

EXAMENSARBETE

Att anpassa tjänster till WAP

Anders Lövgren, Linda Sonnebro

Data- och systemvetenskap

Institutionen för Industriell ekonomi och samhällsvetenskap
Avdelningen för Data och systemvetenskap

Att anpassa tjänster till WAP

Converting services to WAP

Examensarbete utfört inom ämnesområdet DSV vid Luleå Tekniska Universitet

Av
Anders Lövgren
Linda Sonnebro

Luleå 2000-03-02

Handledare:
Hugo Quisbert, Luleå Tekniska Universitet
Bo Sehlberg, Frontec Norr AB

Förord

Denna uppsats är resultatet av ett examensarbete på 10 poäng som ingår i en filosofie magisterexamen i systemvetenskap vid institutionen för Industriell ekonomi och samhällsvetenskap, avdelningen Data och systemvetenskap, Luleå tekniska universitet.

Vi vill i detta förord passa på att tacka de personer som varit oss till hjälp i vårt arbete. Först och främst ett stort tack till vår handledare Hugo Quisbert för rekommendationer och konstruktiv kritik. Ett stort tack vill vi också rikta till Frontec Norr AB i Luleå, och speciellt Bo Sehlberg som varit vår kontaktperson på företaget. Ett stort tack även till Leif Wikman som visade oss Tur och Retur, och till er andra som hjälpt oss med olika delar av arbetet. Till sist vill vi också tacka våra familjer för stöd och uppmuntran.

Sammanfattning

I takt med att antalet mobiltelefoner har ökat har även efterfrågan på fler tjänster för dessa ökat. Detta är en av anledningarna till att WAP-tekniken skapats och att den har fått så stor uppmärksamhet. WAP, Wireless Application Protocol, är ett kommunikationsprotokoll som gör det möjligt att utveckla tjänster för mobila klienter. Den har utvecklats genom ett samarbete mellan en mängd olika företag och är idag en öppen standard.

För att kunna utnyttja den nya tekniken och anpassa nya tjänster bör man tänka på vissa faktorer såsom begränsningar gällande klienten, säkerhet och gränssnittet mot användaren. Genom att ta hänsyn till dessa begränsningar har vi utformat en metod. Metoden har använts för utveckling av en prototyp och har utvärderats med hjälp av en enkätundersökning. Resultatet blev att metoden till stor del lyckades ta hänsyn till alla faktorer och den kan användas vid utveckling av tjänster till WAP.

Abstract

With the increasing number of cellular phones the demand for value added services has also increased. This is one of the reasons for why the WAP-technology has been created, and that it has received a great amount of attention. WAP, Wireless Application Protocol, is a communications protocol that makes it possible to develop services for mobile clients. It has been developed through the cooperation between several companies and is today an open standard.

To be able to take advantage of this new technology and create services there are a number of parameters to consider about the client, security and the user interface. By considering these parameters we have developed a method. With the use of this method we have created a prototype and evaluated it through a survey. The result of it was that the method had been successful at large and could be used to develop services for WAP.

Innehållsförteckning

1	INTRODUKTION	1
1.1	PROBLEMMOMRÅDE.....	1
1.2	SYFTE	2
1.3	MÅL	2
1.4	VETENSKAPLIG METOD	2
1.5	AVGRÄNSNINGAR.....	3
2	WIRELESS APPLICATION PROTOCOL (WAP).....	4
2.1	INLEDANDE BESKRIVNING.....	4
2.2	VARFÖR WAP?.....	5
2.2.1	<i>Marknaden är annorlunda</i>	5
2.2.2	<i>Nätverket är annorlunda</i>	6
2.2.3	<i>Klienten är annorlunda</i>	6
2.2.4	<i>WAP och användargränssnittet</i>	8
2.2.5	<i>WAP och det trådlösa nätverket</i>	8
2.2.6	<i>WAP erbjuder säker trådlös dataöverföring</i>	9
2.2.7	<i>WAP är optim erat för handburna klienter</i>	9
2.2.8	<i>Möjligheten att använda befintliga utvecklingsverktyg</i>	9
2.2.9	<i>WAP anpassas till nya industristandards</i>	9
2.3	TEKNIKEN	10
2.3.1	<i>Protokollstacken</i>	10
2.3.2	<i>Den optimala överföringstekniken för WAP?</i>	11
2.3.3	<i>WAP-klienten och operativsystemet</i>	12
2.3.4	<i>Wireless Markup Language</i>	12
2.4	ANVÄNDNINGSSOMRÅDEN	14
2.4.1	<i>Idag</i>	14
2.4.2	<i>Imorgon</i>	15
2.5	WAP SOM STANDARD	15
2.6	FRAMTIDSUTSIKTER FÖR WAP	16
2.6.1	<i>Spridningen av WAP över tiden</i>	16
2.6.2	<i>Kritik mot WAP</i>	17
3	MÄNNISKA/DATOR-INTERAKTION.....	19
3.1	INTERAKTIONSSSTILAR	19
3.2	GENERELLA RIKTLINJER	20
3.3	RIKTLINJER FÖR NAVIGERING I EN WEBLÄSARE.....	21
3.4	RIKTLINJER FÖR INTERAKTIVA WEBAPPLIKATIONER.....	22
3.5	PROBLEM VID WEBDESIGN	23
3.5.1	<i>Inkompatibla webbläsare</i>	23
3.5.2	<i>Begränsad bandbredd</i>	24
4	SÄKERHET	25
4.1	KRYPTERING.....	25
4.1.1	<i>Diffie-Hellman</i>	25
4.1.2	<i>RSA</i>	25
4.2	AUTENTICERING	26
4.2.1	<i>Digital Signatur</i>	26
4.2.2	<i>SIM-kort</i>	26
4.3	DATAINTEGRITET	27
5	SYSTEMINTEGRATION	28
5.1	KLIENT/SERVER	28
5.2	ATT FLYTTA TILL KLIENT/SERVER	28
5.3	SYSTEMLÖSNINGAR FÖR WAP	29

6	ATT UTVECKLA FÖR WAP.....	31
6.1	KRAVSPECIFIKATION FÖR EN WAP-TJÄNST	31
6.1.1	<i>Vad tjänsten skall utföra och med vilket resultat.....</i>	<i>31</i>
6.1.2	<i>Säkerhetskrav</i>	<i>31</i>
6.1.3	<i>Prestandakrav</i>	<i>32</i>
6.2	TIPS FÖR WAP-UTVECKLING.....	32
6.2.1	<i>Variabelprefix</i>	<i>32</i>
6.2.2	<i>Begränsat minnesutrymme.....</i>	<i>32</i>
6.2.3	<i>Vanliga fel.....</i>	<i>33</i>
6.3	UTVECKLINGSVERKTYG FÖR WAP	33
7	EN METOD FÖR ATT ANPASSA APPLIKATIONER TILL WAP	35
7.1	VAD ÄR MÖJLIGT MED WAP?	35
7.2	ÖVERSIKT AV ANPASSNINGSMETODEN	35
7.3	ATT SKAPA EN KRAVSPECIFIKATION	36
7.4	RIKTLINJER FÖR DESIGN.....	36
7.5	IMPLEMENTATION.....	38
8	PROTOTYP AV EN WAP-TJÄNST	39
8.1	VAL AV UTVECKLINGSVERKTYG	39
8.2	TUR OCH RETUR	39
8.3	KRAV.....	39
8.4	TILLÄMPNING AV METODEN	40
8.5	ENKÄT.....	43
9	SLUTSATSER.....	45
9.1	EGNA REFLEKTIONER	45
10	EFTERORD	47
10.1	METODDISKUSSION	47
10.2	FORTSÄTTA STUDIER	47
11	REFERENSER.....	49
12	FÖRKORTNINGAR OCH ORDFÖRKLARING.....	51
BILAGA A. MOBILE STATION APPLICATION EXECUTION ENVIRONMENT (MEXE)		
BILAGA B. UTVECKLINGSMILJÖER		
BILAGA C. ENKÄT		

1 Introduktion

Mobiltelefonen har gått från att vara en relativt sällsynt statussymbol till att verkligen ha slagit igenom på den breda konsumentmarknaden. I takt med att antalet användare ökar har också efterfrågan på att kunna utföra fler tjänster med hjälp av mobiltelefonen ökat. Via mobilsvaret och textmeddelanden rör vi oss idag mot möjligheten att använda telefonen som ett gränssnitt mot kraftfullare servrar och därmed också mer avancerade tjänster. De dataöverföringsmöjligheter som finns idag är väldigt begränsade och därför har en ny teknik växt fram, den så kallade WAP-tekniken. WAP står för Wireless Application Protocol och är en öppen standard som till stora delar bygger på internetteknik och internetstandarder. Med hjälp av WAP-tekniken kan en mobil klient i form av en mobiltelefon eller en handdator kommunicera med servrar som är kopplade antingen direkt till internet eller till ett intranät.

Huvuddelarna i WAP-specifikationen är:

- En modell för WAP-applikationer som baseras på den befintliga modellen för WWW-applikationer. Detta innebär flera fördelar för utvecklare, såsom en bekant modell för applikationer, en testad arkitektur och möjligheten att använda befintliga verktyg som webbservrar, XML-redigerare med mera. Optimering och anpassning har skett för att klara av den mobila miljön. När det har varit möjligt har man använt befintliga standarder eller använt dessa som en startpunkt.
- Ett beskrivningsspråk som följer XML-standarderna. WML och WMLScript kräver inte ett vanligt (QWERTY) tangentbord eller en mus för att mata in information. Till skillnad från ett HTML-dokument som har en platt struktur är ett WML-dokument uppdelat i olika enheter som kallas ”cards”, och applikationer består i sin tur av ett antal cards som användaren navigerar mellan. WML består av olika ”tags” på samma sätt som HTML men är anpassat till handburna klienter.
- En specifikation för en mikrobrowser i den handburna klienten vilken fungerar på samma sätt som en webbrowser för traditionella webapplikationer. Den här specifikationen visar hur WML och WMLScript ska tolkas av klienten och presenteras för användaren.
- En protokollstack med uppgift att minimera behovet av bandbredd för att passa handburna klienter.

[WAP001] [PHON00] [WAPF99]

1.1 Problemmråde

WAP-tekniken utvecklades med hänsyn till de begränsningar som finns i form av ett kommunikationsmedium med bland annat låg bandbredd samt mobiltelefonens begränsade resurser. En begränsning med den här tekniken är att användargränssnittet är väldigt enkelt och att telefonen skiljer sig markant från en vanlig persondator.

Det finns idag inga metoder eller modeller för att utveckla WAP-applikationer, och det finns också en mängd befintliga applikationer som kommer att behöva anpassas till WAP-tekniken. För att detta ska utföras på ett effektivt sätt krävs någon form av metod, både för rena webbsidor men också för mer traditionella applikationer. Det är också här det blir intressant för oss som systemvetare att titta närmare på den här tekniken. I denna uppsats kommer vi att ta fram en metod för att anpassa befintliga applikationer till denna teknik. Arbetet kommer att ske i samarbete med Frontec Norr AB här i Luleå. I vårt arbete kommer vi att dra paralleller till

webutveckling eftersom WAP-tekniken har en del likheter med webtekniken, men också till traditionell klient/server-utveckling. Vi kommer att utveckla en WAP-prototyp mot en applikation som redan finns i både webmiljö och klient/server-miljö.

1.2 Syfte

Syftet är att utreda:

- Vilka typer av applikationer som kan anpassas till WAP-tekniken
- Hur dessa applikationer kan och bör anpassas
- Varför det kan vara intressant att anpassa applikationer till WAP-teknik

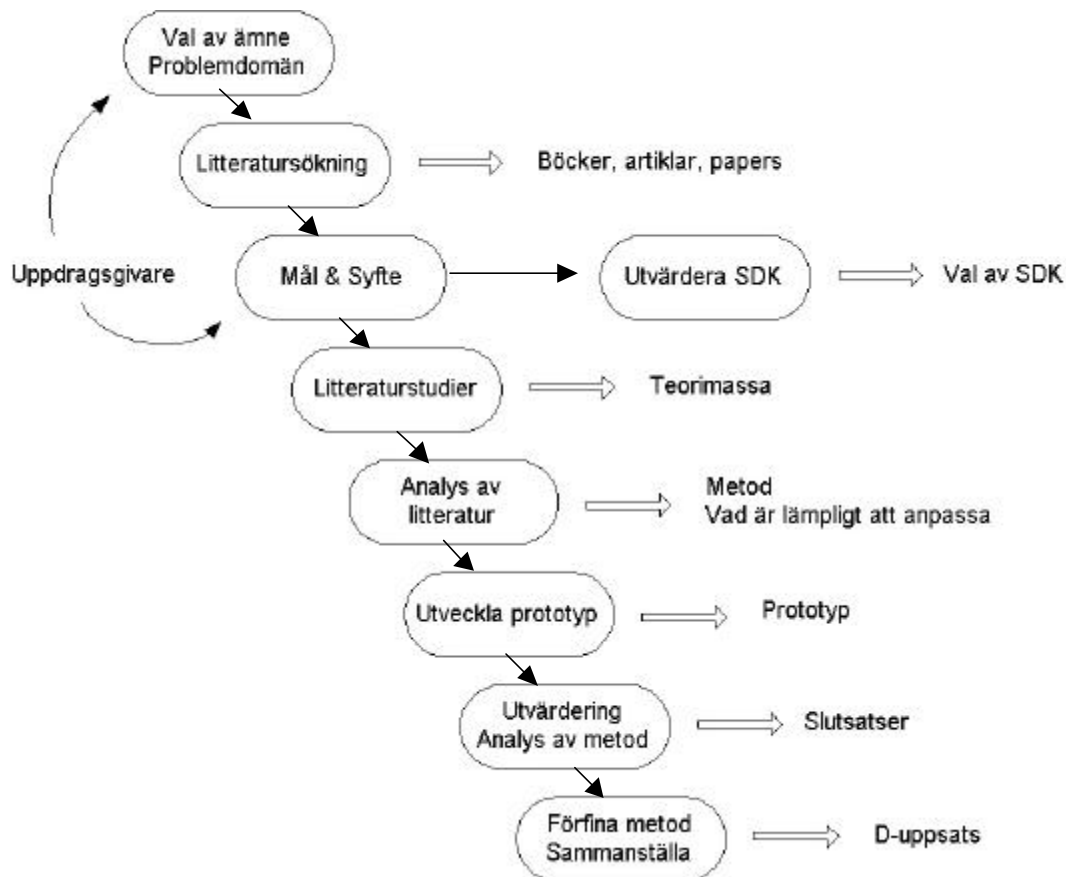
1.3 Mål

Målet med uppsatsen är att ta fram en metod för anpassning av applikationer till de möjligheter som ges med WAP.

1.4 Vetenskaplig metod

Det första steget mot en D-uppsats är att välja ett ämne som känns intressant för författarna, och som kan vara av vikt för det systemvetenskapliga ämnet. Vi valde ämnet mobil datoranvändning eftersom vi anser att det uppfyller båda dessa kriterier. Det känns intressant för oss, och det är ett relativt nytt ämne inom vårt område vilket medför att det finns flera spännande infallsvinklar. Nästa steg är att fokusera på en viss problemomän, och där finna ett eller flera specifika problem att studera. Vi valde WAP-tekniken och fick kontakt med en uppdragsgivare som hade ett specifikt önskemål inom detta område, nämligen att ta fram en metod för att anpassa befintliga applikationer till WAP-teknik. Utifrån det önskemålet och våra egna tankar formulerade vi syfte och mål med uppsatsarbetet. Därefter gjorde vi en del litteratursökning vilket resulterade i ett antal böcker och artiklar om WAP som vi snabbt läste igenom för att få en grov bild av ämnet. Efter en del diskussioner angående det vi hittills lärt oss om WAP kom vi fram till att det kunde vara tänkbart att använda en del befintliga tekniker inom mjukvarudesign på detta nya område. Det framgick av litteraturen om WAP att denna teknik till mångt och mycket bygger på den protokollstack som används på internet, men också att man kan se WAP som en sorts middleware. Vår förhoppning blev att kunna dra nytta av riktlinjer för webdesign och klient/server-utveckling.

Ytterligare en litteratursökning gav oss en rik flora av böcker inom dessa två områden. Efter att ha gått igenom denna litteratur valde vi ut ett antal böcker som kändes relevanta, såsom "Developing Cooperative and Client Server Systems" [FORG94], "Human-Computer Interaction" [PREE94] och "User-Interface Design" [COX93]. Vi fortsatte också våra studier av litteratur som direkt anknyter till WAP-tekniken. Eftersom ett av önskemålen från vår uppdragsgivare var att utveckla en prototyp började vi också titta på de olika utvecklingsmiljöer som finns tillgängliga. Nästa steg i arbetet blev att analysera den litteratur vi valt och utforma en metod för att anpassa befintliga applikationer till WAP. Vi gick raskt vidare och utvecklade en prototyp enligt denna metod. För att utvärdera resultatet lät vi göra en enkätundersökning för att avgöra om metoden bidragit till att ta fram en bra prototyp. Till detta lade vi våra egna erfarenheter av utvecklingsarbetet och gjorde en del tillägg till metoden. Utifrån den analys som gjorts av både litteratur, utvecklingsarbete och resultatet av metoden har vi dragit ett antal slutsatser. En översiktsskild av vår vetenskapliga metod presenteras i Figur 1.



Figur 1. Vetenskaplig metod

1.5 Avgränsningar

Vi förutsätter att läsaren av denna uppsats har grundläggande kännedom om TCP/IP-protokollet samt om HTTP och HTML, därför kommer vi inte att beskriva dessa tekniker på ett ingående sätt. Vi förutsätter vidare en viss kännedom om databasbaserade webapplikationer och klient/serverapplikationer, därför kommer vi bara att ta upp det som är relevant för att dra paralleller till WAP-tekniken inom dessa områden. Läsaren bör också känna till OSI-modellen.

WTAI är en del av WAP-specifikationen som används för att utveckla avancerade telefoni-tjänster, vilket inte ingår i vårt ämnesområde. Därför tar vi inte upp detta i uppsatsen.

De olika transportmedia som används för dataöverföring från telefon till de olika serverna är givetvis en intressant del av WAP-tekniken, men är ett så omfattande ämne att det gör sig bäst i en egen uppsats. Därför tar vi inte upp detta i någon större utsträckning.

Vi kommer att ta upp vilka olika klienter som kan användas för att utnyttja WAP-tjänster, men inte i någon större utsträckning.

