara eller osäkra drivkrafterna intressantast och vi har koncentrerat oss på två motstående par:

- Flexibla kontra bundna anställningsformer
- Teknisk medvetenhet kontra teknisk omedvetenhet

I figur 17 åskådliggörs ovanstående två punkter i ett fyrfält och utsatt är även namnen på de påhittade användarna som vi kommer att beskriva senare. Först ger vi dock en beskrivning av vad de två punkterna innebär och de fyra personernas gemensamma bakgrund.



Figur 17: Scenariernas oförutsägbara drivkrafter

Med flexibla anställningsformer menas att en del av samhällets utveckling leder till kortare anställningsperioder, fler arbetsbyten per människa och över huvudtaget större eget ansvar när det gäller vidareutbildning och karriärutveckling. Man skriver inte på ett kontrakt för ett stort företag när man är ung och lämnar över hela ansvaret för sin utveckling till sin arbetsgivare för att sedan

sätta sig ner och vänta på den dagen då guldklockan ska delas ut.

Det är inte alls säkert att det blir på ovanstående sätt, det kan mycket väl ske en utveckling i rakt motsatt riktning där det tvärtom blir otroligt viktigt att ha en trygghet i en livslång anställning, även om det inte ser ut så just nu. Den typen av anställning kallar vi bunden och syftar då på att man gynnas av att vara kvar en längre tid (mer än 10 år) i samma företag.

Med teknisk medvetenhet menar vi att man är allmänt engagerad och nyfiken på ny teknik, man har lätt för att ta till sig och lära sig och tycker att det är roligt och spännande. Motsatsen till detta är naturligtvis teknisk omedvetenhet.

Vi har förutsatt att de fyra personerna till viss del har en **gemensam bakgrund**. De grundläggande förutsättningarna för alla personerna i scenarierna är att:

- De har användning av en företagsportal i sitt arbete – de kan ha olika användningsområden av portalen, det finns olika grader från att enbart använda telefonboken eller att läsa e-post till att utnyttja avancerade tjänster som tidrapportering eller mötesbokning.
- De har mobila inslag i sitt arbete – det kan innebära att de reser mycket i sitt arbete och därför inte kan använda den stationära datorn. Det kan också innebära att de rör sig mycket inom kontoret, sitter i möten och pratar med människor så att det är bekvämt att ha med sig en företagsportal i mobiltelefonen. Ett tredje alternativ är att de ofta befinner sig hos kunden och där kanske inte har tillgång till stationära datorer
- De har eftergymnasial utbildning – det kan vara en civilingenjörsutbildning, en universitetsutbildning eller en intensivkurs inom ett specialområde.

Scenario 1. Pelle 28 år

Pelle är 28 år och arbetar på en konsultfirma. Han är singel och bor centralt. Han har haft fyra arbeten inom samma bransch sedan han tog sin magisterexamen inom datavetenskap för sex år sedan. Hans uppgifter är kundorienterade, han reser mycket mellan olika kunder och försöker se till att hänga med i tekniknyheter för att framstå som kunnig och trovärdig inför kunderna. Han har ett stort kontaktnät och får ca 50 e-post-brev varje dag från kunder och medarbetare som förväntar sig snabba och alerta svar. Han får dessutom ett antal privata e-post-brev varje dag. Pelle är beroende av sin kalender och kommunikationen med kunderna. Han har ofta flera möten varje dag och andra tider som måste passas. Han försöker också hinna med att träna tre gånger i veckan. Varje morgon läser Pelle Dagens Industri och de dagar Computer Sweden kommer ut läser han även den. Dessutom brukar han bläddra igenom Göteborgs Posten på kvällen då han slår sig ner i TV-soffan en stund efter middagen. Pelle är som nämnts mycket intresserad av ny teknik och ägnar gärna sin privata tid åt att sätta sig in i hur nya prylar eller applikationer fungerar. När Pelle blev erbjuden att använda portalen från sin mobiltelefon tyckte han att det var en strålande idé. Nu kunde han hänga med även när han inte kunde koppla upp sin bärbara dator, hinna med att läsa ett par e-post-brev på spårvagnen och alltid ha kontaktinformation till sina kunder nära till hands.

Pelle hittar direkt finesserna med vårt verktyg. Han sätter sig vid sin stationära dator på morgonen för att planera sin dag, antingen på jobbet, hemma eller hos kunden. När han ser i sin almanacka att han har fem stycken möten och nästan ingen tid mellan dem så bestämmer han sig för att kalenderfunktionen på telefonen är den viktigaste idag och därför lägger han den på nivå 1. Han vet också att han kommer att få e-post från kollegan Nils i Stockholm under förmiddagen och aktiverar då ett filter som gör att endast viktig e-post från Nils hamnar i inboxen. Resterande e-post läggs undan i en katalog med dagens datum som Pelle kan gå igenom på kvällen eller nästa dag.

Scenario 2. Lotta 30 år

Lotta är 30 år och arbetar som systemutvecklare på ett större företag. Det är hennes första jobb

och hon har varit anställd där sedan hon tog sin systemvetarexamen för fem år sedan. Hennes avdelning håller på med produktutveckling och hon får allt mer att säga till om allt efter som åren går. Lotta har en familj med två barn och håller just på att köpa hus. Det är viktigt för henne att ha en stadig ekonomi och att inte behöva rota sig på nya arbetsplatser med bara ett par års mellanrum eftersom hon vill ha fasta tider och ett ganska ”inrutat” liv. Lotta är otroligt engagerad och även om hon bara är på jobbet mellan 8 och 17 så får hon förvånansvärt mycket gjort under den tiden och på kvällarna då barnen gått till sängs sitter hon ofta och läser och lär sig nya saker. Hennes arbetsuppgifter inkluderar ofta mobilitet, hon har ansvar för att göra förundersökningar och anpassningar hos kunderna och det innebär att hon periodvis (en till tre veckor i sträck) sitter ute hos kund. Sedan kan det gå en hel månad då hon bara sitter på kontoret. Hon deltar ofta i vidareutvecklingskurser och rör sig internt på det flera våningar stora kontoret.

Lotta har stor nytta av kalendern när hon är ute hos kund, men när hon är på kontoret använder hon kalendern från den stationära datorn och med hennes kvinnliga minne och strukturerade sätt behöver hon inte kolla den flera gånger per dag. Tidrapporteringen är en annan tjänst som hon bara vill ha då hon är ute hos kund, de dagar hon är på kontoret avslutar hon alltid vid sin stationära dator med att fylla i tidrapporteringen. Hon vill anpassa portalen på mobiltelefonen till sig och sitt sätt att arbeta, men efter en tid hittar hon två lägen som hon växlar mellan beroende på om hon är ute hos kund eller på kontoret.

Scenario 3. Bengt 51 år

Bengt är 51 år och utbildad civilekonom. Han är chef för ett medelstort nordiskt företag med ca 800 anställda. Företaget säljer badrumsinredningar. Bengt har en familj med fru och två tonårsdöttrar. Han reser mycket mellan de lokala kontoren och sitter i långa strategimöten, han har bara kontakt med de riktigt stora kunderna. När han kommer hem sent på kvällen är han ofta helt slut och vill bara ha mat och slå sig ner framför TVn. Bengt försöker hänga med och varje morgon läser han Dagens Industri vid frukostbordet, han koncentrerar sig på ekonominyheterna och perspektivsidorna, det är branschnyheterna som är viktigast.

Bengt använder efter en ganska lång startsträcka den mobila tillämpningen till portalen dagligen. Han har klarat sitt arbete i 25 år utan motsvarande tjänster men han har börjat inse att han inte klarar sig utan vissa tjänster, som att alltid kunna läsa sina mail. Rickard från IT-avdelningen har visat Bengt hur han kan ändra sina inställningar på telefonen men han har inte haft tid att engagera sig mer i hur man gör.

Scenario 4. Sara 24 år

Sara är 24 år och arbetar på en reklambyrå. Det är hennes andra arbete sedan hon avslutade en ettårskurs i media och kommunikation. Hon har ett attraktivt utseende och arbetar med layouter, hon träffar mycket kunder och är ofta ute på kundbesök men hon reser aldrig i jobbet. Sara är totalt ointresserad av teknik. Hon har en sambo och en stor vänskapskrets och känner trygghet i det, hon kan tänka sig att hoppa runt mellan olika spännande jobb bara hon får vara kvar i stan så att hon inte tappar kontakten med vännerna. De tidningar Sara läser är modemagasin för att hänga med i de nya trenderna. Det är otroligt viktigt för henne att hänga med, båda som yrkesperson och som privatperson.

Sara använder portalen dagligen och tycker att det är jättehäftigt med all information man kan ha med sig i sin lilla nalle. Första gången hon blev visad vårt verktyg tyckte hon att det var jättebra att hon kunde ställa in det precis som hon ville ha det så då pillade hon lite med det men sedan glömde hon lika snabbt hur det var man gjorde så hon har aldrig använt sig av funktionen igen.

Sammanfattning av scenarierna

Efter framtagandet av dessa fyra olika scenarier hade vi en grundläggande uppfattning om vem användaren av vårt verktyg kan vara. Det underlättade för oss när vi sedan skulle utföra analysen

och även när vi skulle utforma intervjufrågorna eftersom vi då hela tiden kunde jämföra och dra paralleller till vad Pelle, Lotta, Bengt och Sara skulle ha svarat och på det viset ha ett slags förväntat resultat.

5.1.3. Analys och design

Vi valde att dela in vårt praktiska arbete, vilket innebar utvecklingen av en prototyp, i de tre faserna analys, design och implementering. Dessa faser är tre huvudaktiviteter inom systemutveckling. Under analysen och designen försökte vi ha ett helt system i tankarna för att inte låsa oss och begränsa oss för mycket. Under implementeringen som beskrivs i underkapitel 5.1.4 Implementering genomförde vi dock inte alla steg som vi analyserat och designat. Systemutveckling innebär onekligen en iterativ process vilken vi inte kommer att se slutet på innan tiden för det här examensarbetet är slut.

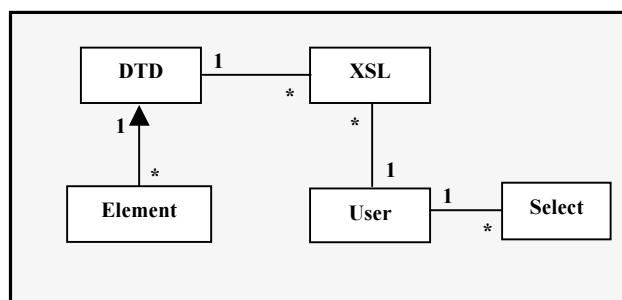
Resultatet från analysen och designen kan, enligt den metod vi beskrivit i kapitel 4 Metod, delas in i följande fyra delar:

- En sammanhängande modell av objektsystemet
- En lista över alla användningsmönster som prototypen ska kunna hantera
- En övergripande bild av prototypens arkitektur
- En beskrivning av de ingående komponenterna

Dessa fyra delar har vi valt att redogöra för i ett analys- och designdokument (se bilaga A). Anledningen till att vi har valt att presentera analysen och designen på det här sättet, i ett och samma dokument, och inte i två separata som Mathiassen (Mathiassen m.fl., 1998) förespråkar beror delvis på det iterativa arbetssättet, analysen och designen går in i varandra och vi anser inte att det ligger något värde i att hålla isär de olika momenten. Dessutom har tidsbristen och den begränsade omfattningen av vår praktiska utveckling lett till att vi inte har gjort en utförlig dokumentation på alla delarna.

Här nedan beskriver vi kortfattat de fyra delarna av analys och design som vi byggt vår dokumentation kring.

Modellen över objektsystemet, se figur 18, består av de fem klasser som vår analys av problemområdet resulterade i. De är DTD, Element, XSL, User och Select. User-objektet ska representera en användare och ha möjlighet att spara val som användaren gör under interaktionen med systemet. För att underlätta detta skapar vi ett antal Select-objekt för varje val som användaren gör vilka sedan kopplas till User-objektet. Det måste även finnas ett objekt som representerar en DTD och som har som uppgift att läsa in en befintlig DTD-fil för att sedan kunna återskapa den struktur som finns i DTD:n. För att DTD-klassen ska kunna göra detta på ett effektivt sätt skapar vi ett Element-objekt. Det representerar således ett element i DTD:n, dvs en rad i DTD-filen. Slutligen behöver vi ett objekt som kan sköta genereringen av ett antal XSL-filer som är utdatat från systemet. Det XSL-objektet behöver en koppling till både User-objektet och DTD-objektet.



Figur 18: Objektmodell av vår prototyp

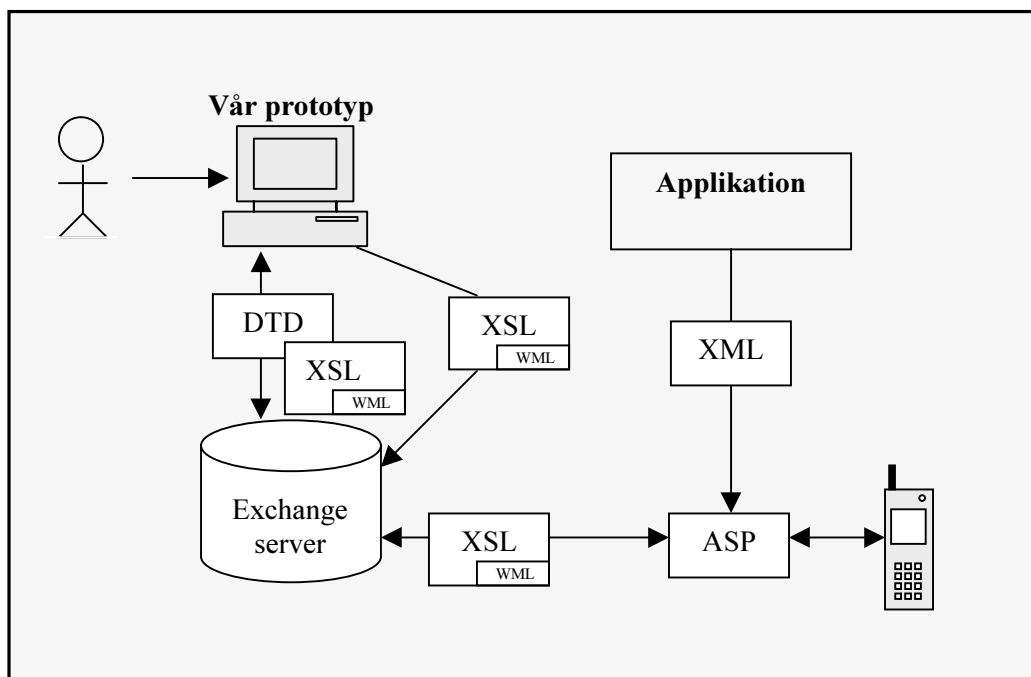
Systemet ska användas för att anpassa ett gränssnitt till en mobil applikation. De flesta **användningsmönstren** handlar således om att användaren ska göra val av olika slag. Vi kom fram till följande lista över användningsmönster.

1. Välja menyelement
2. Välja ordning på menyelementen
3. Välja vad menyelementen ska döpas till
4. Välja vad man vill visa på varje skärmbild
5. Välja hur skärmbilderna ska organiseras
6. Välja vilken startsidan ska vara
7. Välja vilken typ av grafik som ska visas
8. Välja hur skärmbildssekvenserna hänger ihop
9. Välja layout t.ex. vilken font man vill ha
10. Välja i vilket format data ska representeras

Det finns ytterligare förklaring till användningsmönstren i bilaga A.

Vid framtagandet av användningsmönstren hade vi stor nytta av de scenarier vi utformat. Genom att fundera kring hur våra fyra fingerade personer skulle hanterat olika situationer kunde vi komma fram till vilka funktioner som borde finnas med. Pelle t.ex., som är en person med en hög grad av nyfikenhet och som samtidigt är ute och reser mycket, skulle vilja att det fanns en funktion för att ändra vilken startsida han får upp. När han är på kontoret kan han då ha kalendern först men när han är ute från kontoret kommer e-posten upp först. Ett annat exempel är Lotta som engagerar sig mycket i det hon gör och gärna vill ha saker och ting på sitt sätt, hon skulle därför ha stor nytta av en funktion som innebar att hon kunde bestämma hur hennes skärmbildssekvenser skulle hänga ihop. Sara som inte är så bevandrad i tekniken är inte så benägen att använda ett anpassningsverktyg men skulle uppskatta om möjligheten fanns att någon kunde hjälpa henne skapa en enkel miljö med endast de funktioner hon använder och förstår sig på.

Arkitekturen över prototypen och dess omgivning finns i figur 19.



Figur 19: Arkitektur över vår prototyp och dess omgivning

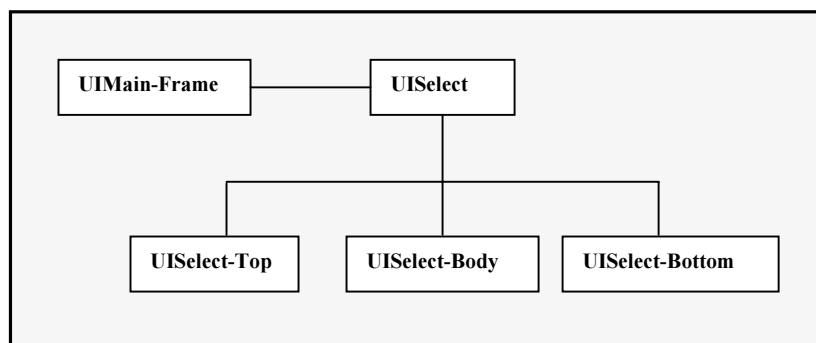
I figuren representeras vår prototyp av en dator. Prototypen interagerar således med användaren och en Exchangeserver. Genom Exchangeservern får prototypen tillgång till den DTD som ligger till grund för applikationen vars gränssnitt ska anpassas. Dessutom ligger det i Exchangeservern lagrat tidigare inställningar som användaren har gjort eller grundinställningar i form av XSL:er. Med hjälp av DTD:n och XSL:erna skapar prototypen frågor till användaren. Utifrån användarens svar skapas sedan nya XSL:er vilka således representerar hur användaren vill att det nya gränssnittet ska se ut.

När användaren sedan använder den mobila applikationen surfar hon med mobiltelefonen till en specifik webbadress som består av en Active Server Page (ASP). ASP:n samordnar vilken datakälla, dvs. vilken XML, det gäller och även vilken XSL som är den första som ska visas. De XSL:er som sedan visas är de som genererats utifrån prototypen.

Vi har gjort en indelning av prototypen i två **komponenter**, vilka är modellkomponenten och gränssnittskomponenten.

Modellkomponenten består av de klasser som finns beskrivna i objektmodellen som visas i figur 18. Istället för att skapa en funktionskomponent, vilket rekommenderas i metoden OOA&D (Mathiassen m.fl., 1998), med extraklasser för att hantera funktionalitet så har vi valt att lägga funktionaliteten i de klasser i modellkomponenten som berörs av den.

I gränssnittskomponenten finns de klasser som används för att rita upp PC-gränssnittet för prototypen. Dessa återges i figur 20.



Figur 20: Gränssnittskomponenten

Vi vill återigen påminna om att det finns ett utförligare dokument som behandlar analysen och designen (se bilaga A). Dessutom vill vi än en gång påpeka att allt från analysen och designen inte är realiserat i implementeringen.

5.1.4. Implementering

Den här delen av resultatet beskriver prototypen och implementeringen av de delar av designen som vi beslutade att realisera.

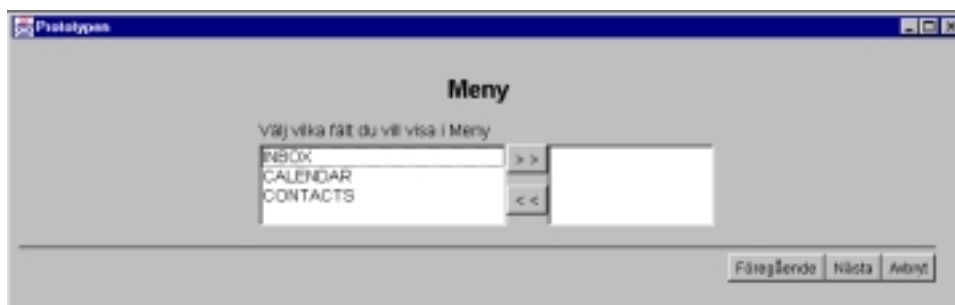
Den prototyp av ett verktyg för anpassning av gränssnitt som vi har byggt är extern. Det innebär att användaren inte kommer åt verktyget från applikationen som ska anpassas. Tanken är istället att användaren ska kunna komma åt att ändra sina inställningar via webben. Den prototyp som finns idag är dock ett helt fristående javaprogram som inte kan nås via webben. Det är inte heller integrerat med några andra applikationer eller system. Det innebär att vi enligt den arkitektur som visades i figur 19 själva har skapat en lämplig DTD som kan fungera som indata till prototypen. Vi har bortsett från att det kan finnas grundinställningar eller tidigare individuella anpassningar i form av lagrade XSL:er som det ska tas hänsyn till, och börjar istället från grunden. När användaren sedan svarat på alla frågor som ställs av prototypen och avslutat programmet skapas XSL:er som lagras undan. När användaren nästa gång använder den mobila

applikationen vars gränssnitt hon modifierat hämtas de nya XSL:erna och XML-datat representeras i gränssnittet med hjälp av de nya inställningarna.

Vår prototyp är utvecklad med hjälp av JDK 1.2 som är ett utvecklingsverktyg för javaapplikationer. Som texteditor har vi använt editplus vilket är en editor som stöder java.

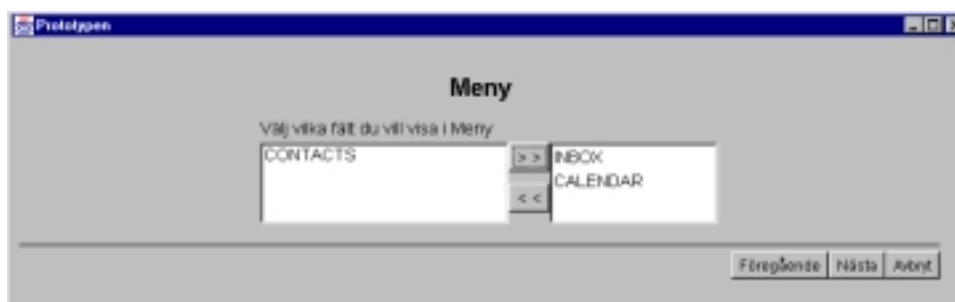
När vi byggde prototypen var vi inspirerade av de olika guider, så kallade wizards, som finns i en rad befintliga applikationer. Det innebär att användaren får upp ett antal alternativ som hon kan välja bland och sedan klickar hon sig vidare till nästa val. Navigeringen sker med hjälp av framåt- och bakåtknappar samt en knapp för att avsluta. Grundidén är att det ska vara lätt att använda prototypen, den ska vara intuitiv och ändå kunna genomföra kraftfulla förändringar i gränssnittet. Prototypens gränssnitt består av två listor. Den vänstra av dessa listor är i utgångsläget alltid fylld med alla de element som är direkt genererade ur den DTD som ska användas. Användaren kan sedan göra val ur denna vänstra lista och det gör hon genom att klicka över ett element till den högra listan som ursprungligen är tom. För att flytta element mellan listorna finns det två knappar med pilar åt höger respektive vänster. När användaren väljer element ur den vänstra listan och flyttar över dem till den högra så innebär det två saker. För det första har användaren nu valt att elementet ska synas i mobiltelefonen på den sida i telefonen som prototypfönstret just nu representerar. Vilken sida det är ser man på rubriken. För det andra väljer hon i vilken ordning de valda elementen ska visas. Vi kommer nedan att exemplifiera vår prototyp med hjälp av en användningssekvens, där vi även visar upp en del av resultatet i mobiltelefonen.