Vid utveckling av en applikation fortsätter arbetet efter implementationen med testning samt drift och underhåll. Eftersom vår huvudsakliga arbetsuppgift inte varit att utveckla en fullt funktionsduglig applikation har vi avbrutit arbetet efter implementationen och istället gjort en utvärdering av prototypen med hjälp av en enkät.

2 Wireless Application Protocol (WAP)

I det här kapitlet kommer vi att beskriva WAP-tekniken och dess olika beståndsdelar.

2.1 Inledande beskrivning

Vad är egentligen WAP? WAP är förslag till standard för användande av internet-kommunikation och avancerade tjänster inom telefoni för digitala mobiltelefoner, person-sökare, handdatorer och andra trådlösa produkter. WAP har utvecklats av många olika anledningar. En av anledningarna är att användningen av mobiltelefoner har ökat kraftigt. I och med den stora ökningen av användandet ser företagen stora möjligheter att utveckla nya tjänster och dessa kan vara allt ifrån att betala räkningar, köpa biljetter och kontrollera e-post.

En annan anledning till att WAP tekniken utvecklas är att den hårda konkurrensen på marknaden ständigt ökar och för att överleva måste företagen utveckla nya affärsområden och kunna erbjuda sina kunder det senaste inom teknologin.

På senare tid har efterfrågan på mobilitet ökat kraftigt. Det betyder att kunder, företagspartners och anställda vill kunna få tag på information från företaget när de behöver den var de än befinner sig. För att kunna genomföra detta krävdes en ny teknik – WAP.

[NOK001] [CA0399] [WAP001] [WAP002]

WAP är egentligen bara ett protokoll – ett standardiserat sätt för en mobiltelefon att utbyta information med en server i ett nätverk.

WAP har fått så stor uppmärksamhet av ett flertal orsaker:

- Det innebär en standard som kopplar ihop internet med mobiltelefoner, och därmed två av de mest expansiva marknaderna i världen.
- Flera tunga företag inom mobiltelefonindustrin har ställt sig bakom WAP, bland annat Nokia, Ericsson och Motorola.
- Wap Forum som driver utvecklingen av standarden har mer än 120 företag som medlemmar.
- Mobila informationstjänster har inte fått det genombrott som många nätverksoperatörer hoppats på, och WAP ses som en möjlighet till detta genombrott.

[MOBI99]

De första företagen som tog initiativet till tekniken var Ericsson, Nokia, Motorola och Unwired Planet (numera Phone.com). De bildade 1997 WAP Forum. Nokia, Ericsson och Motorola är stora företag inom telefonibranschen medan Unwired Planet hade utvecklat något som de alla var intresserade av, nämligen en mikrobrowser – det vill säga, en webbläsare för mobiltelefoner. Det primära målet för WAP Forum var att samla ihop företag från olika områden med intresse för trådlös kommunikation för att få en gemensam integration och operabilitet och ökad tillväxt av trådlös kommunikation. I slutet av februari 1999 hölls GSM World Congress i staden Cannes. Det var då som WAP fick sitt stora genombrott genom att alla stora telekomföretag kom överens om att anta den som standard.

[NOK001] [CA0399] [WAP001] [WAP002]

Det finns också en del smolk i glädjebägaren:

- Microsoft har varit tveksam att stödja WAP.
- En del företag anser att det är bättre att använda de protokoll som är standard för internet istället för att utveckla ett nytt protokoll för den mobila världen.
- Det finns få mobiltelefoner som stödjer WAP och det kan dröja länge innan det finns stöd för WAP i en större del av de mobiltelefoner som finns på marknaden.
- WAP är inte färdigutvecklat, och viktiga delar som Push (att aktivt skicka information till en mobil klient) och stöd för telefonitjänster är ännu inte standardiserade.
- Nya tekniker, som MExE, ger större möjligheter än WAP.

[MOBI99]

2.2 Varför WAP?

De uppenbara begränsningar som finns i form av klienter med begränsad kapacitet samt dyrköpt och relativt långsam dataöverföring skiljer applikationer för mobila enheter markant från traditionella applikationer, men det finns också en del fundamentala skillnader som är viktiga att förstå.

2.2.1 Marknaden är annorlunda

Applikationer för handburna klienter är annorlunda i jämförelse med applikationer för stationära datorer och även för vanliga bärbara datorer på så sätt att användaren har andra behov och förväntningar. Några av dessa är:

- Lätt att använda – På senare år har det blivit mycket enkelt att använda en dator bland annat på grund av ett enkelt grafiskt gränssnitt. En handburen klient måste vara ännu lättare att använda på grund av att dessa klienter kan komma att användas av personer som inte har någon större datorvana, och som dessutom kan utföra andra uppgifter samtidigt. Användarna kommer inte att vara fokuserade på klienten på samma sätt som vid en stationär dator. Applikationerna måste därför ha bästa tänkbara gränssnitt. Det får inte förekomma installationsskript, komplicerade menystrukturer, oförståeliga felmeddelanden eller svåra tangentkombinationer såsom ctrl-alt-del.
- Marknadens storlek – Den befintliga storleken och den troliga tillväxten på marknaden för mobiltelefoni är väldigt stor. Enligt Global Mobile Magazine finns det mer än 200 miljoner användare idag, och enligt Nokia kommer antalet användare att passera en miljard under år 2005. Den trådlösa marknaden kommer att kräva, och ha råd att kräva, optimala lösningar.
- Priskänslighet – Idag börjar det finnas datorer på marknaden som kostar under 10 000 kr, men trots detta spelar inte en prisskillnad på 500 kr så stor roll vid valet av datormärke. På marknaden för mobiltelefoner spelar däremot 500 kr en större roll, speciellt efter subventioner från mobiltelefonoperatörer. Marknadsundersökningar har visat att mobiltelefoner måste ha ett pris under 1500 kr för att konkurrera effektivt på den breda konsumentmarknaden. Utökade tjänster i form av applikationer måste vara kostnad -seffektiva för att slå igenom på den marknaden.

- Användning av applikationerna – Användarna förväntar sig att de utökade tjänsterna ska fungera likadant som nuvarande funktioner i deras klient. Tjänsten ska vara tillgänglig direkt, lätt att använda och vara färdig på ett par minuter. Små timglas som uppmanar användaren att vänta kommer inte att accepteras.
- Väsentliga uppgifter – Så snart man kliver ut från kontoret förändras behovet av information. Användare av handburna klienter kommer inte att vilja surfa på nätet med sin mobiltelefon. De vill ha små, specifika tjänster som går relativt snabbt att utföra. Att bläddra igenom rubrikerna på e-posten istället för att läsa vartenda meddelande, eller bara ta emot de aktiekurser som är intressanta. Att ta emot trafikinformation i realtid är betydligt intressantare för någon som rör sig ute i trafiken än för någon som sitter vid sin dator på kontoret. De bästa applikationerna kommer att ge användaren en utförlig, personlig summering av viktig information och dessutom ge möjligheten att ta emot utförligare information om det som känns extra intressant.

[WAP001] [PHON00]

2.2.2 Nätverket är annorlunda

Det trådlösa nätverk som utgör grunden för mobil datorkommunikation har en del begränsningar i jämförelse med trådbundna nätverk. Beroende på fundamentala begränsningar i strömförsörjning, täckningsmönster och mobilitet har de trådlösa nätverken:

- Lägre bandbredd
- Längre fördröjning
- Mindre stabila uppkopplingar
- Mindre förutsägbar tillgänglighet

Trådlösa applikationer måste övervinna dessa begränsningar, och ändå ge användaren tillfredsställande prestanda.

[WAP001] [PHON00]

2.2.3 Klienten är annorlunda

Förutom de begränsningar i trådlösa nätverk finns det en del begränsningar i själva klienten:

- Mindre kraftfull processor
- Mindre minne (ROM och RAM)
- Begränsad strömförsörjning
- Liten display
- Annorlunda inmatning (telefonens knappsats)

På grund av dessa begränsningar måste användargränssnittet i den handburna klienten vara helt annorlunda i jämförelse med traditionella grafiska gränssnitt för persondatorer. Det här är inte heller nåt som kommer att förändras i någon större utsträckning under den närmaste framtiden. Dagens handburna klienter har skapats för att vara just handburna och ett konsumentkrav är att batteritiden ska vara så lång som möjligt vilket alltid kommer att begränsa bandbredd och strömförbrukningen hos processor, minne och display.

Eftersom det under en överskådlig framtid kommer att finnas ett gap mellan prestanda hos de bästa datorerna och de bästa handburna klienterna måste överföringen av data till dessa klienter överbrygga detta gap. I och med att skillnaden mellan de olika klienterna förändras över tiden är det viktigt att standards för dataöverföring anpassas till dessa skillnader såväl som till marknadens behov. Figur 2 visar en översikt av klientens begränsningar.

[WAP001] [PHON00]



Figur 2. Klientens begränsningar

2.2.4 WAP och användargränssnittet

WAP-tekniken innehåller ett kraftfullt och funktionellt användargränssnitt som är anpassat för handburna klienter. Användare navigerar mellan skärmbilder med hjälp av upp- och nertangenter istället för en mus. Funktionstangenter (soft keys) kan användas för specifika operationer beroende på applikationens kontext eller för att utföra olika menyval. Telefonens knappsats används för att mata in alfanumeriska tecken. Navigeringsfunktioner som Back, Home och Bookmark används för att dra nytta av standardwebläsarnas navigation. Genom att använda den befintliga internetmodellen som en startpunkt kan användare som har erfarenhet av webben snabbt känna igen sig. Det ger också ett användargränssnitt som är lätt att lära in, och som är intuitivt för nya användare. Microbrowsern ger också klienter med större display och mer funktionalitet möjligheten att visa mer innehåll på skärmen, precis som en traditionell webläsare när man maximerar fönstret på skärmen. Vi kommer att gå in djupare på gränssnittet i kapitel 3.

[WAP001] [PHON00]

2.2.5 WAP och det trådlösa nätverket

Protokollstacken som specificeras i WAP optimerar standardprotokoll från webben, till exempel HTTP, för att vara effektivt i trådlösa nätverk med låg bandbredd och längre fördröjning. Förbättringar i sessions-, transaktions-, säkerhets- och transportskiktet syftar till att ge bättre prestanda i trådlösa nätverk. Exempel på dessa förbättringar:

- Texthuvudet i HTTP har översatts till binärt format vilket medför att mindre data måste skickas över det trådlösa nätverket.
- Sessionernas återuppkoppling möjliggör att sessioner pausas och återupptas utan att sessionens overhead måste skickas igen. Detta medför att sessioner kan pausas när så behövs för att frigöra överföringskapacitet och spara batterikapacitet.
- WAP innehåller ett transaktionsprotokoll (WTP) som erbjuder pålitlig datagramservice, ungefär som traditionell TCP, men är bättre anpassat för trådlösa nätverk. Eftersom det bara finns en väg för paketen att förmedlas mellan klienten och WAP-proxyn behövs inte lika avancerad transaktionshantering som i TCP, och WTP reducerar därför utbytet av information för varje transaktion.
- WTP innebär också att en TCP-stack inte behövs i telefonen och på så sätt frigörs både minne och beräkningskapacitet i den handburna klienten.
- WAP-protokollet innebär att betydligt mindre bandbredd behövs än i den traditionella HTTP/TCP/IP-stacken, vilket är nödvändigt för att utnyttja den begränsade bandbredden i de trådlösa nätverken på bästa sätt.

[WAP001] [PHON00]

2.2.6 WAP erbjuder säker trådlös dataöverföring

Många webapplikationer använder idag säker överföring mellan klienten och applikations-servern. WAP-tekniken innehåller ett säkert protokoll för transaktioner via handburna klienter. WTLS-protokollet är baserat på industristandarden TLS, och är optimerat för dataöverföring med liten bandbredd. WTLS hanterar dataintegritet, autentisering och skydd mot denial-of-service attacker.

[WAP001] [PHON00]

2.2.7 WAP är optimerat för handburna klienter

WAP-tekniken innehåller en tunn klient i form av en microbrowser, speciellt anpassad för att klara av det begränsade minnet i en handburen klient. Användningen av proxy-teknik och komprimering i nätverket minskar kravet på processorkapacitet. Detta bidrar till att minska strömförbrukningen och öka batteriets driftstid.

[WAP001] [PHON00]

2.2.8 Möjligheten att använda befintliga utvecklingsverktyg

Webutvecklare kommer relativt enkelt att kunna utveckla WAP-applikationer eftersom programmeringsmodellen för WAP är nära besläktad med utvecklingsmodellen för WWW. WML är ett språk definierat som en dokumenttyp i XML, och därför kan utvecklare använda befintliga XML-verktyg, och även många utvecklingsmiljöer för HTML, för att utveckla WML-applikationer.

Eftersom WAP-tekniken använder HTTP 1.1 för att kommunicera mellan webbservrar och WAP-gateways kan utvecklare implementera sina applikationer på vanliga webbservrar. WML-utvecklare kan använda verktyg som Cold Fusion, CGI, Perl, ASP och liknande för att skapa dynamiska WML-applikationer. Även om det är möjligt att automatiskt översätta HTML till WML är det i praktiken bättre att anpassa applikationerna till WML för att skraddarsy gränssnittet till den handburna klienten. Den viktigaste delen av en webapplikation är oftast det innehåll som presenteras och interaktionen med en databas, inte den specifika HTML-koden som används för att presentera innehållet. Detta innebär att utvecklare kan återanvända delar av befintliga webapplikationer och anpassa presentationen till WML.

[WAP001] [PHON00]

2.2.9 WAP anpassas till nya industristandarder

WAP-tekniken optimerar och utökar befintliga internetstandarder så långt det är möjligt. WAP Forum har tagit element från TCP/IP, HTTP och XML, anpassat dessa till handburna klienter och återsänt dessa specifikationer till W3C som förslag till integrering med nästa generation av HTML och HTTP. WAP Forum kommer att fortsätta utveckla WAP-specifikationen för att passa till nya tekniker. Precis som med internetprotokollen är WAP uppdelat i olika skikt beroende på den funktionalitet som ska utföras, så att varje skikt kan utvecklas oberoende av de övriga. Skikt på en lägre nivå kan ersättas för att använda nya kommunikationsmedia utan att behöva bygga om skikt på en högre nivå. På så sätt är WAP flexibelt och kan användas även med framtida trådlösa kommunikationsprotokoll.

[WAP001] [PHON00]

2.3 Tekniken

WAP använder sig av klient/server-teknik. En relativt enkel microbrowser implementeras i den mobila klienten, vilket inte kräver alltför stora resurser på klientsidan. Detta innebär att WAP lämpar sig för mobiltelefoner och andra tunna klienter. Tjänsterna som användaren utnyttjar finns lagrade på en server istället för att lagras permanent i klienten. Filosofin bakom WAP-tekniken är att ställa små krav på klientens resurser och istället utnyttja funktionaliteten i nätverket på ett bra sätt. WAP anses vara ett omfattande och skalbart protokoll utvecklat för att användas med:

- Olika sorters mobiltelefoner, med en enkel display eller mer avancerade möjligheter.
- Olika sorters trådlösa tjänster som SMS, CSD, USSD och GPRS. WAP ska ge applikationsutvecklare och nätverksoperatörer möjlighet att erbjuda tjänster på olika typer av nätverk, bärarmedia och klienter. WAP separerar de olika protokollskikten vilket ger möjlighet att migrera tjänster från till exempel SMS eller CSD till GPRS.
- Olika standarder för mobila nät såsom CDMA, GSM eller UMTS. WAP är utvecklat för att fungera med alla standarder som används för mobiltelefoni.

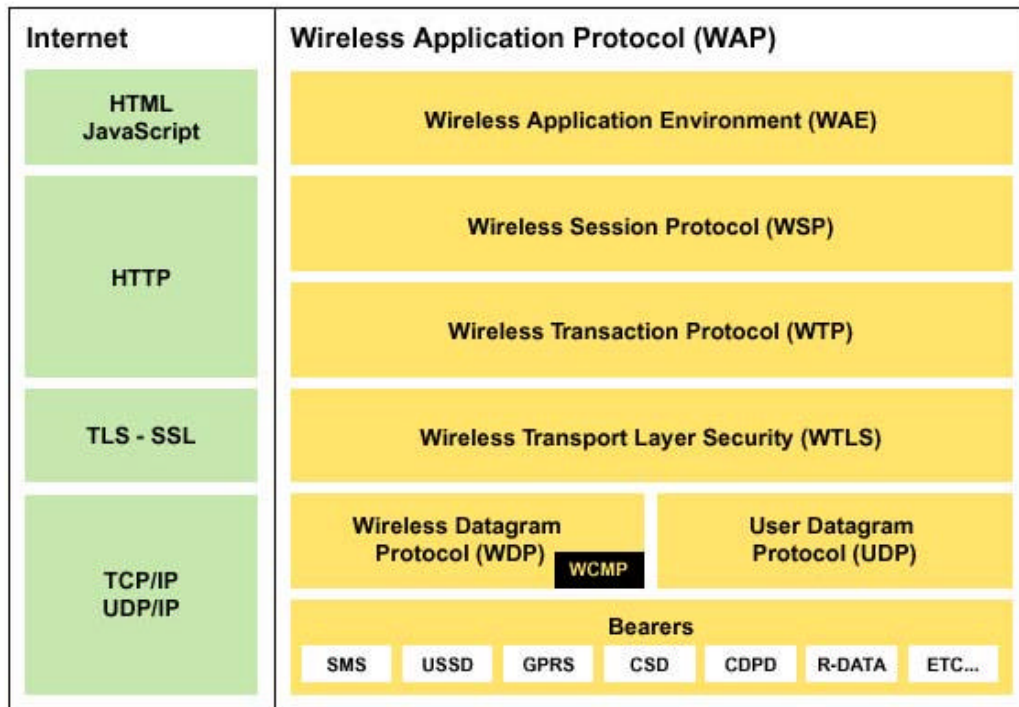
[WAPF99]

2.3.1 Protokollstacken

WAP består av en protokollstack som är uppdelad i olika skikt.