Det första valet handlar om huvudmenyn i telefonen. Användaren väljer vilka aktiviteter hon vill ha tillgång till. I figur 21 visas skärmbilden för det och där kan användaren således välja mellan INBOX, CALENDAR och CONTACTS. Dessa element är direkt hämtade från den DTD vi använder för prototypen.



Figur 21: Skärmbild från vår prototyp: Meny

Nästa skärmbild återges i figur 22, den visar vilka element användaren har valt att ha med i sin meny. Det visar även i vilken ordning användaren vill visa elementen i menyn. Knapparna mellan de två listorna gör att användaren kan modifiera sina val tills dess att hon är nöjd. När hon är det trycker hon på Nästa-knappen.



Figur 22: Skärmbild från vår prototyp: Meny, där val är gjord av användaren

Eftersom INBOX var det element som användaren valde först kommer användaren till en sida som behandlar inboxen, vilken visas i figur 23. Här kan hon välja mellan elementen FOLDER och EMAIL. Användaren väljer alltså om hon i sin inbox i mobiltelefonen vill se bara email som ligger direkt under inbox eller om hon även, eller kanske endast, vill se alla katalogerna (eng. folder).



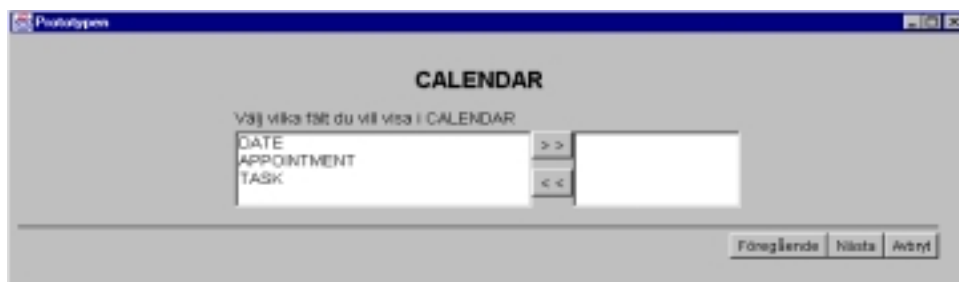
Figur 23: Skärmbild från vår prototyp: Inbox

Nästa skärmbild, se figur 24, visar ett tillfälle där användaren har valt ett element som har barn. I det här fallet valde användaren elementet EMAIL som har en rad underelement, här även kallade barn. Då expanderas skärmbilden och ett nytt listpar visas. I den vänstra listan av detta nya par visas barnen till det valda elementet, i detta fall EMAIL. Det är fortfarande strukturen i DTD:n som är det väsentliga. Det är genom den prototypen vet huruvida ett element har några barn eller inte. Prototypen är således dynamisk i den bemärkelsen att den genererar nya listpar snart användaren väljer ett element som det finns barn till. När det tar stopp dvs. användaren har valt allt hon vill se på en denna sidan så klickar hon på Nästa-knappen och hon kommer till en sida som behandlar nästa aktivitet hon valde att ha med i huvudmenyn.



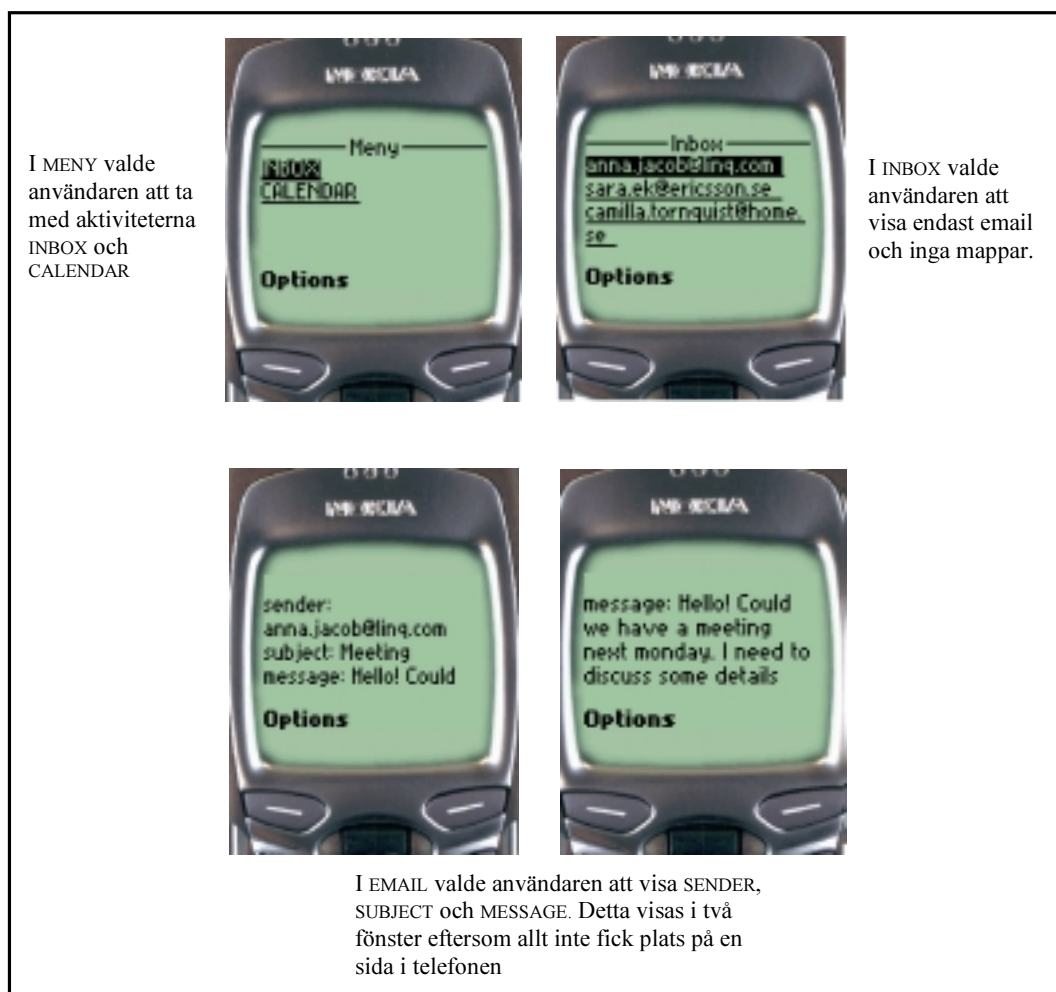
Figur 24: Skärmbild från vår prototyp: Inbox, användaren har valt ett element som har barn. Därmed expanderar skärmbilden med ett nytt listpar.

I vårt exempel var nästa aktivitet som användaren valde att ha med i huvudmenyn CALENDAR. Hon får således se en sida som behandlar detta element, se figur 25. Där fortsätter hon att göra sina val på samma sätt som för INBOX men självklart med nya element som är specifika för kalendern.



Figur 25: Skärmbild från vår prototyp: Calendar

När användaren sedan är klar med hela sekvensen väljer hon Avbryt. Meningen var att det på sista skärmbilden skulle stå Slutför istället för Avbryt på den knappen men det var en typisk detalj som vi valde att prioritera bort. Det som händer när användaren trycker på Avbryt är att programmet avslutas och med hjälp av alla de inställningar som användaren har gjort genereras det ett antal XSL:er som lagras undan. Resultatet av detta blir att nästa gång användaren använder den mobila applikationen som anpassningen utfördes på så är det de nya inställningarna som gäller. I figur 26 visas resultatet i mobiltelefonen för de inställningar som gjorde kring INBOX i exemplet ovan. Den data som visas i skärmbilderna kommer från en XML som vi skapat för att kunna köra prototypen.



Figur 26: Resultat efter anpassning med vår prototyp i mobiltelefonen.

Om man ser till vad vår prototyp har resulterat i så är några av de bitar vi utförde i analysen och designen inte realiserade. För att klargöra vad vi har utfört tänkte vi här i korthet diskutera kring de fyra punkter som vi delade in analysen och designen i.

Objektmodellen som visas i figur 18 på sidan 41 stämmer helt överens med det vi byggt. Av **användningsmönstren** är de två första punkterna realiserade, dessa innebär att användaren själv kan välja vilka funktioner hon vill kunna visa i menyn och i vilken ordning hon vill visa dessa (se lista sid. 42). Dessutom är punkt fyra implementerad och det innebär att prototypen innehåller funktionalitet för att välja vilka element användaren vill visa på en sida. I viss mån kan man även säga att punkt fem, som innefattar organisation av en sida, är realiserad eftersom användaren har möjlighet att välja i vilken ordning elementen ska visas på en sida. Vad gäller **arkitekturen** i figur 19 på sidan 42 från analysen och designen så stämmer den inte riktigt överens med verkligheten. Vår prototyp är till skillnad från bilden helt fristående från andra system. Till sist är de två **komponenterna** vi delade in prototypen i realiserade.

5.2. Intervjusvar

Grunden till vår utvärdering består av fem intervjuer. Målet med intervjuerna var att ta reda på vad respondenterna tyckte om idén att själva få anpassa en mobil applikation. Som referensapplikation har vi använt Mobile Organizer som var känd för fyra av de fem respondenterna. Enligt motivationen som vi återgav i underkapitel 4.2.1 Respondenter är även den femte respondentens svar av värde för oss.

5.2.1. Respondenternas bakgrund

I bakgrundsdelen av intervjuerna var syftet att ta reda på vilka grundförutsättningar de fem respondenterna hade. Vi ställde frågor kring erfarenhet, arbetssituation och teknikintresse.

För att ta reda på vilken **erfarenhet** respondenterna hade kring IT frågade vi efter deras bakgrund och vidare hur många år de arbetat i branschen. Vi fick mycket varierade svar. Alla hade i någon mening studerat på högre nivå än gymnasiet men de skilde sig från varandra i studielängd och inriktning. Studierna varierade sig mellan 30 poäng till 160 poäng och riktade sig mot data eller ekonomi. Vad gällde år i branschen skiftade det från knappt 1 år till 14 år.

Respondenternas **arbetssituation** idag var även den skiftande. Gemensamt var dock att alla arbetade på IT-företag. Fyra av dem arbetade på Linq medan den femte arbetade på ett annat datakonsultföretag i Göteborg. Bland arbetsuppgifterna hittade vi projektledning, systemutveckling, försäljning och metodutveckling. En av respondenterna hade varit med vid utveckling av en mobil klient till ett tidrapporteringsystem. Tre av de tillfrågade arbetade den mesta tiden från kontoret och var i princip aldrig ute och reser. De två andra rörde på sig desto mer. Personen som arbetade med försäljning var ofta ute på kundbesök, minst 2 gånger per vecka och den andre reste ca 1 gång per vecka.

För att få en uppfattning om respondenternas **teknikintresse** frågade vi hur de använder sin mobiltelefon och huruvida de hade dator hemma och om så är fallet hur de använde den.

Två av de tillfrågade använde mobiltelefonen endast till att ringa och ta emot samtal på. De var måttligt intresserade av de finesser som fanns utöver ringfunktionen. Två andra var som de själva uttryckte det medel användare av finesser utöver ringfunktionen. De använde de funktioner de hade nytta av. En av dem uttryckte det så här: ”Så där bra på att använda finesserna, ungefär medel tror jag”. Den sista respondenten var ytterst nyfiken på alla finesser och använde dem till max.

Alla utom en av de tillfrågade hade dator hemma. Av de fyra som hade dator var användningsområdena olika. En använde datorn endast som kommunikationsverktyg. En använde

datorn hemma endast till arbetsrelaterade uppgifter. De två resterande använde dessutom datorn som ett fritidsintresse. För dem var således IT ett intresse även utanför jobbet.

5.2.2. Respondenternas uppfattningar kring Mobile Organizer

Under detta moment var målet att ta reda på hur de tillfrågade använder Mobile Organizer och vad de tyckte om utformningen av produkten. Dessutom har vi ställt några frågor kring produktens omgivning. Eftersom en av respondenterna inte hade någon erfarenhet av Mobile Organizer är det under detta underkapitel endast svar från fyra av respondenterna som återges.

För att ta reda på hur de tillfrågade **använde** Mobile Organizer ställde vi frågor kring hur ofta och när de använde Mobile Organizer, vilka aktiviteter de använde mest, om de tyckte Mobile Organizer var krånglig, om de alltid förstod i vilken nivå i applikationen de befann sig, om de upplevde att de hade kontroll över systemet och om de brukade tänka på vilka konsekvenser deras agerande skulle få när de använde en ny funktion i Mobile Organizer.

Tre av respondenterna använde Mobile Organizer kontinuerligt. En av dessa tre använde den minst en gång om dagen och det var när personen i fråga var ute från kontoret eller hemma. Personen resonerade så här: *”Jag har ett personligt behov av att veta om jag har fått mail och med hjälp av Mobile Organizer slipper jag dra igång datorn hemma och koppla upp mig”*. De andra två som använde Mobile Organizer kontinuerligt gjorde det för att hålla sig uppdaterade då de inte var på kontoret. Den fjärde personen använde som sagt inte Mobile Organizer kontinuerligt utan hade egentligen som syfte att lära sig Mobile Organizer för att kunna förevisa den på mässor.

De aktiviteter de av de tillfrågade som använder Mobile Organizer kontinuerligt använde mest var otvivelaktigt e-post-funktionen, som innebar att man kunde läsa och skicka e-post. Det var att läsa e-post som respondenterna gjorde mest. En av respondenterna kunde någon gång då och då även titta i kalenderfunktionen för att kontrollera om en annan person var ledig vid en viss tidpunkt.

När det gällde om Mobile Organizer var krånglig att använda gav respondenterna varierande svar. En av dem tyckte att applikationen var mycket krånglig. Det tog lång tid att komma in i den och displayen var alldeles för liten. Personen kom senare i intervjun fram till att det nog egentligen inte var produkten som det var fel på utan problemet låg i att tekniken runt om inte var riktigt mogen ännu. Respondenten uttryckte det så här: *”Egentligen är det inget fel på produkten, det är bara det att tekniken ännu inte är mogen”*. Vad gällde de övriga tre respondenterna var de ganska eniga om att Mobile Organizer inte var krånglig att använda till de aktiviteter de använde, vilket huvudsakligen var att läsa e-post. Däremot var kalenderfunktionen mycket krånglig. Att det även var lite besvärligt att skriva brev skyllde de på tekniken runt produkten.

På frågan till respondenterna där vi undrade om de alltid förstod var i de olika nivåerna de befinner fick vi ganska enhetliga svar. Alla fyra tyckte att det var relativt lätt att förstå var i applikationen de befann sig men tre av fyra tyckte att det kunde vara svårt att ta sig tillbaka i nivåerna. *”Det är inte alls lika intuitivt”*, var det någon som uttryckte sig.

Alla respondenter kände att de hade någorlunda kontroll över systemet.

Under intervjuerna uppgav alla att de inte var särskilt eftertänksamma i sitt användande av Mobile Organizer. *”Bara tuta och kör, testat och ser sen hur det går”*, var en av respondenternas svar till denna fråga och de andra tänkte i samma banor.

Nästa frågegrupp kring Mobile Organizer innebar frågor om hur respondenterna uppfattar **utformningen** av produkten. Vi ställde följande frågor: tycker du beteendet i applikationen är konsistent, hur upplever du skärmbilderna och hur upplever du sekvensen mellan skärmbilderna.

Vad gällde applikationens konsistens så hade respondenterna ganska liknande uppfattningar. Vid intervjuerna gav vi ledord i denna fråga där vi beskrev konsistens i sammanhangen Microsoft Outlook, meny, termologi och navigering. Först och främst tyckte alla att det var bra

överensstämmelse mellan Microsoft Outlook, vilket är applikationen som Mobile Organizer inspirerats av, och Mobile Organizer. Även konsistens kring de övriga begreppen var bra och det var bara en av de tillfrågande som hade något väsentligt att påpeka. Det var att navigeringen fungerade olika beroende på var i applikationen man befann sig, vilket tydde på inkonsistens.

Undersökningen visade att respondenterna var nöjda med skärmbilderna. De hade några önskemål om förändringar varav ett var att man skulle kunna se mer än fem e-post-meddelanden på skärmen samtidigt.

Även när det gällde skärmbildssekvenserna verkade respondenterna nöjda. Undantaget var dock när man ville återvända från en djupare liggande nivå. Då kunde det vara svårt att förstå logiken.

Vi hade även en del frågor som behandlade **omgivningen** kring Mobile Organizer. Vi ställde frågorna: hur tycker du att svarstiderna är, använder du dig av möjligheterna till att göra egna inställningar i Mobile Organizer idag, vet du att det finns möjlighet att göra inställningar via webben, varifrån skulle du föredra att göra inställningar och hur blev du motiverad till att använda Mobile Organizer.

Vad gäller svarstiderna var alla nöjda utom en av respondenterna som tyckte att det borde gå snabbare. ”*Man jämför ju med datorn, där det bara är till att klicka så händer det direkt*”, var respondentens ordval. Däremot tyckte alla att uppkopplingstiden var alldeles för lång.

Idag finns det vissa möjligheter till att göra egna inställningar för Mobile Organizer. Det är möjligt via mobiltelefonen men även via webben. Det man kan göra att välja om man bara vill se oläst e-post eller om man vill se all e-post, man kan välja språk och man kan välja vilka av huvudfunktionerna man vill ha med. Huvudfunktionerna består av Inbox, Calendar, Contacts, Drafts, Tasks, Notes, Journal, Sent Items och Que. Två av de tillfrågade hade använt möjligheten till att göra egna inställningar. Det de hade använt var möjligheten att ändra mellan att se all eller bara oläst e-post. Det var bara en som hade sett och testat möjligheten att göra inställningar via webben men det kan finnas en förklaring i att funktionen var helt ny. Alla var överens om att det absolut smidigaste sättet att göra egna inställningar är via webben.

Alla respondenter började använda Mobile Organizer på eget initiativ. Tre av dem hittade motivation till användning genom sin egen nyfikenhet på produkten medan den fjärde behövde lära sig produkten för att kunna demonstrera den på mässor.

5.2.3. Respondenternas åsikter kring vår prototyp

Det tredje momentet i intervjuerna behandlade hypotetiska frågor där respondenterna fick tänka sig att de hade ett verktyg som fungerade som vår prototyp. Innan den här fasen gick vi grundligt igenom hur prototypen fungerar och även de tankar som vi hade om hur den skulle kunna fungera men som inte var implementerade. Efter att ha visat prototypen diskuterade vi således kring ett papper där befintliga och tänkta funktioner fanns listade (se bilaga A 1.6) Frågor som var aktuella i den tredje fasen var följande: tror du att du skulle bli mer nöjd med Mobile Organizer om du hade möjlighet att anpassa den efter dina egna önskemål och arbetsuppgifter, om det fanns möjlighet till anpassning skulle du då vilja vara initiativtagare till anpassningen eller skulle du föredra att systemet var det, om systemet var initiativtagare skulle du då vilja bli tillfrågad innan anpassningen genomfördes eller skulle du föredra att det skedde helautomatiskt, skulle du bli irriterad om det inte var du som hade den kontrollen och om det fanns möjlighet till anpassning, vilka av funktionerna på listan (se bilaga A 1.6) skulle du då vilja kunna anpassa? Vi vill påpeka att vi nedan även kommer att redogöra för den femte respondentens tankar. De är dock en lite annan vinkling på frågorna till den respondenten men vi integrerar det på ett passande sätt.

På frågan om respondenterna skulle bli nöjdare i sin användning av Mobile Organizer om de hade möjlighet till att anpassa var svaren varierande. Mycket berodde på hur de hade svarat tidigare. Respondenten som inte använde Mobile Organizer kontinuerligt utan använde produkten för att kunna demonstrera den på mässor var inte särskilt imponerad av tanken på att kunna

anpassa. Respondenten i fråga tyckte dock att möjligheten skulle finnas för dem som är intresserade, även om respondenten själv inte skulle använda den. Däremot hade respondenten en tanke om att det skulle vara möjligt för en administratör att skapa profiler som var anpassade efter olika individer. Även en av de andra respondenterna uppgav att det inte var av intresse att använda en anpassningsmöjlighet, möjligen om det fanns på webben. Även den respondenten uttryckte ett önskemål om att en administratör skulle utforma profiler till användarna. Två andra av de tillfrågade skulle använda verktyget, och en av dem var mycket positivt inställd till tanken. Den femte respondenten som inte hade använt Mobile Organizer uppgav följande åsikt kring frågan om anpassning skulle öka tillfredsställelsen hos en användare av en mobil klient med Microsoft Outlooks funktionalitet: *”Människor som har telefonbakgrund kommer inte på tanken att de ska anpassa men de om har PC-bakgrund är redan inne i den tankegången och vill nog kunna ändra och ställa in efter sina personliga behov. Det skulle göra dem mer tillfredsställda. Man skulle nog inte använda alla finesser som finns, men det är bra att ha möjlighet till det”*.

Vad gällde frågan om huruvida användaren själv eller systemet skulle vara initiativtagare till eventuella anpassningar var det bara en som uppgav att det skulle vara bra om Mobile Organizer själv anpassade sig efter hur användaren använde systemet. På följdfrågan om det var önskvärt med en förfrågan från systemet innan anpassningen genomfördes svarade respondenten att det inte var nödvändigt så länge möjligheten till att stänga av den automatiska anpassningsfunktionen fanns tillgänglig. *”Även om systemet skulle sköta anpassningarna automatiskt måste det vara jag som har den yttersta kontrollen och därmed måste jag kunna stänga av den funktionen”*, resonerade respondenten. Anledningar som de andra respondenterna uppgav som orsaker till att systemet inte skulle vara automatiskt anpassningsbart var just att de inte ville tappa kontroll men även att det inte kändes väsentligt för denna typ av system. *”Ska det vara någon typ av anpassning ska de vara jag själv som genomför dem”*, sa en av de tillfrågade.

Den sista frågan i detta moment rör den lista av befintliga och tänkta funktioner som vi tog fram under analysen (se bilaga A 1.6). Vi behandlar här huruvida respondenterna finner dessa funktioner användbara eller inte.