- Wireless Application Environment (WAE) – Definierar användargränssnittet på den mobila klienten. Består av hypertextspråket WML, skriptspråket WML-Script och ett gränssnitt mot telefonitjänster som heter WTA. Dessa verktyg används för att utveckla WAP-tjänster.
- Wireless Session Protocol (WSP) – Ett mellanskikt som länkar WAE till två olika sessionslager, ett kopplingsorienterat protokoll som opererar ovanpå WTP och ett kopplingslöst som opererar ovanpå WDP.
- Wireless Transaction Protocol (WTP) – Opererar ovanpå en datagramtjänst, som till exempel UDP, för att erbjuda transaktionshantering anpassad för trådlösa nätverk med låg bandbredd.
- Wireless Transport Layer Security (WTLS) – Erbjuder säkerhetsfunktioner som baseras på standardprotokollet TLS. Inkluderar integritetskontroller, autentisering och kryptering.
- Wireless Datagram Protocol (WDP) – Utgör kopplingen mot överföringsmedia, och gör WAP oberoende av vilket media som används.

Figur 3 visar en jämförelse mellan de protokoll som används på internet och WAP.



Figur 3. The WAP architecture [AUWP99]

2.3.2 Den optimala överföringstekniken för WAP?

Som framgår av inledningen till detta avsnitt kan WAP använda sig av olika överföringstekniker, varav några är:

- Short Message Service (SMS) – Har en begränsning i form av 160 tecken per meddelande, vilket gör att SMS inte är en lämplig teknik för WAP. Den header som WAP-protokollet kräver skulle göra att till och med den enklaste transaktion kräver ett flertal meddelanden. Att använda SMS som överföringsteknik kan därför bli en tidskrävande och kostsam upplevelse.
- Circuit Switched Data (CSD) – De flesta WAP-tjänster som har utvecklats i dagsläget använder CSD som överföringsmedia. CSD innebär att en uppkoppling som kan ta upp till 30 sekunder måste utföras för att klienten ska få kontakt med en server.
- Unstructured Supplementary Services Data (USSD) – Sessionsorienterat överföringsmedia, vilket innebär att en session etableras och uppkopplingen finns kvar tills dess att användaren, tjänsten eller en time out frigör denna.
- General Packet Radio Service (GPRS) – Paketförmedlande överföringsmedia som introduceras i många GSM- och TDMA-nät från år 2000 och framåt. Det är en ny och spännande teknik eftersom den är ögonblicklig (ingen väntetid vid uppkoppling), relativt snabb (upp till 177,2 kbps i bästa tänkbara teoretiska miljö) och tillåter virtuell uppkoppling, vilket innebär att information skickas från nätverket i och med att den genereras.

Eftersom alla utom de tidigaste WAP-klienterna troligtvis kommer att stödja GPRS kan dessa två tekniker ge en synergieffekt och tillsammans få en bred spridning. För de interaktiva, menybaserade tjänster som WAP ska erbjuda räcker inte CSD till på grund av den långa uppkopplingstiden. SMS å andra sidan är ögonblicklig, men är alltid ”store and forward” vilket innebär att även om användaren av tjänsten har begärt information från microbrowsern i klienten måste förfrågan gå via SMS. GPRS erbjuder fördelar för WAP i jämförelse med de övriga alternativen.

[MOBI99]

WAP-specifikationen är skriven för att lätt kunna anpassas till framtida överföringsmedia som till exempel GPRS, Edge och tredje generationens trådlösa nätverk. Som i OSI-modellen finns i WAP möjligheten att byta ut de nedersta lagren i protokollstacken beroende på vilket överföringsmedium som används.

[WAP002]

2.3.3 WAP-klienten och operativsystemet

WAP-tekniken finns eller kommer att finnas för handburna klienter i form av mobiltelefoner, PDA's, smartphones, communicators m.m. I dagsläget handlar WAP mest om mobiltelefoner men utvecklingen av minnen och processorer kan medföra en bredare flora av klienter som kan dra nytta av WAP-tekniken.

WAP är ett kommunikationsprotokoll och en programmeringsmodell för applikationer och kan implementeras i nästan alla operativsystem som till exempel PalmOS, EPOC, Windows CE och JavaOS. WAP ska också erbjuda interoperabilitet mellan olika operativsystem.

[WAP002]

WAP finns till exempel redan implementerat i form av WAPman som är en microbrowser för Palm-datorer. WAPman stöder i dagsläget version 1.1 av WAP-specifikationen.

[EDGE00]

2.3.4 Wireless Markup Language

Wireless Markup Language är det språk som används i WAP-konceptet för att konstruera de grafiska och textuella gränssnitten. WML används även för så kallad händelsehantering, det vill säga för att hantera händelser orsakade av användaren. Det kan handla om att en viss typ av knapp har tryckts ned eller att användaren har gått tillbaks för att titta på något.

Tidsstyrning kan också programmeras med WML och språket har även stöd för att sätta och läsa av variabler. Typer existerar i WML. De deklarerar dock inte, vilket ligger helt i linje med att hålla komplexiteten nere för att språket skall kunna implementeras på små maskiner med begränsad cpu-kraft och begränsat minne.

WML bygger på en modell av kort och en kortlek eller som de engelska beteckningarna lyder; ”cards” och ”deck”. Ett ”deck” består av en mängd kort som användaren navigerar mellan. Ett kort kan vara ett antal inställningar som logiskt hör ihop. WAP och WML gör inga antaganden om exakt hur mycket som ryms på en skärm, inte heller om hur informationen ser ut i

detalj när den presenteras. Enligt samma princip som det besläktade språket HTML beskriver WML bara vad som presenteras och matas in samt ger riktlinjer för hur detta skall göras.

Ett WML-dokument innehåller likt HTML ett antal element vars så kallade "taggar" omslutes av "<" och ">". Mellan elementen förekommer text och referenser till bilder som presenteras av den mobila enheten. Ett element kan ha en start-tag och en matchande slut-tag. Ett WML-dokument inleds med elementet WML och dess start-tag <wml> som avslutas med det matchande </wml>. Vissa element har en styrande funktion och förekommer därför utan någon matchande tagg. Ett exempel är
 som används för att bryta en specifik rad och påbörja en ny rad.

Mellan <wml> och </wml> specificeras, med elementet card, ett antal kort där varje kort anges med start-taggen <card> och den matchande slut-taggen </card>. Denna mängd kort, som vi skulle kunna kalla kortlek, kan ha en viss grundfunktionalitet som dock går att ändra. Detta specificeras med elementet "template" som består av start-taggen <template> och den matchande slut-taggen </template>. Varje kort kan presentera text och bilder. För detta finns ett antal element där några känns igen från HTML. De vanligaste bildformaten kan överföras och ett nytt introduceras också, kallat WBMP, som för närvarande stödjer endast monokroma bilder. De mobila klienter som kan visa grafik måste stödja detta format; WBMP typ 0. Precis som i HTML används elementet för att lägga in bilder, men till skillnad från HTML är attributet "alt" obligatoriskt. Attributet specificerar en alternativ text som visas om den mobila enheten inte har möjlighet att visa grafik.

WMLScript är ett programspråk som kan användas tillsammans med WML för att ge en högre grad av logik åt WML-sidorna. Variabler, hopp och en del möjligheter till styrning existerar visserligen redan i WML men med WMLScript kan man höja graden av logik och komplexitet betydligt. Till exempel kan man i WMLScript använda sig av funktioner, iterationer och villkor.

Man bör dock ha klart för sig att WMLScript precis som WML är framtaget för att i första hand exekvera på små mobila enheter med mycket begränsad processorkraft och lite minne. Följaktligen bör kodmängderna inte bli för omfattande och programmeringsstilen också anpassas därefter.

WMLScript är ett procedurellt språk vilket innebär att ett program är uppbyggt kring ett antal procedurer. Alla procedurer i WMLScript returnerar alltid ett värde och om man inte specificerar något returvärde returneras en tom sträng som returvärde till den funktion som gjorde anropet. En procedur som alltid returnerar ett värde brukar kallas funktion så procedurerna i WMLScript är egentligen funktioner. Ett program exekveras genom ett anrop till en funktion som använder argumenten för att räkna ut diverse saker och eventuellt anropa andra funktioner och så vidare.

Till skillnad från en av WMLScripts släktingar, JavaScript, så kan WMLScript skickas i en komprimerad form över nätet. Källkoden kompileras dels för att den skall kunna skickas snabbt över nätet men också för att koden skall ges en representation som är någorlunda enkel att tolkas av en liten och billig mobil enhet med begränsad processorkraft. Servern kan lagra skriptet färdigkompilerat, det vill säga i binärformat, i filsystemet eller i en databas. Svarstiderna kan på detta sätt kortas ned.

[HILL99] [WAPF99]

2.4 Användningsområden

WAP-tekniken är idag inte så beprövad och använd. Detta kan bero på att tekniken fortfarande håller på att utvecklas och att de nya WAP-telefonerna ännu inte fått sitt stora genombrott på marknaden. Det finns dock en del tjänster som redan är utvecklade och anpassade för den nya tekniken och det finns många nya tjänster som håller på att utvecklas.

Frågan är vilka typer av tjänster man kan förvänta sig med den nya WAP-tekniken? Enligt WAP-Forum ska det inte finnas några större begränsningar för vad man ska kunna göra med WAP.

Ett framtidsscenario från WAP-Forum:

”Tänk dig att du går ut ifrån en kontorsbyggnad för att åka till flygplatsen, du använder din WAP anslutna trådlösa mobila enhet för att kolla hur mycket trafik det är samtidigt som man kollar tidtabellen för tåget. Det visar sig var mycket trafik så man köper en tågbiljett istället för att köra bil. På väg till flygplatsen bokar man en stol i planet och reserverar en speciell maträtt. Sedan packar man upp regnkläder efter att ha kollat vädret hos slutdestinationen”.

Andra typer av tjänster som man kan använda WAP-teknologin till är bland annat:

Kundservice och kundbeställningar, meddelande tjänst och samtalshantering, e-post, ta fram kartor och kunna lokalisera var man befinner sig, få information om väder och trafikvarningar, läsa nyheter och sportresultat, kunna göra transaktioner med hjälp av elektronisk handel och även göra bankärenden.

Många ser stora möjligheter med vad man kan erbjuda för tjänster med den nya tekniken. Dessutom ger WAP, som har hela den trådlösa industrin bakom sig, möjligheter för utvecklare att utveckla, implementera och stödja applikationer för trådlösa mobila enheter. Vid slutet av år 2000 tros antalet enheter vara cirka 100 miljoner världen över. Detta kommer mest troligt att resultera i bra möjligheter för utvecklare att öka sina intäkter betydligt.

Nedan följer några exempel på vad som finns idag och vad som kommer inom snar framtid.

2.4.1 Idag

WAP-butik

PC Superstore är en av de första WAP-butikerna . De säljer bland annat produkter för bärbara datorer och mobiltelefoner. De har även en tjänst som informerar om fler tjänster och om WAP.

[CS10299]

Hotellbokning

Scandic Hotels stamkunder erbjuds ett komplett paket med wap -telefoner som konFigurerats för direktbokning av rum. Med hjälp av telefonen kan kunden direkt få besked om sin status i Scandics klubbsystem, vilka hotell som finns i den aktuella staden och om det finns några lediga rum.

[CS600]

Styrning och processövervakning via en WAP-telefon.

En serviceingenjör kan titta på flera olika delar i en process, som till exempel att titta på status och styra utrustning som pumpar och ventiler.

[CS10299]

WAP för godssökning.

Transportföretaget DHL har utvecklat en WAP-tjänst som innebär att deras kunder kan gå in och få information om varifrån ett gods har skickats, var det befinner sig nu och när det beräknas anlända till sin slutdestination. Tjänsten riktar sig främst till kunder som ofta befinner sig på rörlig fot som exempelvis reparatörer, läkare och servicetekniker. Fler funktioner kommer dock att läggas till och göra det möjligt för övriga kunder att nyttja tjänsten.

[CS9899]

2.4.2 Imorgon

Internet Cellular Smart Access 3.5 (ICSA) – är ett program som kommer att vara anpassad för mobiltelefonwebbläsare. Genom denna programvara och WAP ska man bland annat kunna komma åt e-post som har lagrats på olika pop3-konton och även e-post konton som finns på html-baserade webbplatser. Den ska fungera med olika typer av gsm-telefoner och även med handdatorer som är baserade på CE eller som använder Outlook.

[CS9599]

Meritanordbanken tänker erbjuda sina kunder att sköta sina bankärenden via WAP.

Mobile Services for Domino 1.0 är ett program som gör det möjligt att sända och ta emot e-post, röstmeddelanden och fax via mobiltelefonen och trådlösa terminaler. Den har stöd för WAP tekniken och kommer att fortsätta utvecklas inom det området.

[CS10299]

2.5 WAP som standard

Fram till idag har det varit nödvändigt att utveckla flera versioner av en och samma mobiltelefon-tjänst så att användare med telefoner från olika leverantörer har kunnat använda tjänsten. Därför är det viktigt för mobiltelefonindustrin att ta fram en standard så att en och samma version av en tjänst kan användas av de flesta ägare av en mobiltelefon.

WAP är dock ännu inte kommersiellt gångbar, och det är därför för tidigt att säga om WAP kommer att bli den standard som slår igenom. Det är inte ens säkert att ett protokoll kommer att bli standard för hela industrin, eftersom utvecklingen går så fort kommer det troligtvis alltid att finnas både ny och gammal teknik på marknaden. Det är därför viktigt för utvecklare och nätverksoperatörer att inte binda upp sig alltför hårt till ett enda protokoll.

Dessa protokoll är dock tekniker som söker en tillämpning. Det finns i dagsläget ingen tjänst som verkar tilltala majoriteten av mobiltelefonanvändare på så sätt att de bara måste ha den tjänsten. De tjänster som erbjuds är till exempel aktieinformation, e-post, kalenderfunktioner och liknande – ingenting nytt egentligen.

Ett tänkbart sätt att nå en bredare kommersiell spridning är kombinationen av dessa tjänster som kan marknadsföras som ”den mobila livsstilen” eller nåt liknande. Tekniken ska inte ses som ett mål i sig utan som ett medel där målet främst är att ge ett bredare stöd för generella informationstjänster i mobiltelefoner.

[MOBI99]

2.6 Framtidsutsikter för WAP

Intresseområden för framtida utveckling är förbättrad säkerhet, gränssnitt för smarta kort, hantering av betalningar och push-teknik. Ett annat intressant område är möjligheten att erbjuda multimediatjänster för handburna klienter. WAP 1.1 och 1.2 är öppna protokoll som redan idag erbjuder möjligheten att skicka olika former av multimedia till handburna klienter. Vissa tjänster, som till exempel streaming media, kräver dock en anpassning av WAP-protokollet.

Man kan fråga sig om WAP kommer att behövas när nya överföringsmedia möjliggör bredbandskommunikation till handburna klienter. Även om bandbredden ökar kommer inte kostnaden för denna bandbredd att minska till noll. Kostnaderna kommer dels i form av avgifter för kommunikationen men också i form av ökad strömförbrukning hos klienterna och ökad belastning på nätverket. WAP är anpassat till de handburna klienternas begränsningar, och dessa kommer inte att försvinna i och med att bandbredden ökar. Utvecklingen av applikationer för handburna klienter kan också väntas medföra ökat behov av bandbredd, vilket gör att det fortfarande är intressant att optimera användandet av den tillgängliga bandbredden.

Det främsta målet med WAP-forum är att driva utvecklingen av en global standard för trådlös kommunikation till och från handburna klienter. WAP-forum samarbetar med andra standardiseringsorgan för att undvika konflikter mellan olika standarder. WAP-forum består av en mängd olika företag med intresse i handburna klienter och trådlös kommunikation, och har hittills lyckats undvika konkurrerande standarder och den fragmentering av marknaden som detta skulle innebära. Flera olika standards skulle innebära inkompatibla klienter, och att hårdvaruleverantörerna antingen måste bygga redundanta klienter eller enbart rikta in sig på en del av marknaden.

[WAP002]

2.6.1 Spridningen av WAP över tiden

En del analytiker förutspår hundratals miljoner WAP-klienter redan inom ett par år, medan andra är mer försiktiga i sina prognoser. En prognos som hör till de mer försiktiga visas i tabell 1. Andelen telefoner med stöd för respektive teknik anges procentuellt i förhållande till det totala antalet mobiltelefoner globalt.

År	Telefoner med stöd för SMS	WAP-telefoner	Telefoner med stöd för GPRS
1999	85%	0%	0%
2000	60%	5%	0%
2001	50%	15%	5%
2002	35%	35%	15%
2003	20%	60%	35%
2004	20%	65%	60%

Tabell 1. Prognos över WAP-klienter [MOBI99]

På grund av den långsamma introduktionen av WAP-telefoner kommer tekniken inte att nå någon större spridning innan år 2001 och inte att nå den stora massan förrän alla telefoner som säljs stödjer WAP omkring år 2002. Därefter kommer antalet WAP-telefoner i bruk att öka och nå sin höjdpunkt omkring år 2004 då drygt två tredjedelar av mobiltelefonerna i omlopp stödjer WAP. Samtidigt kommer nya tekniker som MExE (se bilaga A) att introduceras som succesivt kan förväntas ersätta WAP.

[MOBI99]

2.6.2 Kritik mot WAP

Det pratas just nu väldigt mycket om WAP i teknikvärlden, alla fördelar den kommer att ge, vilka användningsområden som är möjliga och hur stor spridning den kommer att få i framtiden. Men det har även börjat publicerats kritiska artiklar av olika personer som inte tror på WAP. De anser att WAP-tekniken inom snar framtid kommer att vara helt slut och andra standarder kommer att ta över på den trådlösa marknaden. En anledning till detta kan just vara att WAP inte klarar av bredbandsaccess vilket gör den begränsad. Kritikerna menar också att det mest är frågan om en ”andrahandteknik” till Ericssons nya teknik Blue Tooth.

Än så läge finns endast ett fåtal telefoner ute på marknaden och när de börjar säljas i någon större utsträckning vet ingen. När de väl släpps på marknaden kommer priset förmodligen att vara rätt så högt vilket gör att spridningen på telefonerna och användningen av tekniken kommer att dröja. Det finns en hel del andra faktorer som också spelar in vid kritiken av WAP. Dessa beskrivs nedan.

Det pratas hemskt mycket i media om den nya WAP-revolutionen och att man ska kunna använd WAP-tekniken till att surfa på Internet med mobiltelefoner. Problemet med att surfa på Internet är att många webbsidor innehåller en hel del bilder vilket WAP, än så länge, inte klarar av. Det finns två lösningar för detta – den första är att man gör speciella wapsidor skrivna i WML eller att man skriver all information som man vill förmedla med XML och sedan har en konverterare som kan konvertera till WML och HTML. Det finns dock en del saker som man bör ha i åtanke, vem vill sitta och surfa på Internet med en sådan liten display som mobiltelefonen har? Uppkopplingstiderna kan bli väldigt långa eftersom den har begränsad bandbredd, vilket gör att det kan bli rätt dyrt om man vill surfa och leta efter saker en längre tid. Det krävs dessutom att de batterier som används till mobiltelefoner idag förbättras avsevärt – de skulle inte räckta länge om man skulle försöka surfa med dem. WAP är främst gjort för att användas till interaktiva tjänster, inte surfning på Internet.