- Vilka funktioner menyn ska innehålla
Tre av respondenterna tyckte att detta var en mycket bra idé. De motiverade med att det kändes väsentligt att kunna välja bort de aktiviteter som man aldrig använder.
- I vilken ordning menyvalen ska visas
Det här kändes inte lika väsentligt för de tillfrågade. Någon uppgav lite halvhjärtat att det kunde vara bra men hade ingen riktig motivering till varför.
- Vad menyvalen ska heta
Ingen av respondenterna tyckte att detta var någon bra idé. Anledningen var att funktioner borde heta det de heter i Microsoft Outlook, det skulle bli svårt för en administratör att hjälpa till om användare skulle döpa saker till egna namn.
- Vad man vill visa på varje sida
Fyra av de tillfrågade tyckte att detta var en bra idé. Den främsta motiveringen var att det var bra att välja bort saker man inte vill se på en sida då skärmbilden redan från början är så begränsad.
- Hur varje sida ska organiseras
Tre av respondenterna tyckte att detta var bra. *”Om det är något på en sida som man är särskilt intresserad kan det vara bra om man själv får placera det först på sidan”*, var en av motiveringarna till denna idé.
- Vilken startsida man vill ha
”Att kunna bestämma sin egen startsida är mycket bra”, sa en av respondenterna och på den punkten höll alla med. Är det en aktivitet man använder mer än andra är det bra komma direkt till den var det vanligaste resonemanget bakom detta.

- Vilken typ av grafik man vill se
Alla tyckte att det kunde vara bra att välja till eller välja bort grafik. En av motiveringarna löd: ”*Om till exempel ikoner finns och de är bra kan det vara bra att få välja om man vill se dem eller om man vill se text*”
- Bestämma hur skärmbildssekvenserna hänger ihop
Fyra av respondenterna tyckte att detta var en bra idé. Önskemål om att det skulle ske på ett mycket enkelt sätt uttrycktes dock av två respondenter. En annan av respondenterna var mycket positivt inställd till tanken: ”*Det är ju jättebra att låta användaren själv välja hur man vill navigera sig i systemet. Det är ju en personlig grej och det är ju inte säkert att det utvecklaren tycker är logiskt och intuitivt passar för den enskilde individen*”.
- Layout: tex. vilken font man vill ha och i vilket format data ska representeras
Den här punkten tyckte tre av respondenterna kändes irrelevant. Två av dem tyckte däremot att det kunde vara bra. Den ena med motiveringen: ”*Jag vill kunna ändra layout och utseende om man börjar irritera sig efter en tid*”. Den andre såg ett mer praktiskt syfte i det hela och tyckte att det kunde vara värdefullt att ändra på t.ex. datumformat eller om det är namn eller e-postadress som ska representera avsändaren för ett meddelande.

5.3. Sammanfattning av resultat

Vår studie har resulterat i två olika delar. Det första resultatet mynnade ut i en prototyp. På vägen till den fann vi en rad tekniska begränsningar som måste beaktas vid utvecklandet av ett verktyg som ska möjliggöra individuella anpassningar i ett mobilt gränssnitt, eftersom tekniken begränsar vad som kan anpassas. Genom prototypen visar vi dock att det **är tekniskt möjligt att anpassa gränssnittet i en mobil applikation.**

Det andra delresultatet innefattar en sammanfattning av respondenternas svar från intervjuerna. Utifrån denna drar vi vissa slutsatser. Ju högre teknikintresse respondenterna hade desto mer positiv var deras inställning till vår prototyp. Respondenten som hade IT som ett stort intresse även utanför arbetet tycktes vara mest imponerad av idén om att själv kunna anpassa Mobile Organizer. Det är också tydligt att respondenternas användning av Mobile Organizer idag lämnar utrymme för ett verktyg för anpassning. Några av respondenterna uttalade detta på ett klart och tydligt sätt medan vi utifrån andra åsikter från respondenterna själva kunde dra slutsatser om att ett verktyg skulle vara positivt. En respondent uttalade t.ex. att displayen är alldeles för liten för all information som ska visas. Det känns rimligt att tänka sig att respondenten skulle ha nytta av att själv välja vilken information som ska visas på en skärmbild och därmed reducera bekymret.

Trots att alla respondenterna i någon mening var positiva till anpassningsidén var det inte alla anpassningar som kändes relevanta för respondenterna. Det var t.ex. ingen som tyckte att möjligheten till att ändra namn på ett menyval var bra.

Utifrån förutsättningarna för vår studie kan vi säga att **en van användare av en WAP-applikation kan ha nytta av att kunna anpassa sitt gränssnitt så att det passar just hennes behov.**

6. Diskussion

Som vi beskrev i bakgrunden har våra tankar sin grund i den tolkning av begreppet informatik som finns på institutionen för Informatik i Göteborg, vilken fokuserar på användningen av IT. Vi vill sträva efter att placera användaren i centrum och därefter utforma den teknik som ska stödja henne.

Dagens användare av IT blir mer och mer mobila då vårt samhälle i allt större utsträckning byggs upp kring tjänster. Man kan vara mobil på olika sätt vilket beskrevs i underkapitel 3.1 Mobil informatik och det är en bra anledning till varför användaren ska kunna utföra egna anpassningar av gränssnittet på en applikation.

Våra tankar kring betydelsen av möjligheten till anpassning stöds av det vi presenterade i underkapitel 3.2 Anpassning om att alla människor är olika på flera olika sätt, därför är det svårt att ta fram en applikation som passar alla. Människor har olika förutsättningar och preferenser och alla kan inte tillgodoseas av samma saker. Det är därför vårt samhälle är så präglad av mångfald inom alla områden jämfört med t.ex. ett myrsamhälle eller något annat levande system.

Första delen av vår undersökning resulterade i en prototyp av ett verktyg för att låta användaren själv ändra gränssnittet för sin mobila applikation, den delen av resultatet besvarar således delfrågan som löd **"Är det tekniskt möjligt att anpassa gränssnittet i en mobil applikation?"**. Svaret är ja eftersom vi genom vår prototyp har sett att det är möjligt att bygga ett verktyg med hjälp av vilket man kan utföra anpassningar som sedan kan testas i en WAP-telefon. Vi är nöjda med det resultat av prototypen vi fick men det hade givit oss mer om vi hade haft tid att integrera vår prototyp med en verklig applikation.

Vi var till en början väl optimistiska till hur prototypen skulle komma att se ut, vi insåg snart att det tog längre tid att komma in i den nya tekniken än vi först trott. Det har påverkat resultatet så att vi inte kunnat implementera alla de delar vi beskrev i analys och design dokumentet (bilaga A). Om vi hade haft en fylligare prototyp tror vi att vi hade kunnat ge respondenterna vid våra intervjuer en bättre bild av våra idéer och därmed hade vi kunnat få mer trovärdiga svar. Nu tror vi istället att våra tankar kan ha tolkats på olika sätt av olika respondenter eftersom de inte kunde se alla delarna av det resultat vi pratade om i prototypen.

Om vi hade hunnit längre hade vi också kunnat komma fram till fler tekniska begränsningar i telefonen och WAP-specifikationerna.

När vår prototyp stod färdig hade vi skapat oss en större förståelse för vad man skulle kunna utnyttja tekniken till även om vi inte hade tid att förverkliga alla våra idéer.

Resultatet från den andra delen av vår studie syftade till att svara på frågan **"Skulle en van användare av en WAP-applikation ha någon nytta av att kunna anpassa sitt gränssnitt så att det passar just hennes behov?"**. Även den här frågan får ett positivt svar då alla respondenter i vår studie var positivt inställda till anpassningsidén. Vi har således lyckats få ett svar på denna fråga trots att vi inte har kunnat grunda vår andra del av studien på en konfrontation med vana användare. När vi först fick idén att vi skulle bygga en prototyp så ville vi hitta just vana användare av mobila applikationer för att intervjua kring huruvida de önskade anpassningsfunktioner. Det gick dock inte att hitta några sådana användare eftersom det helt enkelt inte fanns några mobila applikationer som användes då vi startade vår studie vilket vi nämnde i inledningen till vår metod i kapitel 4. Om vi hade haft några användare att intervjua redan innan vi byggde prototypen skulle vi kanske ha fokuserat på andra delar i vårt utvecklingsarbete.

Ett resultat av intervjuerna var att vi fick uppfattningen att användarna var mer nöjda med den befintliga applikationen Mobile Organizer än vi hade trott att de skulle vara. Ingen av

respondenterna tyckte t.ex. att de delarna av applikationen som de använde var för krångliga. Det kan dock sägas att den enda funktionen som verkligen användes var email-funktionen och några gav uttryck för att de andra delarna av applikationen var för krångliga att använda av olika anledningar. Det här kan mycket väl innebära att användarna verkar nöjda med hela applikationen trots att de egentligen bara använder en enda funktion, det ger alltså en missvisande bild. Den grundläggande orsaken till att respondenterna tyckte att delar av Mobile Organizer var krånglig var att displayen är för liten och att det går alldeles för långsamt att skriva något med hjälp av sifferknapparna. Dessa båda orsaker ligger inte inom vårt område utan kan placeras in under interaktionskategorin utrustning som vi beskrev i underkapitel 2.2 Teoretisk relevans i bakgrunden. En annan orsak till att respondenterna upplevdes vara så nöjda kan vara att de är den typen av människor som har en hög acceptans för ny teknik, de ansåg att tekniken var omogen och kunde därför ha överseende med eventuella brister eftersom det är en så "het" produkt. Att alla respondenterna själva hade tagit initiativ till att börja använda Mobile Organizer från början visar på ett starkt intresse för produkten.

Det ovan nämnda intresset kan ha varit lite för stort, vi tror att vi kan ha fått fler negativa reaktioner om vi fått tala med användare som använt Mobile Organizer under en något längre period. Det tar ofta ett tag innan man börjar irritera sig på att det inte går att göra på det ena eller andra sättet och vi tror inte att alla respondenterna hade uppnått det stadiet ännu.

De enda tillfällen som Mobile Organizer används är när användarna är ute och reser eller när de befinner sig på en annan plats än kontoret, det överrensstämmer alltså med de båda mobila situationerna "travelling" och "visiting" som vi beskrev i underkapitel 3.1 Mobil informatik. I den tredje typen av situation som var "wandering" alltså när man befinner sig på kontoret men inte vid sin dator var det ingen av respondenterna som uppgav att de använde applikationen. Som vi beskrivit tidigare så är en anledning till att man skulle vilja anpassa sin applikation just att man är mobil på olika sätt och Mobile Organizer kan alltså stödja en användare vid olika typer av mobila situationer.

Vi har läst om och försökt kombinera teorier kring anpassning med teorier kring mobilitet och designprinciper. Utifrån detta har vi själva dragit slutsatsen att det är bra att kunna anpassa gränssnittet när man har en mobil applikation. Även om inte alla respondenterna kände ett starkt personligt behov av ett verktyg för att anpassa gränssnittet så kände vi en positiv inställning från alla respondenterna när vi beskrev vilka möjligheter det skulle innebära för dem som användare. Det är dock viktigt att vara medveten om att alla inte kommer att utnyttja anpassningsmöjligheterna och att man som utvecklare därför måste tänka på att applikationen har bra grundinställningar för gränssnittet.

Vi är inte ensamma om att tänka i de här banorna, mot slutet av vår undersökning hittade vi portalen room33 som hanterar e-post, kalender, kontakter, väder, aktieinformation etc. (room33, 2000). Det finns även en mobil klient till room33 som presenterar ovanstående portal i en WAP-telefon eller på en handdator. Till den här mobila klienten är det tänkt att det ska finnas en anpassningsmöjlighet där man i princip ska kunna välja bort de funktioner man inte vill se i telefonen, grundidén liknar alltså till stor del vår utgångspunkt. Ingen av den funktionalitet som omtalas på webbsidan finns ännu implementerad för den mobila klienten. Det visar på att utvecklingen av den här typen av teknik inte har kommit igång ännu men att det finns ett intresse för den typen av tjänster som den kan erbjuda.

Det är naturligtvis viktigt att anpassningen sker inom ramen för de designprinciper som finns presenterade i underkapitel 3.3 Design av gränssnitt, man kan inte låta användaren ändra på allt för då känner hon snart inte igen sig i sin egna applikation. Några av respondenterna uttryckte specifikt att man inte skulle låta användaren ändra för mycket i menynamn och liknande stora

saker för då skulle det vara svårt för dem att få hjälp, om de skulle behöva ringa en administratör så måste de kunna förklara vad de har problem med så att hon förstår dem.

Flera av respondenterna tryckte på att det måste vara enkelt att använda verktyget om de överhuvud taget skulle orka göra det. En förutsättning måste vara att man kan komma åt det från webben, många av inställningarna behöver inte finnas i telefonen men vissa grundläggande på eller av funktioner måste man alltid ha med sig genom telefonen så att man kan ändra även ute i fält.

En ny idé som två av respondenterna gav oss under intervjuerna var att man skulle kunna utforma ett antal profiler som en administratör hade ansvaret för att lägga upp, vilket även är Malcolm McIlhaggas grundidé som vi beskriver i underkapitel 3.2.2 Automatisk och manuell anpassning.

6.1. Rekommendationer till fortsatt arbete

Vi har haft ca 17 veckor på oss att genomföra vårt examensarbete så det finns många saker vi har stött på som vi skulle vilja fördjupa oss i och som vi i det här läget skulle tycka att det vore intressant att undersöka men som inte har rymts inom ramarna för den här studien.

I takt med att Mobile Organizer blir en riktig produkt som når verkliga kunder skulle det vara intressant att göra en grundlig användarundersökning kring hur den verkligen används ute på marknaden och inte bara i den skyddade miljö som Linq utgör. Det skulle efter en tid vara möjligt att se hur användandet av Mobile Organizer ändras i takt med att användaren lär sig fler finesser, eller inte gör det. En sådan undersökning skulle kunna genomföras så som vi gjort med ett antal djupintervjuer, men den skulle också kunna utformas som en kvantitativt mer omfattande enkätundersökning. Det skulle ge ett bättre underlag för kommande prototyper.

Det skulle också vara intressant att som ett nästa steg i prototypingen integrera en prototyp med en verklig applikation. Den skulle kunna förbättras utifrån det resultat vi fick från våra intervjuer enligt ovan. När det är gjort kan man genomföra en riktig och omfattande test av prototypen.

Ett annat förslag på vad man skulle kunna göra är att backa tillbaks till vår avgränsning och spinna vidare på de trådar som vi lämnade för att kunna koncentrera oss på det vi skulle göra. Exempel på det är att undersöka om det skulle vara möjligt att anpassa andra delar av interaktionen mellan applikation och användare än just gränssnittet, det skulle kunna vara själva den tekniska utrustningen eller applikationen och dess funktioner. Ett annat exempel på en förgrening som vi lämnat därhän är automatisk anpassning, vi har ju enbart fokuserat på manuell anpassning, skillnaden mellan dessa finns beskriven i underkapitel 3.2.2 Automatisk och manuell anpassning.

6.2. Slutsats

Syftet med den här magisteruppsatsen var att svara på frågan "**Kan användaren själv anpassa sitt gränssnitt till WAP-telefonen**" dels utifrån tekniken och dels utifrån användaren. Vår studie har med utgångspunkt i båda dessa aspekter visat att svaret på frågan är ja.

Vår slutsats är att idén med anpassningsbara gränssnitt på mobila enheter skulle vara värdefull för Linq att gräva djupare i. En förutsättning för att det ska bli ett bra verktyg är att det är enkelt att använda, dvs. att det är intuitivt och lättåtkomligt t.ex. via ett webbgränssnitt. Det är också viktigt att man genom en grundlig förundersökning tar reda på vad slutanvändarna verkligen vill anpassa innan man sätter igång och utvecklar funktioner som kanske aldrig kommer att användas.

Enligt vår diskussion är att det dock så att alla inte utnyttjar möjligheterna till anpassning som finns och därför måste grundinställningarna av applikationen vara optimala.

Avslutningsvis vill vi poängtera att anpassningsmöjligheter i rätt sammanhang kan ge nöjdare användare.

7. Referenser

Böcker

- Allwood C. M. (1991). *Människa – datorinteraktion – ett psykologiskt perspektiv*. Lund: Studentlitteratur.
- Allwood C. M. (1998). *Människa – datorinteraktion – ett psykologiskt perspektiv*. Lund: Studentlitteratur.
- Constantine L. & Lockwood L. (1999). *Software for Use – a Practical Guide to the Models and Methods of Usage-Centered Design*. New York: Addison-Wesley, ACM Press.
- Dahlbom B. (1999). The new Informatics. I F. Ljungberg (Red.), *Informatics in the next millennium* (pp. 15-35). Lund: Studentlitteratur.
- Dahlbom B. & Mathiassen L. (1993). *Computers in Context – The Philosophy and Practice of Systems Design*. Oxford: Blackwell.
- Easterby-Smith M., Torpe R. & Lowe A. (1997). *Management Research – An Introduction*. Wilshire: Sage publications Ltd.
- Magoulas T. & Pessi K. (1999). The old and the new informatics. I F. Ljungberg (Red.), *Informatics in the next millennium* (pp. 67-91). Lund: Studentlitteratur.
- Mathiassen L., Munk-Madsen A., Nielsen P. A. & Stage J. (1998). *Objektorienterad analys och design*. Lund: Studentlitteratur.
- Norman D. A. (1988). *The Design of Everyday Things*. New York: Currency and Doubleday.
- Oppermann R. (1994). *Adaptive User Support – Ergonomic Design of Manually and Automatically Adaptable Software*. New Jersey: Lawrence Erlbaum Associates, Inc., Publishers.
- Preece J. et.al. (1994). *Human-Computer Interaction*. Great Britain: Addison Wesley.
- Shneiderman B. (1998). *Designing the User Interface – Strategies for Effective Human-Computer Interaction*, third edition. USA: Addison Wesley.

Artiklar

- Benyon D. & Murray D. (1993). *Developing Adaptive Systems to fit Individual Aptitudes*. Intelligent user interfaces '93, ACM.
- Dahlbom B. & Ljungberg F. (1999). *Mobile Informatics*.
URL: <http://www.viktoria.informatik.gu.se/groups/mi3/>, 2000-05-21.
- Fickas S., Johnsson W. L., Karat J. & Potts C. (1994). *Using scenarios to elicit user requirements*. CHI '94, ACM.
- Kawachiya K. & Ishikawa H. (1998). *New interaction devices and techniques for mobile IT*. CHI '98, ACM.
- Kristoffersen S. & Ljungberg F. (1998), *Representing Modalities in Mobile Computing*. IMC '98, Rostock, Germany.
- Kristoffersen S. & Ljungberg F. (1999). *Mobile informatics: innovation of IT use in the mobile setting*. SIGCHI Bulletin, Vol 31, nr, 1, pp 29 - 34.

Kuutti K., Marcus A., Ferrante J. V., Kinnunen T. & Sparre E. (1998). *Baby faces: User-interface design for small displays*. CHI '98, ACM.

Mackay W. E. (1991). *Triggers and barriers to customizing software*. Human factors in computing systems conference proceedings on reaching through technology '91, ACM.

McIlhagga M. (1998). *Giving Users the Choice between a Picture and a Thousand Words*.
URL: <http://www.dcs.gla.ac.uk/~johnson/papers/mobile/mcilhagga/glasgow.html>, 2000-02-10

Nokia mobile phones (1999). *Style Guide for Service Developers*.
URL: <http://www.nokia.com>

Phone.com Application Development Team (1999). *Developing User Friendly Applications*.
URL: <http://www.phone.com>

Wapforum (1999a). *WAP White Paper*.
URL: http://www.wapforum.com/what/WAP_white_pages.pdf

Internetsidor

Dictionary.com (2000).
URL: <http://www.dictionary.com>, 2000-05-08.

GSM Data Knowledge Site (1999). *GSM Overview*.
URL: <http://www.gsmdata.com/whatsgsm.htm>, 2000-04-26.

Hjelm J. (2000) *Ways to get information with mobile devices*.
URL: <http://www.sto.dfs.se/wap99/Dok/hjelm.zip>, 2000-05-10.

Phone.com (2000).
URL: <http://www.phone.com>, 2000-04-27.

Schwartz P. (1995). *Using Scenarios to Navigate the Future*. Global Business Network
URL: <http://www.gbn.org/Scenarios/Usingsp.html>, 2000-02-21.

Svensk-Engelskt Lexikon (april 2000).
URL: <http://www-lexikon.nada.kth.se/skolverket/sve-eng.html>, 2000-03-27.

Room33 (2000).
URL: <http://www.room33.com>, 2000-04-20.

Wapforum (1999b).
URL: <http://www.wapforum.org>, 2000-05-17.

Whatis?com (1999).
URL: <http://www.whatis.com>, 2000-05-10

Wilkinson L. (1998). *How to Build Scenarios?*
URL: <http://wired.lycos.com/wired/scenarios/build.html>, 2000-02-22

Övrigt

Björk S. (2000) *Re: FW: Litteraturförfrågan!*
Viktoriainstitutet, [bjork@viktoria.informatik.gu.se] 2000-03-28. Privat e-post.

Financial Times, 2000-01-18

Gravas H. (2000) *Re: Hej!*
Göteborgs universitetsbibliotek, [Helen.Gravas@ub.gu.se] 2000-04-03. Privat e-post.

Johansson S., Lind L. & Wahlsten K. (1997). *Är den enskilde användaren tillfredsställd med ett gränssnitt utformat för flera användare?* Serie: Examensarbete/Luleå tekniska universitet, Systemvetenskapliga programmet; 1997:03. Luleå.

Kongress i Stockholm 10/12 1999 *Linq Congress December*

Kongress i Stockholm 17-18/2 2000. *Linq Congress February*

Kongress i Lissabon 5-9/4 2000. *Linq Kick Off Congress April*

Bilagor

A Analys och Design

B Intervjuunderlag

Bilaga A - Analys och design

Uppgiften består i att bygga ett verktyg för en PC som användaren av en mobiltelefon ska kunna använda för att anpassa gränssnittet i mobiltelefonen efter sina egna behov.

Syfte

Syftet med utvecklingen av det här verktyget är att undersöka möjligheten att bygga ett sådant verktyg dvs. är tekniken tillräcklig. Dessutom ska prototypen användas i en intervjusituation i syfte att utvärdera om det går att låta användaren bestämma själv hur hennes gränssnitt ska se ut.

Systemdefinition

Villkor	Systemet ska vara lätt att förstå eftersom det är en service som inte ska ta tid eller kraft i anspråk.
Användningsområde	Yrkesmänniskor som använder sig av LinqPortal-tillämpningar genom mobiltelefonen
Teknologi	PC, Nokia 7110 (mobiltelefon med WAP-stöd)
Objektsystem	Systemet ska läsa en DTD och skapa frågor till användaren utifrån denna. Sedan ska systemet med användarens hjälp utforma en XSL som tillsammans med data ur en XML ska utgöra gränssnittet i mobiltelefonen
Funktionalitet	Systemet ska ställa frågor till användaren om hur hon vill ha sitt gränssnitt och sedan visa i en simulerad mobiltelefon hur det skulle se ut.
Filosofi	Anpassningen ska vara av typen användarinitierad, systemet ska inte ändra sig automatiskt och det ska heller inte gissa vad användaren vill utifrån vad hon gjort tidigare och sedan fråga om det var så hon ville ha det utan det ska bara göra Anpassningar som användaren direkt har bett om.