Vad gäller användarvänligheten är gränssnittet med WAP-tekniken väldigt begränsad. Det beror på att man till att börja med har väldigt liten display (skärm) på en mobiltelefon.

Man rymmer endast fyra rader text vilket gör att man inte kommer att kunna tillhandahålla särskilt mycket information åt gången. Att skriva text med en mobiltelefon är inte särskilt smidigt eftersom det tar lång tid att få fram rätt bokstav eller tecken vilket gör att man blir tvungen att använda rullmenyer så långt som möjligt. Detta medför dock att man snabbt tappat greppet om vilken sida man är på och var man kom ifrån. Än så länge kan man inte heller använda färger eller grafik i form av bilder vilket gör att helhetsintrycket av skärmbilden blir rätt primitivt.

[CS10799] [CS9299] [CS11399] [WIRE00] [BRG99]

En av de stora stötestenarna som kritikerna framför är att telefonerna inte går att uppgradera till nya versioner av WAP-specifikationen. Den som har en WAP 1.1-telefon kommer att kunna använda en 1.2-funktion men inte de funktioner som är typiska för WAP 1.2. Exempelvis går det inte att identifiera telefoner som använder WAP 1.1. Tillverkarna av telefoner jobbar på att ta fram en lösning som gör det möjligt att uppgradera telefonerna. En uppgradering via telenätet vore det bästa men till att börja med kommer uppgraderingen att kräva ingrepp av en tekniker.

[CS300]

3 Människa/Dator-Interaktion

I det här kapitlet kommer vi att titta på interaktionen mellan människa och dator. Vi kommer att visa på skillnader och likheter mellan en klient för WAP och en traditionell dator. Eftersom WML har stora likheter med HTML kommer vi att ta upp olika riktlinjer för webdesign och tillämpa dem på WAP-design.

3.1 Interaktionsstilar

Vid användandet av en traditionell dator sker inmatning främst med hjälp av tangentbord och mus, och utmatning via en bildskärm. Musen och tangentbordet tillsammans med ett grafiskt användargränssnitt gör datorn till ett flexibelt verktyg. Den vanligaste klienten för WAP-tjänster är en mobiltelefon, som är relativt begränsad i jämförelse med en dator. Den har ingen mus utan använder istället navigationstangenter för att ge användaren möjlighet att navigera mellan olika val. Inmatning av alfanumeriska tecken sker med hjälp av en knappsats som skiljer sig från det traditionella tangentbordet på så sätt att varje knapp används till att mata in ett flertal tecken beroende på antalet gånger en knapp trycks in. Utmatning sker på en display som skiljer sig markant från en bildskärm. Telefonens display kan bara visa ett fåtal rader text och primitiva bilder, till skillnad från en bildskärm som kan visa en stor mängd text och bilder med hög upplösning och en mängd färger.

En WAP-tjänst kan inte byggas på samma sätt som en applikation för en persondator utan påminner mer om en websida, eller ett formulär, beroende på hur den utformas. WML-språket ger möjlighet att skapa ett gränssnitt som använder sig av menyer, dialoger och hyperlänkar. Det finns en del riktlinjer för formulärbaserade gränssnitt som kan vara användbara vid utvecklandet av WAP-tjänster.

- Meningsfulla titlar – undvik en teknisk jargong.
- Tydliga instruktioner – använd ett lätt språk
- Logisk gruppering och sekvensiering av fält – fält som hör ihop bör grupperas tillsammans, fälten bör ha en logisk ordning.
- Visuellt tilltalande layout – var uppmärksam på fältens placering på skärmen.
- Meningsfulla beskrivningar av fält – använd ett allmänt språk.
- Konsistent terminologi och konsistenta förkortningar.
- Ge användaren en indikation på vilka inmatningar som förväntas.
- Enkel navigering mellan fält – använd tab eller piltangenter.
- Felkorrigerig för enstaka fält – enkelt att radera och korrigera inmatningar.
- Felmeddelande för otillåtna inmatningar – felmeddelande ska visas när det felaktiga värdet fyllts i och tillåtna värden ska visas i felmeddelandet.

- Icke obligatoriska fält ska vara tydligt markerade – Obligatoriska fält ska visas före de icke obligatoriska.
- Förklaring för fält ska placeras på ett konsistent sätt.
- Tydlig markering när ett formulär är ifyllt.

[PREE94]

I en traditionell kontorsmiljö finns användarens primära arbetsuppgifter i datorn. Orsaken är enkel: kontorsarbete handlar till stor del om att bearbeta dokument, kalkylark, e-post och liknande, och dessa dokument finns i datorn. I ett mobilt scenario är situationen en helt annan. Underhållsarbetare, försäljare och annan mobil personal har i allmänhet inte sina primära arbetsuppgifter i datorn. Deras arbetsuppgifter är istället utanför datorn, till exempel att möta kunder, reparera trasig utrustning och så vidare.

Många av dagens problem inom mobil datoranvändning härrör från det faktum att gränssnitten för mobila klienter har utvecklats enligt samma principer som bordsdatorer. Gränssnitten på bordsdatorer handlar om direkt manipulation, med filer och foldrar, dra-och-släpp, och så vidare. Direkt manipulering är en interaktionsstil som ställer stora krav på användarens uppmärksamhet. Detta kan vara lämpligt i en kontorsmiljö, men det är inte alltid den bästa interaktionsstilen för en mobil klient. Orsaken är att direkt manipulering är för exklusiv för det mobila arbetet: det ställer för stora krav på användarens uppmärksamhet, istället för att låta användaren rikta sin uppmärksamhet på sina primära arbetsuppgifter.

[KRIS99]

3.2 Generella riktlinjer

Generella riktlinjer ger en del goda råd om hur gränssnittet mellan människa och dator bör utformas, men det ställs också krav på de personer som ska utforma systemet. Man skapar inte ett bra användargränssnitt genom att tillämpa ett antal riktlinjer, men riktlinjerna kan vara en hjälp i designarbetet.

- Konsistens – Ett konsistent gränssnitt gör att användaren känner igen liknande situationer och kan dra lärdom av detta i en ny applikation.
- Användaren ska kontrollera applikationen, inte tvärtom. Ett exempel är när användaren ska mata in ett datum och det måste vara i ett visst format som dag/månad/år. Istället kan användaren mata in enbart en siffra som representerar dagen och sedan fyller applikationen själv i innevarande månad och år, eller tolkar ett tomt fält som dagens datum.
- Använd användarens språk – Användare och utvecklare har ofta olika språkbruk. En RAM används för att hänga upp bilder och en PORT spärrar vägen. Det är viktigt att tänka på vilket språk de förväntade användarna har och inte använda terminologi som kan vara okänd för dem.
- Motivation – Gränssnittet ska motivera användarna. Ett dåligt gränssnitt gör helt enkelt att applikationen inte används.

- Feedback – Användaren ska få respons på utförda handlingar. Antingen i form av ett svar på en inmatning eller en upplysning om att applikationen utför något som kan ta lite tid.
- Val istället för inmatning – Det är lättare för användaren att välja mellan ett antal olika alternativ istället för att mata in kommandon eller information.
- Enkelhet – Det ska vara enkelt för användaren att använda applikationen. Eftersom människor skiljer sig åt kan det som är enkelt för en person vara svårt för en annan, och därför är det viktigt att komma ihåg vilken typ av användare applikationen förväntas ha.
- Förlåtande – Tillåt användaren att göra fel eller ändra sig. De flesta applikationer bör ha en ångra-funktion. Om användaren kan ångra sig blir applikationen lättare att använda och lättare att lära sig.
- Felhantering – Om någon form av fel inträffar är det viktigt att ge användaren en indikation om detta, men det är minst lika viktigt att användaren kan tolka felmeddelandet och förstå vad som gick fel och hur detta kan åtgärdas. Undvik därför kryptiska felmeddelanden.
- Avlasta minnet – Tänk på det mänskliga minnets begränsningar. En applikation ska inte kräva att användaren måste komma ihåg en mängd olika saker. I den mån det är möjligt bör applikationen förse användaren med vettiga default-värden, eller ge möjligheten att välja från en lista istället för att mata in information.

[COX93]

- Reducera den kognitiva belastningen på användarna. Användare ska inte behöva komma ihåg stora mängder detaljer för att kunna använda en applikation.
- Ta hänsyn till fel. En vanlig ursäkt är att problem uppstår på grund av mänskliga fel. Men människor kommer alltid att göra fel ibland, och måste göra fel för att lära sig saker. Att ta hänsyn till fel innebär att göra det svårare för användaren att göra fel, att ge felmeddelanden som förklarar vad som gått fel och tillåta användaren att själv korrigera ett misstag.
- Var konsistent och tydlig. Konsistens kan innebära att följa någon standard för design, eller att använda metaforer som hjälper användaren att forma en mental modell av systemet. En tydlig design utgår ofta från information som samlats in från de presumtiva användarna.

[PREE94]

3.3 Riktlinjer för navigering i en webbläsare

Vid navigering i en webapplikation finns det en del riktlinjer som bör följas:

- Avgör varför en användare besöker er website. Vilka är de förväntade sidor som möter användaren första gången, och vilka sidor möter användaren in nan han/hon lämnar siten? Vilket syfte har användaren med sitt besök?

- Lär er vilken kunskapsnivå era användare har. Följer de länkarna på det sätt som ni har tänkt er? Har de alternativa vägar att nå informationen? Är det vana användare med stor erfarenhet av att navigera på webben?
- Skapa en design som ger användaren medvetenhet om var han/hon befinner sig i eran site. Är användaren alltid medveten om var han/hon befinner sig? På vilken nivå de befinner sig? Kan de relatera den information de ser för tillfället med var de kommer ifrån och vad de söker?
- Använd tydliga rubriker eller dylikt för att underlätta återbesök. Använder er website tydliga rubriker eller dylikt på ett konstruktivt sätt? Använder ni informativa titlar på de olika sidorna? Kom ihåg att varje sida kan bli ett bokmärke i en webläsare, och därmed en ingångspunkt för ett återbesök.
- Använd tydliga ledtrådar. Har ni skapat distinktiva ikoner som visar användaren var de befinner sig i er website? Eller tvärtom, använder ni standardikoner som redan har en viss betydelse för användaren och därmed ger tydliga ledtrådar? Är alla funktioner, och delar av siten, tydligt markerade och lätta att hitta?
- Använd den typ av ledtrådar som användaren förväntar sig. Använder ni ikoner, och rubriker på länkar, på ett konsistent sätt i hela websiten? Skapa en konsistent känsla, och ett konsistent utseende. Användarna kan komma in i websiten genom i stort sett vilken sida som helst, så gör det lätt för dem att navigera mellan sidorna.
- Avgör om den information ni vill sprida är strukturerad eller ostrukturerad. Om användaren förväntar sig en strukturerad site bör denne mötas av en tydlig struktur i alla delar av siten. Om informationen är ostrukturerad måste det ändå gå att navigera den på ett lättförståeligt sätt.
- Utnyttja användarens förmåga till problemlösning för att skapa ett sätt att navigera en ostrukturerad website. Kan användaren förutse vilken information som kommer att möta han/hon under en rubrik? Om användaren tappar tråden, ger ni dem då möjlighet att hitta rätt?
- Pröva i en verklig miljö. För att navigera en website krävs det två saker för att miljön ska kallas verklig: ett antal tänkbara användare, och ett antal tänkbara datorsystem. Kan användare ur er målgrupp hitta siten, och navigera på ett bra sätt? Klarar användarnas tänkbara datorsystem av er design?

[FORS98]

3.4 Riktlinjer för interaktiva webapplikationer

Det finns en del riktlinjer för interaktiva webapplikationer som bör följas:

- Acceptans av data – applikationen ska förse användaren med positiv feedback (frånvaron av respons är inte tillräcklig feedback) på att den data som skickats till systemet antingen har tagits emot eller inte kommit fram.
- Inloggning – Applikationer som kräver någon typ av inloggning ska ge tydlig indikation på varför ett inloggningsförsök misslyckats.

- **Hjälp** – On-line hjälp ska finnas och vara åtkomlig från applikationen. Dessutom ska hjälpen vara organiserad på ett sånt sätt att användaren kan få svar på frågor om applikationens användning. Användaren ska kunna välja mellan hjälp om applikationens olika delar och funktioner, och om det är tillrådligt, olika nivåer av hjälp.
- **Icke-obligatoriska fält** – Applikationer där användaren fyller i formulär med olika uppgifter ska tydligt visa vilka fält som är obligatoriska och vilka som kan utelämnas.
- **Felkorrigering** – Applikationen ska ha en enkel mekanism för att korrigera felaktigt ifyllda fält, och ska göra användaren uppmärksam på dessa fel. Dessutom ska användaren bara behöva korrigera felaktigt fält, och inte börja om från början med formuläret.
- **Meningsfulla felmeddelanden** – Meddelanden från applikationen ska vara meningsfulla och uttryckas i ett språk som användaren förstår. Dessutom ska de tala om för användaren hur denne kan fortsätta med applikationen.
- **Positiv indikering** – Applikationen ska på ett positivt sätt (frånvaron av respons är inte tillräckligt) indikera när en funktion har utförts.
- **Bekräftelse** – Användaren ska få bekräfta alla inmatningar som kan ha en potentiellt destruktiv effekt (till exempel radera eller skriva över data).
- **Minimera användarens inmatningar** – Användaren ska behöva fylla i så lite data som möjligt, vilket också innebär att denne inte ska behöva fylla i samma uppgift flera gånger. Så långt det är möjligt bör applikationen föreslå tänkbara värden i de olika fälten.
- **Enkla inmatningar** – Komplexa regler för inmatningar bör undvikas. Inmatningsfält ska vara så flexibla som det är möjligt, och ta emot data i olika format.
- **Obligatoriska fält** – Dessa ska markeras på ett tydligt sätt.
- **Möjlighet att avbryta** – Om det är praktiskt möjligt ska användaren, när han/hon vill, kunna avbryta applikationen.
- **Fältstorlek** – Inmatningsfält ska vara tillräckligt stora för att acceptera alla rimliga inmatningar.
- **Sekvensiering** – Fält bör komma i en logisk ordning, och följa uppgiftens utformning.

[FORS98]

3.5 Problem vid webdesign

En del problem från WWW kan återfinnas i WAP.

3.5.1 Inkompatibla webläsare

Ofta vill man skapa en applikation för den senaste webläsaren eftersom den har mest funktionalitet, men många användare har inte den senaste webläsaren och kanske inte ens den näst senaste. Det är därför viktigt att applikationen ser ut och fungerar på samma eller ett likartat sätt i olika läsare.

Utvecklaren bör känna till att och hur olika läsare tolkar informationen på olika sätt och vilka nya funktioner som går att utnyttja. Det finns olika sätt att lösa detta problem:

- Använd endast funktioner som finns i alla läsare. Om applikationen går att implementera på detta sätt är det här den enklaste lösningen, både tekniskt och tidsmässigt.
- Olika lösningar för olika läsare. Antingen känner applikationen själv igen vilken läsare som används och dirigerar olika läsare till olika applikationer eller så får användaren själv ange vilken läsare som används.

[ALBE95]

3.5.2 Begränsad bandbredd

Bandbredd mäts i Kbs (kilobits per second) eller Mbs (megabits per second) och är en viktig begränsning eftersom den avgör hur snabbt information överförs från servern till klienten. Ett 14.4 modem har en effektiv överföring på ungefär 1K per sekund, vilket innebär att en fil på 10K låter användaren vänta i ungefär 10 sekunder. Så länge filerna innehåller ren text brukar de inte vara så stora, men bilder tar alltid mer plats och skapar längre väntetider. Långa väntetider innebär både irritation och kostnader för användaren. Tänkbara lösningar:

- Använd bilder sparsamt. Det ska finnas en mening med att använda bilder, inte bara för att göra en häftigare site.
- Använd inte större bilder än du måste. Upplösning och storlek har stor inverkan på den tid det tar att ladda ner en bild.
- Feedback: Användarens tålamod ökar om denne får någon indikation på vad det är som händer och vad personen i fråga väntar på.

[ALBE95]

4 Säkerhet

Mobiltelefonen har tillsammans med den nya WAP-tekniken ökat marknaden för elektronisk handel. Detta har medfört att säkerheten blivit en allt viktigare faktor. För att kunna köpa och sälja utan risk över trådlösa nätverk måste man kunna utföra säkra och pålitliga transaktioner. För att åstadkomma detta kan man använda sig av kryptering och autentisering. Det finns olika krypteringsmetoder med olika funktionalitet. WAP-tekniken stödjer en del av dessa metoder. Vad gäller autentisering finns det olika sätt att identifiera användare. Nedan beskrivs vad kryptering och autentisering är och hur man använder dem.

[ENTR00] [WAPF99]

4.1 Kryptering

Kryptering är en process som går ut på att förvanska ett meddelande så att det inte blir läsbart för personer som inte har tillgång till en så kallad "nyckel". Bara om man har tillgång till nyckeln kan man återfå det ursprungliga meddelandet. Processen används huvudsakligen vid utbyte av viktig information över avlyssningsbara medier, men kan också användas för att skydda sig mot stöld av viktig information. Kryptering finns i två olika former, dels i symmetrisk form dvs den som krypterar och den som dekrypterar använder samma krypteringsnyckel och dels i asymmetrisk form där den som krypterar använder en nyckel (den privata eller hemliga) och den som dekrypterar använder en annan nyckel (den öppna eller publika). Dessa båda nycklar hör dock till ett nyckelpar som framställs samtidigt.

[DATR0]

Nedan följer de olika krypteringsmetoder som stöds av WAP. De finns i olika varianter, tex med olika längdrestriktioner på nyckellängden och med eller utan autentisering, men de beskrivs endast generellt nedan.

4.1.1 Diffie-Hellman

Diffie-Hellman nyckel överenskommelse protokoll (kallas också exponentiell nyckel överenskommelse) utvecklades av Diffie och Hellman år 1976. Protokollet tillåter två användare att utbyta en hemlig nyckel över ett osäkert medium utan att man i förväg bestämt någon hemlighet.

[LUTZ00]

4.1.2 RSA

RSA är ett öppen-nyckel kryptosystem för både kryptering och autentisering. Det uppfanns 1977 av Ron Rivest, Adi Shamir och Leonard Adleman och fungerar enligt följande: Kryptering och autentisering kan ske utan utbyte av hemliga nycklar. Varje inblandad part använder andras öppna nycklar och sin egen privata nyckel. Vem som helst kan skicka ett krypterat meddelande eller verifiera en digital signatur genom att enbart använda öppna nycklar, men bara med rätt privat nyckel kan man dekryptera ett meddelande eller signera ett meddelande.

[WWWR00]

4.2 Autenticering

Med autenticering menas fastställande av en identitet eller verifierande av identitet, exempelvis när man fastställer identiteten på en användare av ett system. Man använder autenticering för att se till att inte obehöriga får tillgång till system eller för att kontrollera att användaren är den den utger sig för att vara. Den vanligaste formen av autenticering är att man får ett användarnamn som visar att man är en viss person – det kallas identifiering. Sedan anger man ett lösenord vilket är beviset att man är den man är. Därför är det viktigt att lösenordet är personligt och inte sprids till obehöriga eftersom det medför att man inte kan vara säker på att personen som loggar in i system är den den utger sig för att vara.

Autenticering kan bygga på tre olika principer: någonting man kan (exempelvis ett lösenord), någonting man har (exempelvis ett passerkort) och någonting man är (exempelvis ett fingeravtryck). Vad gäller WAP-tekniken pratas det mest om digitala signaturer och sim-kort för identifiering. Nedan kommer därför en beskrivning av hur de fungerar.

4.2.1 Digital Signatur

En digital signatur är ett stycke data som genererats med en hemlig (privat) nyckel i något asymmetriskt chiffer. Och där möjlighet finns att via en publik (öppen) nyckel verifiera att informationen verkligen skapats via den korresponderande (avsändarens) privata nyckeln. Det är med andra ord ett autenticeringssystem (verifiering av avsändare).

Den digitala signaturen kan i den verkliga världen jämföras med namnunderskrift eller fingeravtryck.

[DATR0]

Nedan följer en digital signatur som används i samband med en krypteringsmetod för WAP.

ECDSA (Elliptic Curve Digital Algorithm)

ECDSA är en digital signatur som kan användas tillsammans med olika krypteringsmetoder. Den skiljer sig dock från den tidigare nämnda RSA. De två metoderna är båda standardiserade av ANSI, ISO och IEEE. De är alla kända standardiseringsorganisationer. ECDSA är den första standardiserade digitala signaturen som inkluderar specifikationer av metoder för att validera domänparametrar och publika nycklar – den kontrollerar att: det inte har blivit något räknefel under skapandet av nycklar och att ingen har försökt använda falska eller konstiga domänparametrar eller att falska publika nycklar skapats som förstör säkerheten. ECDSA tillåter att man använder större nycklar och större certifikat. Den har små datastrukturer och räknar effektivt vilket gör den lättare och billigare att implementera.

[CERT00]

4.2.2 SIM-kort

SIM står för Subscriber Identity Module som ungefär betyder Abonentens Identitets Modul. SIM-kortet är den delen av telefonen som är den egentliga telefonen. Det är i SIM-kortet som all personlig fakta om telefonen är sparad. Telefonen i sig själv har inget telefonnummer utan det är SIM-kortet som har det. Man kan med andra ord låna en annan telefon och sätta in ett eget SIM-kort i den, ringa som vanligt. Samtalet debiteras sedan på den egna räkningen precis som man skulle ha ringt från den egna telefonen. I SIM-kortens värld är man inte längre beroende av en och samma telefon, det enda man behöver är sitt eget SIM-kort. Eftersom all fakta om telefonen sparas i SIM-kortet kan man även använda det vid autenticering.

Den nya WAP-tekniken ger möjligheten att kunna använda den kod som finns inlagd på SIM-kortet för att identifiera användare. För att detta ska vara möjligt måste SIM-kortet på ett säkert sätt knytas till den person som ska autentiseras.

[OSBY00] [HILL99]

4.3 Dataintegritet

Ett exempel på transaktionshantering kan vara en kund som köper en flygbiljett och betalar för den med sitt kreditkort. När flygbolagets datorsystem tar hand om transaktionen är det viktigt för båda parter att utfärdandet av biljetten och debiteringen på kreditkortet sker som en enhetlig händelse. Flygbolaget vill inte utfärda biljetten innan den är betald och kunden vill inte betala innan biljetten är utfärdad. För att detta ska vara möjligt kontrollerar systemet att både utfärdandet av biljetten och debiteringen kan slutföras innan transaktionen påbörjas. Om transaktionen påbörjas men sedan av någon orsak avbryts återställs allting till det tillstånd som fanns innan transaktionen påbörjades. Med andra ord, antingen slutförs hela transaktionen eller ingenting alls. I ett system där flera användare utför mängder av transaktioner krävs det också att varje transaktion går rätt till oavsett vad som händer i övrigt i systemet, att en transaktion inte påverkar en annan, och att transaktionshanteringen sköter sin uppgift även vid fel i systemet. Man kan säga att en transaktion ska ha fyra egenskaper:

- Atomär – Transaktionen sker som en enhet
- Konsistent – Transaktionen har alltid samma effekt
- Isoerad – Transaktionen utförs som om inget annat händer samtidigt
- Uthållig – Resultatet av transaktionen sparas

[DAVI96]

5 Systemintegration

En av fördelarna som nämns med WAP är att tekniken ska kunna utnyttja befintlig infrastruktur och utvecklingsverktyg samt att det snabbt ska gå att anpassa befintliga webapplikationer till WAP. I det här kapitlet kommer vi att ta upp hur WAP kan integreras i ett befintligt system.

5.1 Klient/server

Det är viktigt att komma ihåg att WAP-applikationer bygger på klient/server-konceptet där den mesta funktionaliteten ligger på en server och inte på den handburna klienten. Riktlinjer för klient/server kan användas för WAP. Det finns ofta ett krav på kort utvecklingstid som gör att "Rapid Application Development" används. Tre koncept från RAD kan med fördel användas vid utveckling av WAP-applikationer:

- Försök att så snabbt som möjligt ta fram en användbar version av tjänsten för att sedan genom en iterativ process, antingen i fälttest eller i utvecklingsmiljö, involvera slutanvändarna och låta dem delta för att på så vis förbättra tjänsten.
- Orörliga deadlines, vilket innebär att en delvis fungerande version kan levereras även om en del faser tar längre tid än väntat.
- Förkorta den totala utvecklingstiden genom att återanvända så mycket som möjligt (eller som är praktiskt genomförbart).

En återkommande frågeställning i alla datorsystem och kanske speciellt i klient/server-miljö är dimensionering av hård- och mjukvara, vilket antal och vilken typ av servrar, nätverkskomponenter och klienter som krävs. Det gäller inte bara att tillfredsställa de behov som finns idag utan också försöka avgöra vilka behov som kommer att finnas i den närmaste framtiden. Man måste göra systemet skalbart för att tillåta en anpassning till ökade krav på systemet. Ett angreppssätt är att dimensionera systemet genom två iterationer, där man i den första iterationen uppskattar de krav som ställs idag och räknar med applikationer som är under utveckling. Man identifierar flaskhalsar och tar reda på vilka servrar som kommer att vara mest belastade. I den andra iterationen jobbar man för att strukturera bort flaskhalsar och överlastade servrar eller lägga till ytterligare kapacitet där det behövs. Det är viktigt att testa prestanda både på servrar och nätverk för att slutanvändarna ska slippa otrevliga överraskningar i form av långsamma eller oanvändbara system.

[FORG94]

5.2 Att flytta till klient/server

Här följer ett antal punkter som anses vara viktiga när man ska flytta en fristående applikation till en klient/servermiljö:

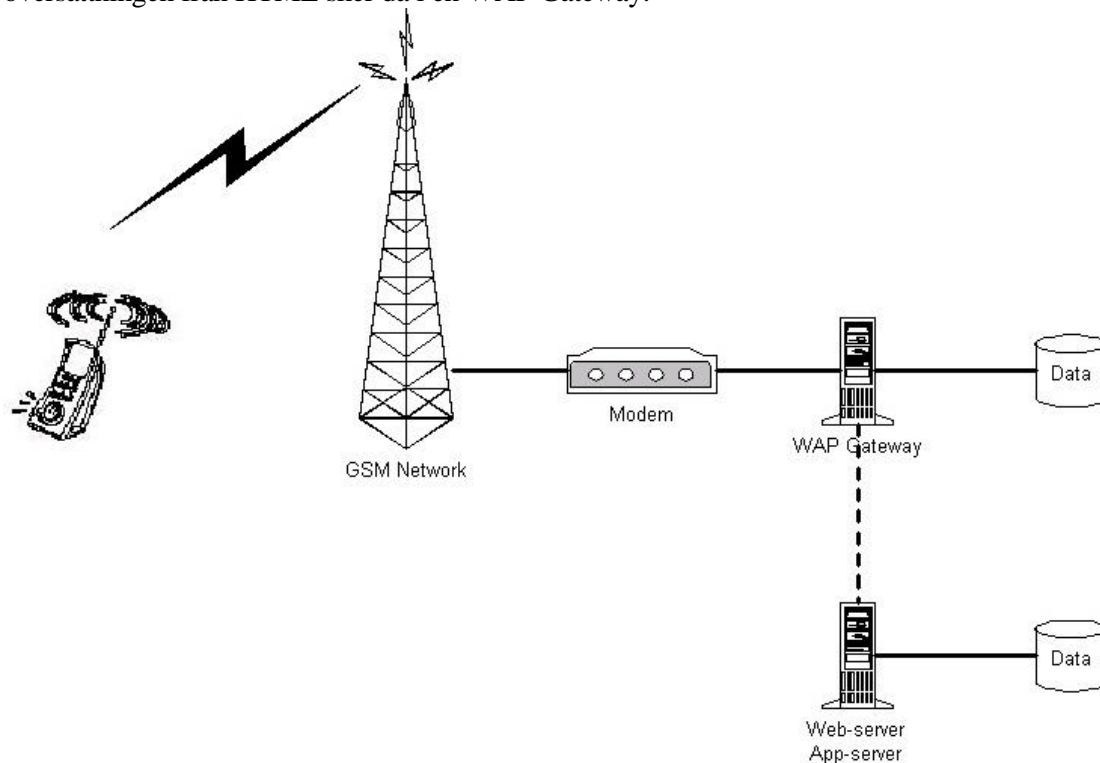
- Identifiera målen med migreringen i form av ny funktionalitet, prestanda eller reducerade kostnader.
- Samla in så mycket information som möjligt om applikationen, speciellt på hög nivå – det vill säga applikationslogik och de affärsregler som ligger bakom denna logik.
- Analysera applikationen och identifiera nyckelfunktioner och de processer som exekverar varje funktion.

- Jämför den ursprungliga applikationslogiken och verifiera att den överensstämmer med de bakomliggande affärsreglerna.
- Avgör vilka funktioner som ska utföras av server respektive klient.

[FORG94]

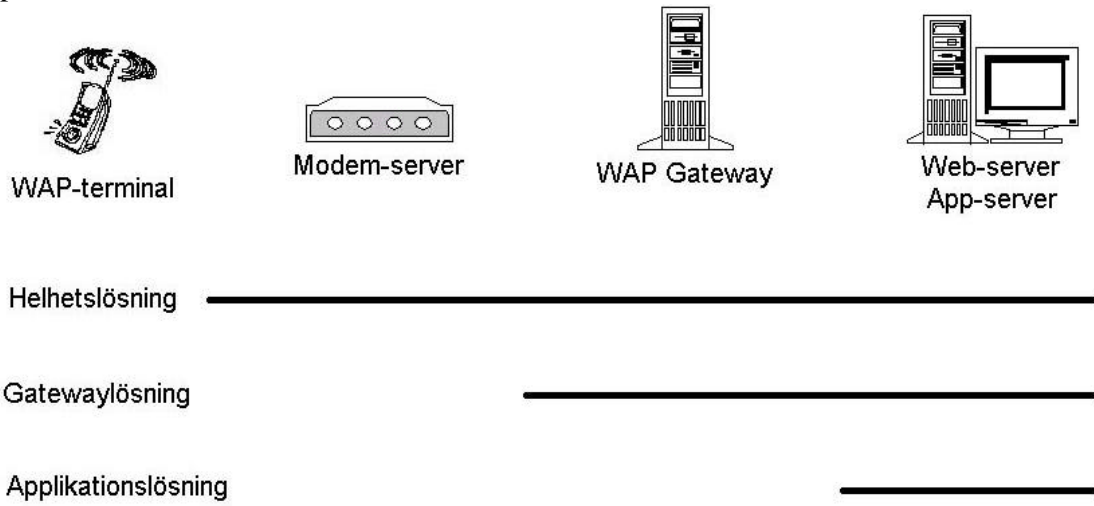
5.3 Systemlösningar för WAP

En typisk systemlösning för WAP-tjänster visas i Figur 4. Den mobila klienten kommunicerar med en WAP Gateway via GSM-nätet och ett modem. En WAP Gateway kan innehålla tjänster med koppling till en databas eller ett gränssnitt mot en speciell applikationsserver som i sin tur arbetar mot databasen. Applikationsservern kan ersättas med en vanlig webserver och översättningen från HTML sker då i en WAP Gateway.



Figur 4. Systemlösning för WAP-tjänster

Figur 5 visar de olika koncept som finns för företag. Det som skiljer de olika koncepten åt är vilka komponenter som finns i företagets ägo och vilka som erbjuds av en utomstående operatör.



Figur 5. Företagskoncept

6 Att utveckla för WAP

I det här kapitlet kommer vi att ta upp hur en kravspecifikation för en WAP-tjänst kan skrivas, ge en del tips vid WAP-utveckling och presentera en del allmän information om utvecklingsverktyg för WAP.

6.1 *Kravspecifikation för en WAP-tjänst*

Kravspecifikationen syftar till att ge en otvetydig beskrivning av vad systemet skall klara av. Den innehåller inte nödvändigtvis krav på hur systemet skall klara det. Ofta skrivs kravspecifikationen som vanlig löpande text. Man bör vara mycket noga med att formulera sig på ett icke tvetydigt sätt. Det skall inte gå att tolka texten på olika sätt. En kravspecifikation bör inte innehålla krav på hur systemet skall implementeras, eftersom det kan skapa problem senare i projektet när det kanske visar sig att just den algoritmen eller den tekniken inte passar till den infrastruktur som ska användas. Man bör inte heller låsa fast sig genom att redan i kravspecifikationen rikta sig mot en viss typ av klient eller server.

[HILL99]

6.1.1 Vad tjänsten skall utföra och med vilket resultat

Denna del beskriver först och främst vilka tjänster användaren skall kunna utnyttja med sin mobila enhet. Om det rör sig om kommersiella tjänster bör krav ingå som rör betalningen. Detaljer som rör hur eventuell betalning går till skall dock inte inkluderas i kravspecifikationen utan hör istället hemma i design och implementation. I samband med att krav ställs på vad WAP-tjänsterna skall utföra bör även krav ingå som specificerar hur lång tid tjänsterna maximalt tar att utföra. Kravspecifikationen innehåller ofta krav på användargränssnittet. Viss information som utgör kärnan i systemet kanske måste presenteras på ett visst sätt. Precis som HTML så är WML ett språk som specificerar vilken information som skall presenteras och endast i grova drag hur. Därför kan inte kravspecifikationen innehålla detaljer som rör layout. En mobiltelefon från en viss tillverkare kommer med största sannolikhet att använda en annan font än den som en PDA från en annan tillverkare använder. En mobil enhet av något slag kanske stödjer ett par olika bildformat utöver det obligatoriska WBMP typ 0 och andra kanske enbart stödjer detta format. Detta får till följd att en primitiv mobil enhet med mycket liten display endast presenterar en textuell version av gränssnittet samtidigt som andra enheter presenterar ett gränssnitt med färgbilder.

[HILL99]

6.1.2 Säkerhetskrav

Kravspecifikationen innehåller i vissa fall information som rör säkerheten i systemet. Man specificerar till exempel om systemet skall kräva användarnamn och lösenord för att användaren skall få tillgång till WAP-tjänsten och om säkerheten i systemet skall bygga på certifikat. Tunga tillämpningar inom bank- och finansväsendet kräver extremt hög säkerhet och då är kanske det ramverk för datasäkerhet som finns i WAP inte tillräckligt. Vissa institutioner och företag har valt att i sina internetapplikationer använda SSL i kombination med ytterligare hård- eller mjukvara.

Vilka åtkomsträttigheter som gäller bör också ingå i kravspecifikationen. Specifikationer som rör åtkomsträttigheter skall ange vem som skall ha tillgång till vad, vid vilken tidpunkt och hur länge. Kravspecifikationen bör alltså ange om åtkomsträttigheterna skall vara relaterade till serverdomän, tidpunkt och till exempel om lösenord krävs. Det bör klart framgå om

användaren skall kunna ha tillgång till sin tjänst från fler än en mobil enhet. Anledningen till detta är att avancerade WAP-tjänster kan använda sig av en kod som finns inlagd på en mobiltelefons sim-kort för att identifiera användaren. Användaren kanske önskar komma åt tjänsten från en kollegas telefon eller från en annan mobil enhet, till exempel en PDA. I detta fall måste man ta beslutet om tjänsten även då skall fungera och om den även i det fallet ska vara personlig.

[HILL99]

6.1.3 Prestandakrav

I kravspecifikationen kan det vara lämpligt att ange vid vilka tillfällen användaren tillåts utsättas för långa svarstider till följd av kommunikation över det mobila nätet och om systemet skall garantera svarstider under en viss nivå. Andra krav som rör prestanda kan till exempel vara att systemet garanterar, eller inte garanterar, att användarens mobila enhet fungerar även om nätförbindelsen är dålig.

Vad gäller krav på utnyttjande av systemets alla resurser på bästa sätt kan man försöka identifiera svaga punkter i systemet; hårddisk, primärminne, processorkraft, en databas på serversidan eller kanske minnet i de mobila enheterna som presenterar informationen. Det är inte alls ovanligt att man i specifikationen ställer krav på ett visst beteende då antalet användare ökar och systemet därför pressas hårdare. Kraven bör då vara ställda så att de svaga punkterna inte orsakar flaskhalsar. För en WAP-server är det till exempel viktigt att antalet trådar inte sätter begränsningar och att de exekverar tillräckligt snabbt. Detta är mer relaterat till design och implementation men redan i kravspecifikationen kan man specificera att en sådan plattform skall användas så att man är garanterad hög och bibehållen prestanda när antalet användare ökar.