Dessa förhållanden kan sammanfattas i följande systemdefinition:

Ett system för att erbjuda yrkesverksamma människor som dagligen använder sig av LinqPortal-tillämpningar genom sin mobiltelefon att anpassa gränssnittet efter deras behov.

Systemet, som är av typen servicesystem, ska vara lättillgängligt och inte ta för mycket uppmärksamhet från användarens huvudsakliga aktiviteter.

Verktyget läser den DTD som definierar strukturen för den XML-fil där den data som ska få ett gränssnitt finns. Verktyget skapar sedan en struktur efter vilken den ställer frågor till användaren. Med hjälp av svaren på dessa frågor skapar verktyget en XSL som sedan kan köras mot den ursprungliga XML-filen för att visa det gränssnitt som användaren själv skapat.

Systemet gör inga Anpassningar som inte är initierade av användaren, varken automatiska eller halvautomatiska som kräver bekräftelse

Problemområde

Systemet ska användas för att erbjuda mobila användare till LinqPortal att anpassa gränssnittet efter egna behov.

Systemet ska identifiera en användare och sedan lagra hennes inställningar i en databas.

Alla användare till systemet har en gemensam nämnare och det är att de arbetar på ett företag som använder sig av LinqPortal och eftersom användarna är mobila i sitt arbete använder de de mobila tillämpningarna av denna.

Eftersom mobilt arbete ofta innebär att man har uppmärksamheten på flera aktiviteter samtidigt så

är det betydelsefullt att användaren enkelt kan dels möblera om i nivåerna och dels anpassa hur data visas på varje sida.

Användningsområde (Miljö)

Användaren är mobil i sitt arbete, helt eller delvis, den typiske användaren växlar mellan olika grader av mobilitet och vill därför kunna anpassa sin arbetsmiljö till sitt arbete.

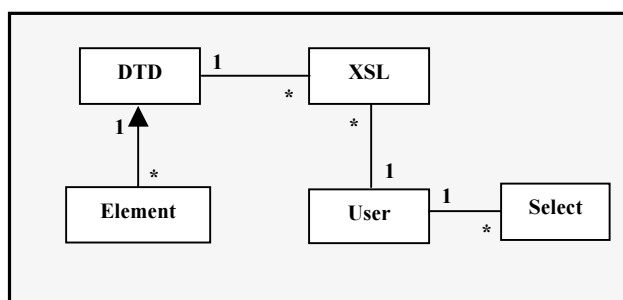
Systemet ska erbjuda servicen att anpassa gränssnittet, det ska därmed vara enkelt att använda så att användaren kan ha sin uppmärksamhet riktad på andra aktiviteter.

Viktigaste funktionerna för användaren:

- Möblera om nivåer
- Välja vilka data som ska visas och hur

Objektmodell

Objekten som vi väljer att ha med i vår objektmodell är hämtade från systemdefinitionen och problemområdet som vi definierat ovan. Vi behöver ett objekt som kan representera en användare och som har möjlighet att spara val som användaren gör under interaktionen med systemet. För att underlätta detta skapar vi ett antal Select-objekt för varje val som användaren gör vilka sedan kopplas till User-objektet. Det måste även finnas ett objekt som representerar en DTD och som har som uppgift att läsa in en befintlig DTD-fil för att sedan kunna återskapa den struktur som finns i DTD:n. För att DTD-klassen ska kunna göra detta på ett effektivt sätt skapar vi ett Element-objekt. Det representerar således ett element i DTD:n, dvs en rad i DTD-filen. Slutligen behöver vi ett objekt som kan sköta genereringen av ett antal XSL-filer som är utdatat från systemet. Det XSL-objektet behöver en koppling till både User-objektet och DTD-objektet.



Objektmodell

DTD

Syfte: Läser en DTD och strukturerar upp den. Den skapar instanser av Element som tilldelas data.

Attribut: *elementArray*
nodElement

Operationer: *readRow:* läser en rad från DTD-filen
setChildren: sätter barnen på det element med vilket funktionen anropas
setParent: sätter förälder till det anropande elementets barn
setLevel: rekursiv funktion som sätter varje elements nivå
getElementArray: returnerar elementArray

XSL

Syfte: Skapar en XSL utifrån DTD-strukturen och användardata som hämtas från gränssnittet.

Attribut: *user*
selectArray

toFile: sträng med XSL-kod

Operationer: *writeMenu*: skapar den första XSL-filen

writeEmail: skapar en XSL-fil som visar ett email

writeInbox: skapar en XSL-fil som visar huvudena till alla email i inboxen

writeDoMenu: skapar en sträng med kod för optionsmenyn

writeToFile: skriver det som finns i *toFile* till en XSL-fil

Element

Syfte: Den representerar ett element i DTD:n

Attribut: *elementName*: anger namnet på elementet

parentName: anger namnet på elementets förälder

childrensName: array som anger namnen på elementets barn

level: anger elementets nivå

Operationer: *getElementName*: returnerar *elementName*

getParentName: returnerar *parentName*

getChildrensName: returnerar *childrensName*

getLevel: returnerar *level*

setElementName: sätter *elementName*

setParentName: sätter *parentName*

setChildrensName: sätter *childrensName*

User

Syfte: Representerar en användare

Attribut: *userName*: anger namnet på användaren

userArray: en array som innehåller de för användaren aktuella Select-objekten

Operationer: *setUserName*: tilldelar *userName* sitt värde

getUserName: returnerar *userName*

setUserArray: tilldelar *userArray* sina värden

getUserArray: returnerar *userArray*

Select

Syfte: Representerar ett val som användaren har gjort

Attribut: *name*: anger namnet på en nivå i DTD:n för vilken en användare har gjort ett val

sellItemsList: en lista som innehåller ett antal av användaren valda element.

Operationer: *setName*: tilldelar *name* sitt värde

getName: returnerar *name*

setSellItemsList: tilldelar *sellItemsList* sina värden

getSellItemsList: returnerar *sellItemsList*

Händelser

Klass	Läs in DTD	Skapa DTD-struktur	Skapa XSL	Skapa anv.fråga	Lagra XSL	Ta emot anv.svar
DTD	X	X		X		
XSL			X		X	
Element		X		X		
User			X			X
Select						X

Användningsmönster

Systemet ska användas för anpassa ett gränssnitt till en mobil applikation. De flesta användningsmönstren ska således tillåta användaren att göra val av olika slag.

Logga in

Användaren anger userID och password och trycker sedan på logga in knappen. Systemet kontrollerar då om användaren är auktoriserad, mot en databas.

Välja menyelement

Användaren väljer vilka element som ska vara med i huvudmenyn och anger i vilken ordning de ska stå där. Systemet sparar användarens val i ett svarsformulär i användarobjektet.

Välja ordning på menyelementen

Användaren kan välja i vilken ordning hon vill att elementen ska visas i menyn. Därmed kan hon lägga funktioner som hon använder ofta mer lättillgängliga än övriga.

Välja vad menyelementen ska döpas till

Användaren kan ändra menyalternativets namn så att det passar hennes uppfattning om alternativet.

Välja vad man vill se på varje skärmbild

Användaren väljer vad hon vilka element hon vill se och inte se på en skärmbild.

Välja hur en skärmbild ska organiseras

Användaren väljer i vilken ordning elementen ska presenteras. Hon kan dessutom välja om hon t.ex. vill visa vissa data i tabellform eller inte.

Välja vilken startsidan ska vara

Användaren väljer vilken sida hon vill komma till först om det inte är huvudmenyn då den är default. Hon får därmed möjlighet att direkt komma till en funktion som hon ofta använder.

Välja vilken typ av grafik som ska visas

Användaren kan slå på eller av grafik.

Välja hur skärmsekvenserna hänger ihop

Användaren kan välja i vilken ordning de olika skärmbilderna ska hänga ihop.

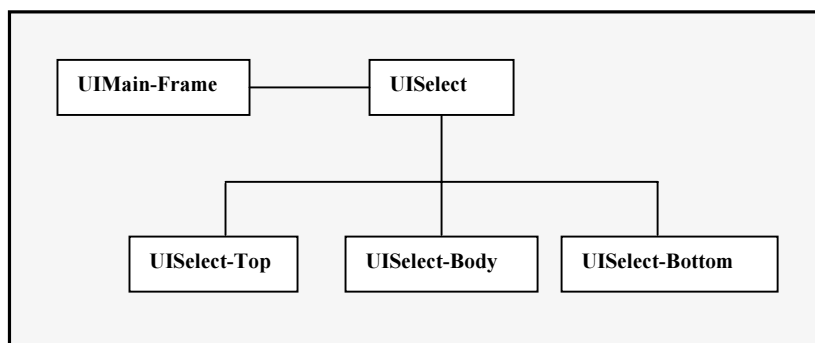
Välja layout

Användaren kan göra individuella inställningar vad gäller layouten. Exempel är möjlighet till att ändra font och färg.

Välja i vilket format data ska representeras

Användaren kan välja vilket format vissa data ska presenteras i. Exempel på olika format för när ett e-postbrev är mottaget är: to 2000-05-04 15:34 eller endast 2000-05-04.

Användargränssnittet



Gränssnittsmodell

UIMainFrame

Syfte: Är av typen Frame och ansvarar för att visa Select-objekt.

Attribut: *uiSelect*: instans av klassen UISelect. Är av typen Container.
dtd: instans av klassen DTD.

Operationer: *Konstruktorn*: Först skapas en ny instans av klassen DTD. Se klassen DTD för att förstå vad det innebär. Därefter anropas den interna metoden showUISelect. Sist anropas metoden settings som även den är intern.
showUISelect: skapar och visar ett nytt UISelect-objekt.
settings: ställer in bakgrund, storlek mm.
quit: avslutar programmet

UISelect

Syfte: UISelect målar upp en sida som består av UISelectTop, UISelectBody och UISelectBottom. Är av typen Container.

Attribut: *sellItemsList*: en lista som innehåller ett antal av användaren valda element.
UISelectTop: en instans av klassen UISelectTop
UISelectBody: en instans av klassen UISelectBody
UISelectBottom: en instans av klassen UISelectBottom
usbArray: en array som innehåller UISelectBody-objekt

Operationer: *paintUISelectTop*: målar upp UISelectTop
paintUISelectBody: målar upp UISelectBody. Därefter sparas UISelectBody-objektet i *usbArray* för att man hela tiden ska kunna hålla reda vilka val en användare gör i de olika UISelectBody-objekten.
paintUISelectBottom: målar upp UISelectBottom
getUsbArray: returnerar *usbArray*

UISelectTop

Syfte: Representerar den övre delen på en UISelect-sida. Skriver ut en överskrift där datat till överskriften är hämtat från *sellItemsList[index]*. Är av typen Panel.

Attribut: *headline*: representerar en rubrik

Operationer: -

UISelectBody

Syfte: Representerar den mellersta delen på en UISelect-sida. Skriver ut två listor där den första innehåller alla element en användare har att välja på vid det aktuella tillfället. Elementen är barn till det element som UISelect-sidan representerar enligt den DTD som lästs in tidigare i programmet. Användaren kan sedan med hjälp av två knappar flytta element över till den andra listan som när användaren är klar kommer att representera de val användaren gjort. Om det skulle vara så att användaren väljer ett element som i sin tur har barn kommer det upp ett nytt UISelectBody-objekt med två nya listor.

Attribut: *list*: en lista som fylls med element ur *dtd*-objektet och som visar vilka element en användare kan välja från.

list2: den lista till vilka de av användaren valda elementen läggs.

sellItemsList: innehåller tidigare val som användaren har gjort.

b1 och *b2*: två knappar som gör att användaren kan göra val mellan de två listorna.

user: en instans av User-objektet dit alla val i slutändan ska sparas.