[HILL99]

6.2 *Tips för WAP-utveckling*

Här presenteras en del tips som kan vara bra att känna till.

6.2.1 Variabelprefix

Eftersom varken WML eller WMLScript använder sig av typdeklarationer kan det vara lätt att tappa kontrollen över hur informationen representeras. Kneppiga buggar uppstår lätt i språk där man inte har någon omfattande typcheck. Ett tips är att använda sig av ett prefix eller postfix på variabelnamnen för att indikera dess typ. Alla variabler som innehåller värden av typen bool kan börja eller sluta med bokstaven b och alla heltal kan börja eller sluta med bokstaven i som i integer. På så sätt minskar man risken för fel som bygger på att en variabel av en viss typ av misstag tilldelas ett värde av en annan typ.

[HILL99]

6.2.2 Begränsat minnesutrymme

När man utvecklar för WAP bör man tänka på att många av de klienter som kommer att presentera informationen med all sannolikhet kommer att ha ett tämligen begränsat minnesutrymme. Meningen är att tillverkarna skall kunna använda sig av enkla mikroprocessorer. Strängar bör till exempel inte användas så att de tar alltför mycket minne. Om möjligheten finns bör istället en server användas för att hålla reda på större mängder infor-

mation. WML och WMLScript lämpar sig bättre för att hantera gränssnittet och samtidigt hålla reda på mindre datamängder relaterade till olika val och liknande.

WMLScript som språk har inte heller något stöd för felhantering vilket gör det ännu viktigare att tänka på minnesresurserna när man designar ett system. Visst stöd ges dock av de utvecklingsverktyg som används för att bygga systemen. Man beskriver tämligen omfattande vilka typer av fel som kan uppstå och vid vilka tillfällen. Återigen frigör detta processorkraft i de mobila enheterna vars interpretatorer för WML och WMLScript kan förenklas betydligt. En stor del av logiken för felhantering har flyttats till utvecklingsmiljöerna.

[HILL99]

6.2.3 Vanliga fel

Misstag i WML, förutom de rent syntaktiska, brukar handla om att man använder fel variabel, att en variabel borde ha nollställts (eller inte nollställts) vid bakåtnavigering eller att optionelementets attribute "value" innehåller ett felaktigt värde. Vid vissa tillfällen kan en uppdatering behövas. Det kan även finnas behov av att rensa hela WML-browser från variabelvärden för att få tillämpningen att fungera ordentligt. Variabelvärden som till exempel "name" eller "address" ligger vanligtvis kvar i minnet då man laddar ner en ny uppsättning kort (ett deck) över nätet. Skulle det vara så att den nya uppsättningen någonstans använder sig av samma variabelnamn kan man få problem. Därför har man möjlighet att använda attributet "newcontext" tillsammans med elementet "card". Om "newcontext" är satt till sant rensas minnet från alla variabelvärden då interpretatorn tolkar det kortet.

Tänk på att använda strängar och strängoperationer på rätt sätt. En väldigt lång sträng kan få en mobil enhet att krascha på grund av minnesbrist. I allmänhet bör man inte förlita sig på att text och bilder kommer att presenteras på ett visst sätt. Det är mindre lyckat att göra en WAP-tjänst där till exempel kursiv text har en speciell mening eftersom det inte finns några garantier för att den mobila enheten klarar av att visa kursiv text.

Fel kan uppstå under exekveringen om WML-interpretatorn instrueras att hoppa till en adress som inte finns. Här är man lite grand beroende av den utvecklingsmiljö som används. En riktigt bra utvecklingsmiljö bör erbjuda en kompileringsfunktion där verktyget inte bara tokeniserar koden utan även gör en omfattande semantisk check. URLs (<http://www.luth.se>) och URNs (#card1) specificeras naturligtvis med en viss syntax. URNs kan även undersökas i detalj för att säkerställa att de har definierats någonstans i koden.

[HILL99]

För tidig dekryptering – WTLS definierar kryptering mellan den mobila klienten och WAP Gateway. För att skapa en säker kommunikation med en applikationsserver, till exempel en bankserver, måste WAP Gateway etablera en säker (https) uppkoppling till denna server. I det fallet har proxyservern faktiskt tillgång till avkrypterad data från både klient och server.

[MOBI99]

6.3 Utvecklingsverktyg för WAP

WML och WAP är olika implementerat i telefonernas browser. Visserligen finns det en standard, men dels har inte alla browser hela standarden implementerad, dels finns i standarden saker som kan tolkas på olika sätt. De är skillnader av samma karaktär som de som

finns mellan olika databasleverantörers sätt att implementera SQL. Dessutom kan ambitiösa leverantörer lägga till egna finesser, precis som är fallet med HTML-bläddrarna. Ett enkelt exempel på hur telefonbläddrarna kan skilja sig åt är adresseringen i länkar; det finns WML-browser som kräver absolut adress men det finns även de som klarar relativ.

Programmerarna har än så länge inte tillgång till något program som kan kontrollera WML-syntaxen. Det finns ett par enkla program med vars hjälp man kan kontrollera statisk WML-kod, men det finns inget som kontrollerar dynamisk. Man får testa sina program genom att köra dem i browsern och se om de fungerar. Mer primitivt kan det knappast bli. Tänk dig en värld där det saknas syntax-kontrollmöjligheter för HTML eller Java!

Alla större leverantörer och många små har syntaxkontroll på gång, men dessa verktyg finns ännu inte på marknaden. Antalet utplaceringar och funktionstangenter skiljer sig åt mellan olika telefonmodeller. Det säger sig självt att den telefon som har fler funktionstangenter inte kräver så många "softkeys", knappar som syns på skärmen, som en som har färre. Och även om två modeller har lika många funktionstangenter kan användningsmöjligheterna skilja sig åt. Kanske placeringen av funktionstangenterna skiljer sig så mycket åt att det är svårt att använda funktionstangenterna på samma sätt och fortfarande ha ett logiskt gränssnitt. Telefoner har olika stora fönster. I höjddled spelar det mindre roll, för det går att rulla upp och ned på en sida. Men se, det går inte att rulla i sidled, och WML/WAP är idag inte bättre än att en rad som är för bred för ett visst fönster bryts av och fortsätter på nästa rad. Försöker man presentera en sida med 12 kolumners bredd i ett fönster som klarar 10 kommer varje rad att bli två, där man har 10 tecken på första raden och två på den andra. Det blir knappast läsbart. Det säkra sättet att lösa detta är förstås att man anpassar sina WAP-sidor till marknadens smalaste fönster. Men då drar man inte nytta av den större bredd som finns i mer avancerade klienter.

De två sista faktorerna, telefonernas skilda utseende och beteende, lär knappast växa bort med tiden. Alla erfarenheter från alla IT-marknader visar tvärtom att just utseende och beteende är en slags teknik- och marknadsdriven tävling där man aldrig når fram till målet. Därför måste man som WAP/WML-utvecklare kunna hantera att telefonerna inte betar sig likadant. För att klara det måste man veta vilken sorts telefon det är som man ska skicka information till. I WAP-standarden finns stöd för detta. Det kan ta ett tag innan alla tillverkare har stöd för det men den som inte har det inom en rimlig framtid lär få svårt att ta marknad. Därför måste den som vill utveckla WAP/WML-system redan nu tänka igenom hur deras program ska struktureras för att kunna hantera flera telefonmodeller. Likheten med att anpassa HTML-program för olika typer av browser och datorer är uppenbar. Men det faktum att fönstren kan variera så mycket i storlek och att det inte går att rulla i sidled kräver lite mer eftertanke; exempelvis vill man kanske i ett bredare fönster visa viss tilläggsinformation i form av en tabell eller liknande. Sedan bör man också tänka på att WML-standarden kommer att utvecklas snabbt de närmaste åren.

Allt det ovanstående tyder på att man ska undvika hårdkodade program. Istället bör man satsa på någon form av neutralt presentationsformat från vilket man enkelt kan översätta till olika sorters specifik kod. Sedan bör man kombinera detta med någon form av parameterstyrning per telefonmodell. När man talar om neutralt presentationsformat ligger XML bra till. För parameterstyrning passar det bra med en databas med information om olika modeller och hur data ska presenteras på respektive modell. Oracle har nyligen lanserat en produkt, Panama, som bygger på detta koncept. I bilaga B presenteras olika utvecklingsverktyg mer utförligt.

[DATA99]

7 En metod för att anpassa applikationer till WAP

I det här kapitlet kommer vi att knyta ihop de olika teoriavsnitten till en metod för att anpassa en befintlig applikation till WAP-teknik.

7.1 Vad är möjligt med WAP?

När man studerar en ny teknik är det en bra idé att fråga sig vad som överhuvudtaget är möjligt att göra med tekniken, och sedan vad som är lämpligt att göra. Det är nämligen inte säkert att svaret på de båda frågorna blir detsamma. I avsnitt 2.2 och 2.3 tar vi upp de begränsningar och möjligheter som WAP-tekniken har. En begränsning är det trådlösa överföringsmediet som ger en lägre bandbredd och mindre stabilitet i uppkopplingen i jämförelse med fasta nätverk. En annan begränsning är de resurser som den mobila klienten har, och som framstår som torftiga i jämförelse med de datorer som finns på de flesta skrivbord idag. WAP-tekniken har tagits fram just för att möta dessa begränsningar. En stor del av funktionaliteten kan hanteras av en server med de resurser som klienten saknar. Detta gör att den tydligaste begränsningen ligger i användargränssnittet, som beskrivs i avsnitt 2.2.4, och i presentationspråket WML, som beskrivs i avsnitt 2.3.5. Möjligheten att lägga funktionalitet på en eller flera servrar gör att avancerad databehandling och lagring av stora mängder information som samlats in från klienterna är fullt möjlig. Det är alltså i första hand på klientsidan som vi hittar de begränsningar som finns. Det går endast att presentera en begränsad mängd information på varje skärmbild, eller kort som det heter i WML. Givetvis finns det möjlighet att sprida informationen över flera kort eller låta användaren bläddra upp och ner i informationen men det innebär en del påfrestningar på användarens tålamod. Möjligheten att visa bilder på en mobil klient bestäms främst av displayens storlek och upplösning, och när det gäller mobiltelefoner är dessa möjligheter i stort sett obefintliga.

Även för inmatningar från användaren finns det begränsningar i form av att knappsatsen på mobiltelefonerna inte lämpar sig för att mata in stora mängder information. För handdatorer finns det möjlighet att använda en form av skriftspråk på den tryckkänsliga displayen, och för mobiltelefoner finns det ett litet tangentbord som kan kopplas på telefonen, men båda dessa metoder ger inte alls samma möjligheter som ett alfanumeriskt tangentbord och en mus. Kontentan av detta är att WML och WAP-tekniken lämpar sig väl för tjänster som visar en begränsad mängd information för användaren, och som kräver begränsade inmatningar, gärna i form av menyer eller flervalsoalternativ. Detta ger också en fingervisning om vilka tjänster, eller delar av tjänster, som är lämpliga att anpassa till WAP-tekniken.

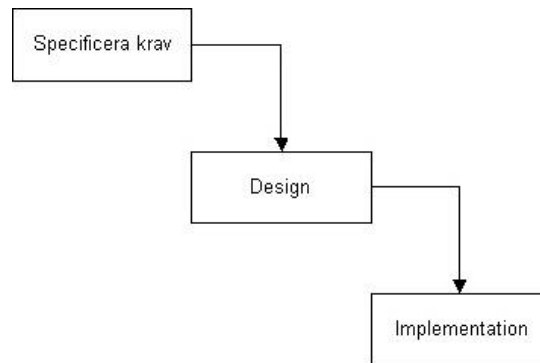
7.2 Översikt av anpassningsmetoden

När vi sammanställde vår metod har vi utgått ifrån Pressmans generella syn på mjukvaruutveckling. Enligt detta synsätt består all mjukvaruutveckling av tre faser: Definition, Utveckling och Underhåll. I definitionsfasen söker man svar på frågan om vad som ska göras. Man försöker identifiera vilken information som ska behandlas, vilka funktioner som behövs och vilka krav som ställs på det system som utvecklas.

I utvecklingsfasen fokuserar man på hur de olika kraven ska realiseras, hur gränssnittet mot användaren ska se ut och hur detta ska implementeras och testas. I underhållsfasen upprepar man de båda föregående stegen, men tar hänsyn till den befintliga mjukvaran.

[PRES94]

Vi har valt att dela in vår metod i tre steg och avbryta arbetet efter implementationen. Det första steget innebär det Pressman kallar Definition och de två följande stegen faller under Utveckling. Denna process ska ses som ett iterativt förlopp i och med att man kan gå tillbaka och göra om ett steg i den mån det behövs. Figur 6 visar en översikt av metoden.



Figur 6. Metodöversikt

I avsnitt 7.3 , 7.4 och 7.5 går vi igenom de olika stegen mer utförligt.

7.3 Att skapa en kravspecifikation

För att skapa en kravspecifikation har vi två resurser till vår hjälp; dels har vi den befintliga applikation som vi ska skapa en WAP-tjänst av och dels har vi den kund som vi ska skapa WAP-tjänsten åt. När man ska ta fram en kravspecifikation är det viktigt att involvera de tänkta användarna av systemet. Detta gör man genom att ge olika representanter för kunden möjlighet att ställa krav på det tänkta systemet. Vi ser också att de fyra första punkterna i avsnitt 5.2 underlättas markant av kundens medverkan. Vi kan använda de fyra punkterna som en grund för skapandet av kravspecifikationen samt ta hänsyn till det som sägs i avsnitt 6.1 om kravspecifikation för en WAP-tjänst. Det är också viktigt att känna till det som sägs i kapitel 4 angående WAP och säkerhet. Givetvis måste vi också diskutera med kunden vilka begränsningar som finns med WAP-tekniken och då framföra det vi kommit fram till i avsnitt 7.1 angående vad som är möjligt och vad som är lämpligt att implementera som en WAP-tjänst.

7.4 Riktlinjer för design

Här kommer den femte punkten i avsnitt 5.2 in: Vilka funktioner ska utföras av server respektive klient. Här kommer också hela kapitel 3 in: Riktlinjer för människa/dator-interaktion. Vi kan dela in designarbetet i två delar – funktionell design, som främst handlar om vilka funktioner som ska hanteras av server respektive klient och hur funktionerna ska utformas, samt gränssnittsdesign, som handlar om hur gränssnittet som möter användaren ska se ut. Nedan har vi sammanställt riktlinjerna från kapitel 3 och grupperat dessa under ett antal olika rubriker. Detta har vi gjort för att på ett tydligt sätt i avsnitt 8.4 presentera hur vi har realiserat de olika riktlinjerna med hjälp av WAP-tekniken.

Instruktioner till användaren

- Meningsfulla titlar – undvik en teknisk jargong.
- Meningsfulla beskrivningar av fält – använd ett allmänt språk.
- Tydliga instruktioner – använd ett lätt språk

- Använd användarens språk
- Konsistent terminologi och konsistenta förkortningar.
- Ge användaren en indikation på vilka inmatningar som förväntas.
- Icke obligatoriska fält ska vara tydligt markerade – Obligatoriska fält ska visas före de icke obligatoriska.

Feedback

- Tydlig markering när ett formulär är ifyllt.
- Feedback – Användaren ska få respons på utförda handlingar.
- Acceptans av data
- Positiv indikering
- Bekräftelse

Arbetsgång

- Användaren ska kontrollera applikationen, inte tvärtom.
- Val istället för inmatning
- Enkelhet – Det ska vara enkelt för användaren att använda applikationen.
- Minimera användarens inmatningar
- Enkla inmatningar
- Möjlighet att avbryta
- Sekvensiering

Navigering

- Logisk gruppering och sekvensiering av fält – fält som hör ihop bör grupperas tillsammans, fälten bör ha en logisk ordning.
- Enkel navigering mellan fält – använd tab eller piltangenter.

Informationsmängd/Presentation

- Visuellt tilltalande layout – var uppmärksam på fältens placering på skärmen.
- Förklaring för fält ska placeras på ett konsistent sätt.
- Konsistens – Ett konsistent gränssnitt gör att användaren känner igen liknande situationer och kan dra lärdom av detta i en ny applikation.

- Motivation – Gränssnittet ska motivera användarna.
- Avlasta minnet – Tänk på det mänskliga minnets begränsningar.
- Reducera den kognitiva belastningen på användarna.
- Var konsistent och tydlig.

Felhantering

- Felkorrigerigering för enstaka fält – enkelt att radera och korrigera inmatningar.
- Felmeddelande för otillåtna inmatningar – felmeddelande ska visas när det felaktiga värdet fyllts i och tillåtna värden ska visas i felmeddelandet.
- Förlåtande – Tillåt användaren att göra fel eller ändra sig.
- Ta hänsyn till fel.
- Hjälp – On-line hjälp ska finnas och vara åtkomlig från applikationen.
- Meningsfulla felmeddelanden

7.5 Implementation

Vid implementationen utför vi själva kodningen av applikationen och omsätter krav och design till en fungerande applikation. Här tar vi hänsyn till det som sägs i avsnitt 5.1 och i kapitel 6. I kapitel 8 visas hur olika krav kan realiseras med WAP-teknik.

8 Prototyp av en WAP-tjänst

I det här kapitlet kommer vi att beskriva vilka krav Frontec Norr AB ställde på prototypen. Vi kommer vidare att beskriva hur vi har realiserat dessa krav och visa några exempel från prototypen. Slutligen kommer vi att presentera resultatet av en enkätundersökning som gjordes vid Frontec Norr AB för att utreda hur väl vi lyckats med prototypen.

8.1 Val av utvecklingsverktyg

Det finns, som vi redan nämnt, olika typer av utvecklingsmiljöer på marknaden. Innan vi kunde börja arbetet med prototypen gick vi igenom de olika alternativen och valde därefter att jobba med Nokias utvecklingsmiljö. De fem utvecklingsmiljöer som vi tittade närmare på presenteras i bilaga B. WAP Developer Toolkit och UP Software Development Kit innehåller en WML-editor och en WAP-emulator, och saknade den möjlighet till databaskoppling som vi behövde för att lagra informationen. Vi försökte ta kontakt med Oracle för att diskutera möjligheterna att använda deras utvecklingsmiljö Portal-To-Go men eftersom de inte hörde av sig så föll det alternativet också bort. Kvar fanns då Ericssons och Nokias miljöer, och båda dessa erbjuder en komplett utvecklingsmiljö med möjligheter till databaskoppling. Man kan ladda ner Ericssons WapIDE och använda den utan kostnad, men WAP Gateway/Proxy finns att köpa komplett med hårdvara vilket föll utanför de ekonomiska ramarna för vår uppsats. Nokia erbjuder en 30-dagars provversion av hela sin miljö så vi valde att arbeta med denna.

8.2 Tur och Retur

Den applikation som Frontec Norr AB ville att vi skulle anpassa till WAP-teknik heter Tur och Retur. Tur och Retur är ett reseräkningsprogram, det vill säga ett program för att skriva in olika utgifter som uppkommer i samband med resor inom arbetet. Det kan vara såväl inrikes som utrikes resor och det kan gälla utgifter som exempelvis bilersättning, logi kostnader eller matkostnader. Programmet hanterar många olika typer av kostnader och olika beräkningar vilket gör det ganska omfattande. Därför valdes enbart ett antal specifika funktioner ut som skulle vara lämpliga att realisera i en WAP-applikation. De tjänster som man ska kunna utföra är att: registrera bilersättning och registrera olika utlägg. Bilersättning innebär att man skriver in mellan vilka sträckor man åkt och hur långt avståndet mellan dem var. Med utlägg kan menas olika saker, exempelvis mat, transport (tåg, taxi med mera) eller övriga kostnader.

8.3 Krav

För att genomföra detta hade Frontec Norr AB en del krav på applikationen. Den skulle endast utgöra en begränsad del av originalet och man skulle kunna läsa, uppdatera och spara data samtidigt som man är ute från kontoret. De delar som ska realiseras med WAP-teknik är dels att boka in en ny resa eller välja en befintlig resa. Man ska också kunna lägga in vilka utlägg man gör under resan för att på så sätt skapa en grund för utbetalning av traktamente. Även bilersättning ska kunna läggas in. För ny resa ska man lägga in datum, tid och resmål. Som en förebild för vår prototyp fick vi en web-version av reseräkningsmodulen till Tur och retur. En förutsättning är att man kan planera resan hemma och via webapplikationen fylla i en del uppgifter. Dessa grunduppgifter lagras i en databas för senare användning. En annan begränsning i prototypen är att bilersättning endast behöver hantera bilen som fordon. Identifiering av användaren ska ske via telefonen med någon form av inloggning. Befintliga användarnamn och lösenord finns i databasen. Om användaren inte har bokat in någon resa ska möjlighet finnas att göra detta via telefonen. Sedan ska valet stå mellan utlägg och bilersättning. Utlägg ska ta in posterna kostnad inklusive moms, utläggstyp och eventuellt en notering. Bilersättning ska ta in färdväg och antal kilometer samt eventuellt släpvagn. Eventuellt ska åtskillnad göras mellan inrikes och utrikes resor.

Dessa krav visar vad tjänsten ska utföra och med vilket resultat, enligt det som presenteras i avsnitt 6.1.1. Eftersom det rör sig om en prototyp har vi inte haft några krav på säkerhet och prestanda, enligt avsnitt 6.1.2 och 6.1.3.

8.4 Tillämpning av metoden

Med ledning av Frontecs krav utformade vi den prototyp som beskrivs i detta avsnitt. Figur 15 visar den logiska design som vi har arbetat efter. En enkel cirkel representerar ett card med WML-kod och en dubbel cirkel representerar en java-servlet som hanterar någon typ av databaskoppling. När användaren väljer tjänsten Tur och retur i sin mobila klient möts han/hon av ett inloggningsförfarande. Användaren anger namn och lösenord, sedan jämförs de inmatade uppgifterna med en tabell över användare som finns lagrade i en databas. Figur 7 visar den skärmbild som möter användaren vid inloggningsförfarandets början och Figur 8 visar skärmbilden när användaren matat in uppgifterna.



Figur 7. Inloggning startar



Figur 8. Inmatade uppgifter

Om de inmatade uppgifterna stämmer med någon användare i databasen så är inloggningen genomförd och användaren får möjlighet att välja om han/hon vill skapa en ny resa eller lägga till uppgifter till en befintlig resebokning. Figur 9 visar en skärmbild där användaren gör detta val. Vid val av Skapa ny resa får användaren sedan mata in de uppgifter som behövs för att registrera en resa i databasen. Figur 10 visar skärmbilden där användaren matar in Ortsnamn för avreseort och slutdestination.



Figur 9. Ny eller befintlig resa



Figur 10. Inmatning av Ortsnamn

När användaren har skapat en ny resa eller hämtat en befintlig kan han/hon välja att registrera en begäran om bilersättning eller ersättning för utlägg under resans gång. Här visar vi endast tjänsten utlägg. Användaren väljer mellan olika typer av utlägg som visas i Figur 11 och får sedan skriva in ett belopp, som visas i Figur 12.

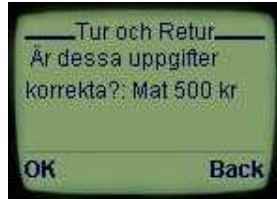


Figur 11. Utläggstyper



Figur 12. Inmatning av kostnad

När användaren har matat in dessa uppgifter får han/hon en sammanställning av de inmatade uppgifterna innan de lagras i databasen. Detta ger användaren möjlighet att gå tillbaka och ändra felaktigt inmatade uppgifter. Detta visas i Figur 13. Om användaren godkänner uppgifterna lagras de i databasen och användaren får sedan välja mellan att registrera ett nytt utlägg eller logga ut. Detta visas i Figur 14.



Figur 13. Feedback

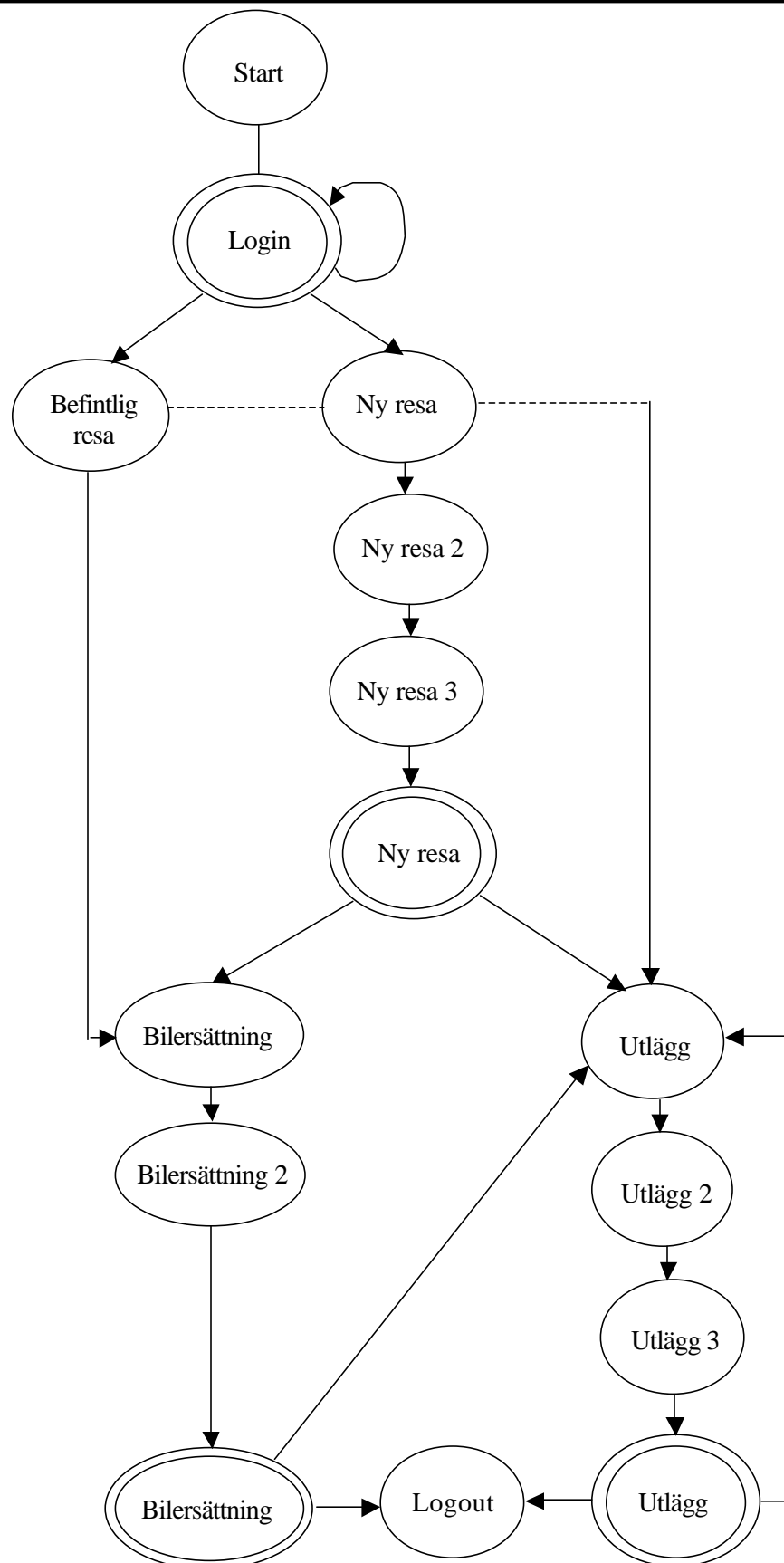


Figur 14. Val av tjänst

Under utvecklingen av prototypen har vi försökt ta hänsyn till de olika riktlinjer som vi specificerat i avsnitt 7.4. Vi ger användaren tydliga instruktioner genom att använda logiska namn på de olika fälten. Till stor del har vi valt att använda samma namn som i den ursprungliga applikationen för att användaren ska känna igen sig.

Vi ger användaren feedback på utförda inmatningar genom att visa de inmatade värdena på skärmen. Vi visar också en sammanställning av de inmatade uppgifterna innan de lagras i databasen. För att arbetsgången i tjänsten ska vara så enkel som möjligt har vi valt att inte visa en speciell skärmbild som talar om att uppgifterna lagrats. Istället förstår användaren att så skett när nästa skärmbild visas. Arbetsgången är sekventiell och följer till största delen den ursprungliga applikationen. Navigering sker med hjälp av telefonens funktionstangenter, och två val på varje skärmbild – OK eller Back. OK godkänner de inmatade uppgifterna och Back går tillbaka till föregående skärmbild. Utrymmet för att presentera information är begränsat men det finns möjlighet att bläddra upp och ner i fönstret. Vi har dock valt att inte presentera så mycket information på varje skärmbild för att tjänsten ska vara lättöverskådlig.

Vad gäller felhantering har vi försökt undvika felaktiga inmatningar genom att använda flervalstlistor istället för inmatningsfält. Där detta ej är möjligt eftersom de förväntade inmatningarna inte är kända i förväg har vi ingen felkontroll. Vi visar istället de inmatade uppgifterna innan de lagras i databasen för att ge användaren möjlighet till korrigering. Vi har valt detta tillvägagångssätt för att undvika onödiga skärmbilder i prototypen.



Figur 15. Logisk design av prototyp

8.5 Enkät

För att utnyttja möjligheten att differentiera längs någon tänkt underliggande dimension används ofta en graderad skala, vanligtvis i fem steg (Instämmer starkt, Instämmer delvis, Varken för eller emot, Tar delvis avstånd, Tar starkt avstånd). Statistiska centralbyråns erfarenhet är dock att en 11-gradig skala ger ett bättre resultat. Skalan uttrycks i procent och sträcker sig från 0 till 100 där ändpunkterna verbaliseras med orden Helt säkert och Absolut ingen möjlighet.

SCB förordar användandet av den så kallade Tratt-tekniken där undersökningen börjar med ett antal generella frågor för att sedan övergå i mer detaljerad form. Ett skäl till detta sätt att fråga är att folk lättare och bättre antas kunna svara på specifika frågor om de fått chansen att begrunda dem i ett större sammanhang. Ett annat skäl är att placeringen av de specifika frågorna först skulle kunna inverka på svaren på den mer allmänna.

Enligt SCB ska en blankett vara:

- Intresseväckande
- Lätt att förstå och fylla i
- Rationell att hantera

Blanketten bör därför vara systematisk och överskådlig, funktionell och tilltalande. Rätt utformad minskar den databearbetningskostnaderna och ger en bättre kvalitet på insamlade data.

Vad vill vi ha ut av enkäten?

Två saker vill vi ha ut av enkäten, dels om vi har lyckats tillämpa de riktlinjer vi satt upp för gränssnitt i en WAP-tjänst och dels om vi lyckats överföra de valda delarna av den ursprungliga applikationen på ett bra sätt. Enligt SCB's rekommendationer börjar vi enkäten med ett antal generella frågor för att sedan gå ner på mer specifika frågor för vårt syfte med enkäten.

[WÄRN90]

Vi har sammanställt enkäten och räknat ut ett medelvärde för de fyra första frågorna där vi använt oss av en 11-gradig skala. För de övriga sex frågorna har vi låtit varje person motivera sitt svar för att få en djupare förståelse av personens intryck. Slutligen har vi gett personen möjlighet att ge övriga kommentarer.

Fråga 1 "I vilken utsträckning är du bekant med tur och retur?" gav ett medelvärde på 78 vilket visar att enkätdeltagarna har god kännedom om den ursprungliga applikationen.

Fråga 2 "I vilken utsträckning är du bekant med webversionen av tur och retur?" gav ett medelvärde på 36 vilket visar att enkätdeltagarna har betydligt mindre erfarenhet av webversionen. Webversionen av tur och retur har inte släppts ännu, men trots detta har tre enkätdeltagare 100 i erfarenhet av denna version vilket kan bero på att de har arbetat med utvecklingen av denna.

Fråga 3 "Anser du att prototypen ger ett bra helhetsintryck?" gav ett medelvärde på 70 vilket visar att flertalet enkätdeltagare anser att vi lyckats med vårt arbete att skapa en fungerande prototyp.

Fråga 4 "Anser du att tjänsten utlägg fungerar på ett likvärdigt sätt i prototypen som i den ursprungliga applikationen?" gav ett medelvärde på 63 vilket visar att vi åtminstone till viss del lyckats överföra denna del av tur och retur till WAP-teknik.

Fråga 5 "Tycker du att det framgår vilken inmatning som förväntas i de olika fälten?" fick 8 positiva svar, 2 neutrala och 1 negativt vilket visar att vi uppfyllt denna riktlinje på ett för användarna tillfredsställande sätt. Kommentarer till denna fråga var bland annat "Eventuellt bör man förtydliga vilken typ av data som förväntas", "enkla, talande ledtexter" och "Ja, om man kan tur och retur".

Fråga 6 "Ger prototypen tillräcklig feedback på utförda operationer?" fick 9 positiva och 2 neutrala svar vilket visar att vi lyckats uppfylla denna riktlinje på ett för användarna tillfredsställande sätt. Kommentarer till denna fråga var bland annat "Ja, allt avslutas med en total sammanställning" och "Ja, kontrollfrågor finns om inmatad info är korrekt".

Fråga 7 "Har prototypen en logisk arbetsgång?" fick 9 positiva och 2 neutrala svar vilket visar att vi lyckats uppfylla denna riktlinje på ett för användarna tillfredsställande sätt. Kommentarer till denna fråga var bland annat "Någotsånär, liten skärmyta", "Ja, frågor kommer i rätt ordning" och "Ja, enkel och strukturerad arbetsgång".

Fråga 8 "Är det lätt att veta var i applikationen man befinner sig?" fick 5 positiva, 4 neutrala och 2 negativa svar vilket visar att vi inte lyckats riktigt lika bra med denna riktlinje men att det positiva intrycket ändå överväger. Kommentarer till denna fråga var bland annat "Ja, om man är bekant. En sitemap kanske kunde passa", "Verkar lite stökigt och mycket knapptryck" och "Nja, displayen begränsar överblick".

Fråga 9 "Är det för mycket eller för litet information på varje skärmbild?" fick 7 positiva, 1 neutralt och 3 negativa svar vilket visar att vi lyckats anpassa informationsmängden på ett övervägande positivt sätt. Kommentarer till denna fråga var bland annat "Det kunde vara mer", "Nja, lite rörigt" och "Precis det som krävs, varken mer eller mindre".

Fråga 10 "Är det lätt att ångra en felaktig inmatning?" fick 3 positiva, 4 neutrala och fyra negativa svar vilket visar att vi inte lyckats speciellt bra med att göra tjänsten förlåtande. Kommentarer till denna fråga var bland annat "Ja, det är lätt", "Nej, inte i prototypen. Men det är fullt möjligt att hantera" och "Nja, bättre att välja ja eller nej för godkännande eller underkännande".

Övriga kommentarer var "Bra utifrån de möjligheter som WAP ger", "Intressant teknik. Början på något nytt", "Sammanfattningsvis ett intressant arbete som går att vidareutveckla", "Intressant och spännande" och "Demon visar på begränsningarna i WAP-standarderna och den begränsade bildskärmsstorleken.

9 Slutsatser

Utifrån de erfarenheter vi har fått under arbetet med prototypen och rapporten anser vi att de riktlinjer som gäller för generell gränssnittsdesign såväl som de riktlinjer som finns för webutveckling också gäller för WAP-tekniken. När man implementerar dessa riktlinjer får man ta hänsyn till att en mobil klient skiljer sig markant från en traditionell dator och därför anpassa utformningen av gränssnittet till detta. Man måste också tänka på att själva användningen av en dator skiljer sig från en mobiltelefon eller en handdator. När en användare sitter vid en dator och utför en arbetsuppgift har denne sin fulla uppmärksamhet riktad mot skärmen eftersom arbetsuppgiften i sig utförs på skärmen. När det gäller en mobil klient är denna mer ett stöd för att utföra en arbetsuppgift som inte direkt finns i klienten. En annan sak som man måste ta hänsyn till är att in- och utmatningsenheter ser ganska olika ut. Den mobila klientens lilla skärmyta och begränsade knappats ger inte samma möjligheter som en monitor och ett tangentbord med tillhörande mus. WAP-tekniken har inte heller samma möjligheter att hantera färg och bilder.