Operationer: *Konstruktorn*: här fylls den första listan med aktuella data. Om det är första UISelectBody-objektet man målar upp på denna UISelect-sida så är det genom *sellItemsList* man vet vilket element som är det aktuella. Det är således det elementets barn som ska läggas ut i den första listan. Man använder DTD:n för att se vilka barn elementet har. Skulle det vara så att det är det andra eller tredje UISelectBody objektet man målar ut så har det skickats med ett element-objekt som pekar på vilket som är det aktuella elementet och då är det således det

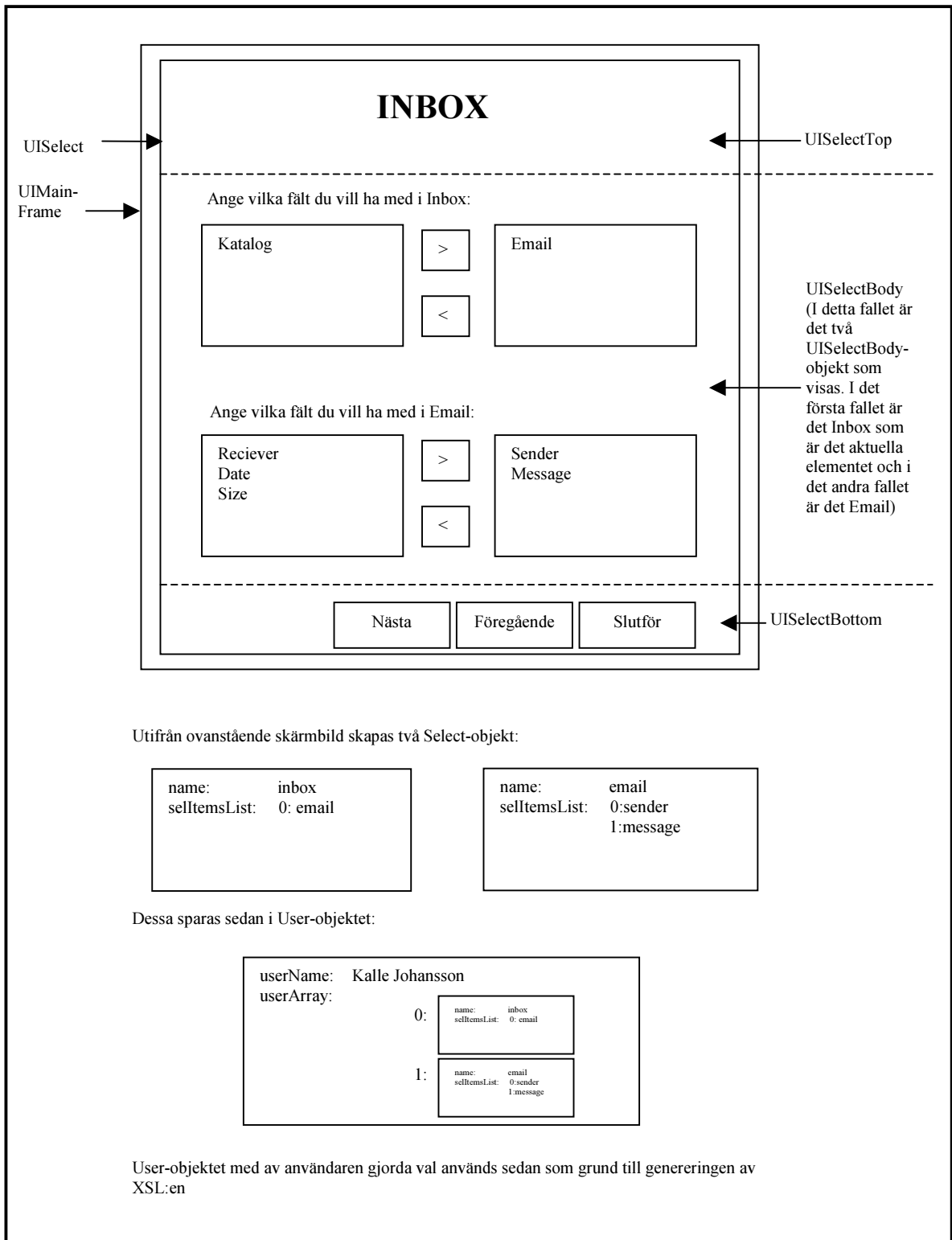
elementets barn som ska läggas ut i den första listan. När en användare har fyllt den andra listan och bekräftat sina val så fylls ett Select-objekt. Detta Select-objekt läggs sedan i User-objektets userArray. Det medför att alla val som en användare gör sparas undan i User-objektet.

UISelectBottom

Syfte: Representerar den undre delen av en UISelect-sida. Innehåller en knapprad med knapparna nästa, föregående och slutför. Knapparna nästa och föregående fungerar som navigeringsknappar mellan de olika skärmbilderna. Slutför avbryter programmet och startar genereringen av XSL:en som är resultatet av alla val användaren gör under programmets gång. Det är sedan den XSL:en som bestämmer hur användarens gränssnitt kommer att se ut i telefonen.

Attribut: -

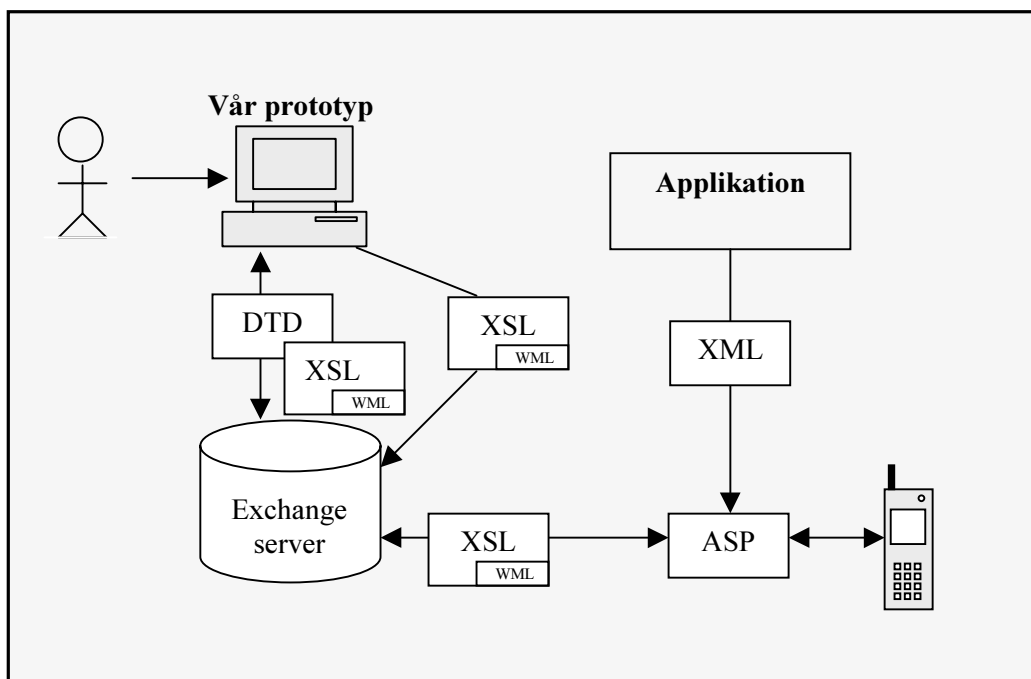
Operationer: -



Användargränssnittet med sina objekt kopplade till objektmodellen

Övergripande arkitektur

Arkitekturen över prototypen och dess omgivning visas i figuren nedan. I figuren representeras vår prototyp av en dator. Prototypen interagerar således med användaren och en Exchangeserver. Genom Exchangeservern får prototypen tillgång till den DTD som ligger till grund för applikationen vars gränssnitt ska anpassas. Dessutom ligger det i Exchangeservern lagrat tidigare inställningar som användaren har gjort eller grundinställningar i form av XSL:er. Med hjälp av DTD:n och XSL:erna skapar prototypen frågor till användaren. Utifrån användarens svar skapas sedan nya XSL:er vilka således representerar hur användaren vill att det nya gränssnittet ska se ut. När användaren sedan använder den mobila applikationen surfar hon med mobiltelefonen till en specifik webbadress som består av en Active Server Page (ASP). ASP:n samordnar vilken datakälla, dvs. vilken XML, det gäller och även vilken XSL som är den första som ska visas. Den XSL som nämns här är den som genererats utifrån prototypen.



Arkitektur över vår prototyp och dess omgivning

Bilaga B - Intervjuunderlag

Bakgrund

ERFARENHET?

1. Vilken bakgrund har du?
2. Hur många år har du jobbat i branschen?

ARBETSSITUATION?

3. Hur ser din arbetsituation ut idag?
 - a. Vilka arbetsuppgifter har du?
 - b. Vad innebär ditt arbete i form av resor, möten utanför kontoret och arbete på kontoret?

TEKNIKINTRESSE?

4. Hur använder du din mobiltelefon, om man bortser från WAP-tjänster?
 - a. Använder du telefonen enbart till att ringa på eller utnyttjar du de finesser som finns tillgängliga, tex. adressbok?
5. Har du dator hemma?
 - a. Använder du datorn till bara arbetsrelaterade uppgifter eller är IT ett intresse även utanför jobbet?

Mobile Organizer

UTGÅNGSPUNKT I DESIGNTEORIER.

ANVÄNDNING?

6. Hur använder du MO?
7. Vilka aktiviteter i MO använder du dig mest av?
 - a. Hur var det första gången du skulle utföra dem, hittade du rätt med en gång, var det som du hade tänkt dig?
8. Tycker du att MO är krånglig att använda?
9. Förstår du alltid var i de olika nivåerna du befinner dig?
10. Upplever du att du behärskar systemet mer efter hand som du arbetar med det?
11. Känner du att du har kontroll över systemet, är det du som styr eller blir du överraskad?
12. Brukar du tänka på vilka konsekvenser ditt agerande ska få när du använder en ny funktion i MO?

UTFORMNING AV PRODUKTEN?

13. Tycker du att beteendet är konsistent med avseende på Microsoft Outlook, meny, terminologi och navigering?
14. Hur upplever du skärmbilderna?
15. Hur upplever du sekvenserna av skärmbilderna, finns det en logisk följd i dem?
16. Är felmeddelandena bra?

OMGIVNING?

17. Hur tycker du att svarstiderna är?
18. Använder du dig av någon av möjligheterna till att göra egna inställningar på MO idag?
 - a. Vet du att det finns möjlighet att göra inställningar från webben?
 - b. Varifrån föredrar du att göra dina inställningar?

19. Hur blev du motiverad att använda MO?

Vår prototyp

*HYPOTETISKA FRÅGOR KRING OM DU HADE ETT SÅDANT VERKTYG!
UTGÅNGSPUNKT I ANPASSNINGSTEORIER.*

1. Tror du att du skulle bli mer nöjd med MO om du hade möjlighet att anpassa den efter din egna önskemål och arbetsuppgifter?
2. Om det fanns möjlighet till anpassning, skulle du vilja vara initiativtagaren till anpassningarna då eller skulle du vilja att systemet var det?
 - a. Om systemet är initiativtagaren skulle du vilja blir tillfrågad innan anpassningen genomförs?
 - b. Skulle du bli irriterad om det inte var du som hade kontroll, jämför med Word, menyerna möbleras om efter vad man senast använt eller använder oftast.
3. Om det fanns möjlighet till anpassning, vilket av följande skulle du vilja kunna anpassa, se listan?
4. Vilka tror du att de utlösande faktorerna för att du skulle börja använda det skulle vara?
5. Vilka faktorer tror du skulle hindra dig från att använda verktyget?