Under utvecklingen av prototypen har vi försökt ta hänsyn till allt detta, och enkäten vars resultat presenteras i avsnitt 8.5 visar att vi har lyckats relativt bra med detta. Av de riktlinjer vi har försökt följa i vår prototyp var de endast möjligheten att ångra inmatade uppgifter som de olika enkärdeltagarna upplevde som mindre bra. I vår prototyp består möjligheten att ångra inmatningar av en funktion som gör att man kan backa till föregående skärmbild och ändra det inmatade värdet. Detta skiljer sig från det traditionella arbetssättet genom att man ibland får backa flera skärmbilder för att ändra en uppgift och detta tyckte inte användarna om. Vi har valt denna lösning för att minimera antalet skärmbilder som användaren får gå igenom men det kanske inte var rätt sätt att lösa det på.

Eftersom vidare förbättringar av tjänsten inte faller inom ramen för vårt arbete kommer vi inte att åtgärda detta, men vid utveckling av WAP-tjänster kan det vara en bra idé att stämma av resultatet mot de tilltänkta användarna och sedan genomföra den här typen av ändringar.

9.1 Egna reflektioner

Under arbetet med uppsatsen har vi reflekterat över vad tekniken kan åstadkomma och vad den kan få för framtid. I dagsläget finns det inte så många klienter på marknaden men detta kan komma att ändras inom en snar framtid. Som vi ser det finns det tre typer av köpare:

- Entusiaster – kan tänka sig köpa en telefon för att komma åt vissa tjänster
- De som köpt en telefon och kan tänka sig använda vissa tjänster (den breda massan)
- De som via jobbet fått en telefon för ett visst ändamål men kan tänka sig att använda fler tjänster

För att tekniken ska få en bred genomslagskraft räcker det inte med att sälja telefoner, det måste finnas intressanta tjänster och tillämpningar också. Man pratar oftast om tre typer av tjänster:

- Fristående WAP-tjänster, exempelvis nyheter och börsinformation
- Integrerade WAP-tjänster, exempelvis Tur och Retur
- Tjänster med koppling till telefonen, exempelvis en LDAP-lösning för telefonbok

Det som känns extra intressant med den här tekniken är möjligheten att integrera en WAP-lösning med andra system, ungefär som vi har gjort i vår prototyp. Vissa funktioner kan utnyttjas från den mobila klienten, vissa från en webbläsare och andra från en traditionell klient/server-applikation. Det är en spännande framtid som går oss till mötes.

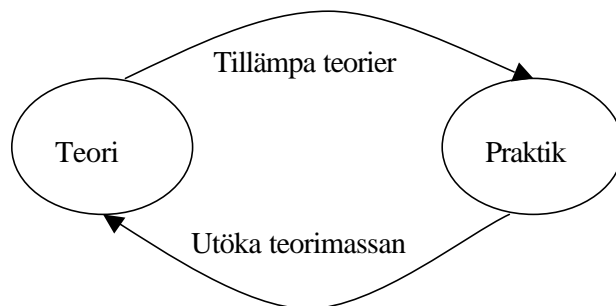
10 Efterord

I det här kapitlet kommer vi att diskutera den metod vi använt oss av i vårt arbete, och vilka fortsatta studier av mobil datoranvändning som kan vara intressanta.

10.1 Metoddiskussion

Vi har i vårt arbete använt oss av etablerade teorier inom området gränssnittsdesign och tillämpat dessa på mobila klienter i form av mobiltelefoner. Det finns en mängd riktlinjer för gränssnittsdesign, och vi ville ta reda på vilka som går att använda när man utvecklar tillämpningar för mobiltelefoner. För att få en djupare förståelse för de problem som finns vid utveckling av sådana tillämpningar valde vi att skapa en prototyp med hjälp av WAP. Vi tillämpar alltså generella teorier för gränssnittsdesign på en specifik teknik, men trots detta är de slutsatser vi drar giltiga även om man arbetar med andra tekniker än WAP vid utveckling för mobiltelefoner. Att det förhåller sig så beror på att gränssnittet för mobiltelefon ser likadant ut oavsett vilken teknik som används vid utvecklingen. För att undersöka om riktlinjerna medfört ett önskvärt resultat gjorde vi en enkätundersökning som visade att det förhåller sig så.

I avsnitt 1.4 visar vi vårt fullständiga tillvägagångssätt, men lite förenklat kan man säga att vi börjar i ett teoretiskt resonemang om vilka riktlinjer som bör tillämpas för att sedan med hjälp av en praktisk undersökning utreda om så är fallet. Vi drar sedan slutsatsen att det verkligen går att tillämpa dessa riktlinjer även vid utveckling för mobiltelefoner och kopplar på så sätt tillbaka till teorin. Detta kan illustreras med Figur 16 som visar hur vårt arbete rör sig från teori till praktik och sedan tillbaka.



Figur 16. Växelspelet mellan teori och praktik

Vi anser att metoden har varit väl anpassad för vårt ändamål och att vi med hjälp av denna nått en god förståelse för hur man kan utveckla tjänster för mobila klienter.

10.2 Fortsatta studier

Det finns en mängd intressanta områden med anknytning till WAP-tekniken som lämpar sig för vidare studier.

I vårt arbete har vi jobbat med version 1.1 av WAP-specifikationen, men nu är version 1.2 under införande och den innehåller en del intressanta förändringar. Möjligheten att utveckla avancerade telefonitjänster med hjälp av WTAI tar vi inte alls upp i vårt arbete och dessa har utvecklats en hel del i version 1.2. Detta skulle kunna vara en intressant aspekt att undersöka djupare.

En annan aspekt som kan vara av intresse är användarens syn på mobiltelefonen som klient för olika tjänster.

Som vi nämnt i vår avgränsning utgör säkerhetsaspekten på WAP-tekniken en uppsats i sig själv – en uppsats som ännu är oskriven.

Det kan också vara intressant att undersöka hur mobila klienter kommer att integreras med de befintliga system som finns i en organisation. Att undersöka vilka möjligheter och problem som uppstår när mobila klienter får en större betydelse för verksamheten helt enkelt.

11 Referenser

- [ALBE95] "Designers Guide to the Internet" Rick Albertson et al, 1995, Hayden Books, ISBN: 1-56830-229-0
- [AUWP99] "WAP White Paper" AU-Systems, februari 1999
- [BRG99] "Life starts at megabit speed" Anders Berg, BRG, Januari 2000
http://www.brg.se/sime99_fernstedt.htm
- [CA0399] "WAP – här är svaren" Calling nr 3, 1999
- [CERT00] "SECG Letter to NIST on ECDSA inclusion in FIPS 186-1", Certicom, Januari 2000 http://www.secg.org/collateral/SECG_FIPS_ECDSA_Ltr.pdf
- [COX93] "User-Interface Design" Kevin Cox och David Walker, 1993, Prentice Hall, ISBN: 0-13-952888-1
- [CS10299] "Trådlöst" Computer Sweden nr 102, 1999
- [CS9899] "DHL tog fram wap-tjänst rekordsnabbt" Karl-Johan Byttner, Computer Sweden nr 98, 1999
- [CS9599] "Internet Cellular Smart Access" Computer Sweden nr 95, 1999
- [CS10799] "WAP tynar bort inom två år" Joel Åsblom, Computer Sweden nr 107, 1999
- [CS9299] "Hysteri kring fyra rader text" Bo Nordlin, Computer Sweden nr 92, 1999
- [CS11399] "Trådlöst" Computer Sweden nr 113, 1999
- [CS600] "Scandics bonuskunder bokar via wap" Joel Åsblom, Computer Sweden nr 6, 2000
- [CS300] "Wap ger dig ständig uppgraderingsångest" Henrik Jakobsson, Computer Sweden nr 3, 2000
- [DATA99] "Ingen hjälp att få för Wap-utvecklare" Datateknik nr 4, 1999
- [DATR0] "Kryptering" DataRgistrering Hå-Es, Januari 2000
<http://www.datareg-hs.y.se/fortj.htm>
- [DAVI96] "Securing Client/Server Computer Networks" Peter T. Davis, 1996, McGraw Hill, ISBN: 0-07-015841-X
- [EDGE00] "WAPman" The Edge Consultants, Januari 2000
<http://virtuacom.com/wap>
- [ENTR00] "Wireless Products" Entrust Technologies, Januari 2000
<http://www.entrust.com/products/wap/index.htm>

- [FORG94] "Developing Cooperative and Client Server Systems" Simon Forge, 1994, McGraw-Hill, ISBN: 0-07-707878-0
- [FORS98] "Human Factors and Web Development" edited by Chris Forsythe et al, 1998, Lawrence Erlbaum Associates inc, ISBN: 0-8058-2824-9
- [HILL99] "WAP: Mobil Internet" Mikael Hillborg, 1999, Pagina Förlag AB, ISBN: 91-636-0600-3
- [KRIS99] "Designing Interaction Styles for a Mobile Use Context" Steinar Kristoffersen och Fredrik Ljungberg, 1999, Viktoriainstitutet
- [LUTZ00] "What is Diffie-Hellman?" Lutz Donnerhacke, Januari 2000-01-20
<http://www.iks-jena.de/mitarb/lutz/security/cryptfaq/q24.html>
- [MOBI99] "Data on WAP" Mobile Lifestreams ltd, 1 augusti 1999
- [NOK001] "WAP – The Corporate Perspective" Nokia, Januari 2000
<http://www.nokia.com/wap>
- [OSBY00] "SIM-kort" OsbyMikro, Januari 2000
<http://home.swipnet.se/OsbyMikro/sims.htm>
- [PHON00] "The Wireless Application Protocol" Phone.com, Januari 2000
<http://www.phone.com/pub/feb99WAPWP.pdf>
- [PREE94] "Human-Computer Interaction" Jenny Preece et al, 1994, Addison-Wesley, ISBN: 0-201-62769-8
- [PRES94] "Software Engineering" Roger S. Pressman, 1994, McGraw-Hill, ISBN: 0-07-707936-1
- [WAP001] "WAP White paper" WAP-Forum, Januari 2000
http://www.wapforum.org/what/WAP_white_pages.pdf
- [WAP002] "WAP – Frequently asked questions" WAP-Forum, Januari 2000
<http://www.wapforum.org/faqs/index.htm>
- [WAPF99] "Official Wireless Application Protocol" WAP-Forum ltd, 1999, John Wiley & Sons Inc, ISBN 0-471-32755-7
- [WIRE00] "Phones of the future" Wireless Review, Januari 2000
<http://www.wirelessreview.com/issues/1999/90315/feat21.htm>
- [WWW00] "Säker handel med WWW" WWW Report, Januari 2000
http://www.dtek.chalmers.se/~d0dak/WWWReport/WWWBook.book_61.html
- [WÄRN90] "Att fråga" Bo Wärneryd et al, 1990, Statistiska Centralbyrån, ISBN 91-618-0382-0

12 Förkortningar och ordförklaring

ASP – Active Server Pages
CDMA – Code Division Multiple Access
CGI – Common Gateway Interface
CSD – Circuit Switched Data
ECDSA – Elliptic Curve Digital Algorithm
GPRS – General Packet Radio Service
GSM – Global System for Mobile communication
HTML – HyperText Markup Language
HTTP – HyperText Transfer Protocol
Kbs – Kilobits per second
LDAP – Lightweight Directory Access Protocol
Mbs – Megabits per second
MExE – Mobile Station Application Execution Environment
OSI – Open Systems Interconnection
PDA – Personal Digital Assistant
RAD – Rapid Application Development
RAM – Random Access Memory
ROM – Read Only Memory
SCB – Statistiska CentralByrån
SIM – Subscriber Identity Module
SMS – Short Messages Service
SQL – Structured Query Language
SSL – Secure Socket Layer
TCP/IP – Transport Control Protocol/Internet Protocol
TDMA – Time Division Multiple Access
TLS – Transport Layer Security
UDP – User Datagram Protocol
UMTS – Universal Mobile Telecommunications System
URL – Uniform Resource Locator
URN – Uniform Resource Name
USSD – Unstructured Supplementary Services Data
W3C – World Wide Web Consortium
WAE – Wireless Application Environment
WAP – Wireless Application Protocol
WDP – Wireless Data Protocol
WML – Wireless Markup Language
WSP – Wireless Session Protocol
WTA – Wireless Telephony Applications
WTAI – Wireless Telephony Application Interface
WTP – Wireless Transaction Protocol
WTLS – Wireless Transaction Layer Security
WWW – World Wide Web
XML – eXtended Markup Language

Denial-of-service attack – Sabotage mot en server genom att så många meddelanden sänds till servern att denne inte kan hantera dem alla och därför slutar att fungera.

Bilaga A. Mobile Station Application Execution Environment (MExE)

Det är vida förutspått och förväntat att mobiltelefoner kommer att få mer kraft och ökad funktionalitet över de kommande åren. Telefonerna kommer förmodligen att stödja överföring av video och andra liknande funktioner. Sådana sofistikerade telefoner kommer förmodligen att kräva ett mer sofistikerat trådlöst protokoll än WAP för att kunna använda dem. Dagens WAP protokoll används för de begränsade telefonerna som stödjer den. MExE däremot är framtiden.

Målet med Mobile Station Application Execution Environment (MExE) är att erbjuda en omfattande och standardiserad miljö för mobiltelefoner för att utveckla applikationer för mobiltelefoner. MExE är gjort för att kunna "köra" applikationer på mobila klienter och erbjuda intelligenta kundmenyer och avancerade nätverkstjänster. Med MExE tekniken planerar man även att kunna integrera en positionsbestämning för mobiltelefoner. Tekniken stödjer också många människa-maskin gränssnitt så som röstigenkänning, ikoner och softekeys. På grund av allprogrammeringsmöjligheter som tekniken ger måste den också involvera strikta säkerhetsregler för att förhindra att obehöriga kommer åt användarens data. MExE har dock en hel del gemensamt med WAP. Bägge protokollen har designats för att fungera med olika tjänster i GSM nätverken, allt från SMS till Gprs och senare även med UMTS. WAP stödjer en del användning av scripts, grafik animering och text medan MExE tillhanda håller ett fullständigt programmeringsspråk.

MExE bygger en Java Virtuellt Maskin i klienten, mobiltelefonen. Java som programmeringsspråk fungerar som så att man skriver en gång och kan sedan exekvera var som helst. För att kunna programmera och köra java krävs ganska mycket processorkraft av den mobila klienten, vilket gör att MExE främst är kommer att kunna användas på nästa generation av avancerade mobiltelefoner.

Men MExE klienter skulle också kunna inkludera dagens vanliga telefoner eftersom MExE integrerar en metod för att identifiera olika typer som kallas Classmark. Denna metod definierar de MExE relaterade tjänsterna som en specifik klient stödjer. Det finns så kallade Classmarks som matchar och en del som överskrider WAPs funktionalitet. Den mobila MEXE klienten kan informera MExE servern om dess Classmarks och vilket berättat vilken funktionalitet den har.

Data on WAP, 1 augusti 1999, Mobile Lifestreams ltd

Bilaga B. Utvecklingsmiljöer

WAP Developer Toolkit 1.0 av Dynamic Systems Research Ltd

Plattformsberoende, Java-baserad implementation som gör det möjligt att utveckla WAP-tjänster på PC, Unix och Macintosh förutsatt att platformen stöder Java fullt ut. WAP Developer Toolkit kostar 589 GBP för en licens, och 199 GBP för ytterligare licenser.

Version 1.0 innehåller:

- WAP-läsare
- WML Encoder/Decoder
- WMLScript Compiler
- WMLScript Assembler/Disassembler
- WMLScript Virtual Machine
- WML-exempel
- WMLScript-exempel
- Komplet implementation av WMLScript Standard Libraries
- Extra WMLScript Debug Library
- Dokumentation för WML, WMLScript och utvecklingsmiljön

Källa: <http://www.wap.net/devkit>

Unwired Planet Software Development Kit av Phone.com

UP SDK innehåller en emulator för att testa WML-tjänster. Emulatorn kan antingen användas lokalt eller kopplas mot en UP-link för att testa tjänsten i en skarp miljö. Phone.com erbjuder en publik UP-link utan någon kostnad för att ge utvecklare möjlighet att skapa tjänster även om de inte har tillgång till en WAP-terminal. De tjänster som utvecklas kan användas på en vanlig webserver, det behövs ingen licens för att använda UP SDK.

UP SDK innehåller:

- UP.Simulator, en applikation som emulerar en telefon från Unwired Planet och som kan användas för att testa WML-tjänster. Fungerar under Windows 95/NT.
- Perl- och C-bibliotek som underlättar skapandet av WML-kod och hanteringen av HTTP-förfrågningar.
- Verktyg i C++ (Solaris) och COM (Windows) för att hantera fax och meddelanden.
- Verktyg för SSL (Secure Socket Layer).
- Exempel på WML-kod och WMLScript.
- Dokumentation i HTML-format

Källa: <http://updev.phone.com>

Portal-To-Go av Oracle

Oracles Portal-To-Go är en ny server som kommer att revolutionera användningen av mobila klienter som Palm, WinCE och mobiltelefoner med internetkoppling. Genom att konvertera befintlig information till ett format för en specifik klient blir det möjligt att komma åt avancerade web-tjänster från i stort sett vilken klient som helst. Förutom att konvertera information mellan olika format innehåller Portal-To-Go en portalfunktion som kan anpassas

Bilaga B

för varje enskild användare. Portalfunktionen ger användaren möjlighet att definiera, organisera och anpassa olika tjänster för en mobil klient.

Problemet

Det grundläggande problemet med internet är att dagens webssiter är skapade för att beskådas med den senaste versionen av Netscape Navigator eller Internet Explorer på en färgskärm. Alla som har försökt komma åt en webssite med hjälp av en PDA vet att det antingen inte fungerar överhuvudtaget eftersom webläsaren på en PDA inte kan hantera det komplexa innehållet eller att innehållet inte kan visas till fullo på en så liten skärm som PDA:n har. Mobiltelefoner innebär ett annat problem. Även om WAP löser problemet med att koppla telefonen till internet och hantera problemet med den begränsade bandbredden uppstår ett nytt problem eftersom innehållet på de flesta webssiter inte är anpassat för WAP. Detta innebär en moment 22-situation. Vem kommer att köpa WAP-telefoner om det inte finns några tjänster för dem och vem kommer att skapa tjänster om det inte finns några telefoner? Ett annat problem för både PDA:s och telefoner är att de saknar ett riktigt tangentbord vilket gör det svårare att skriva in information. Den vanligaste inmatningen vid internetanvändning är att mata in URL:s vilket är rätt svårt med mobiltelefonens knappsats.

Befintliga lösningar

Lösningen på problemet med innehållet är att skapa speciella versioner av webssiter för varje typ av klient. Eftersom väldigt få webssiter finns i flera versioner innebär detta att de mobila klienterna i realiteten bara har tillgång till en begränsad del av internet. Det finns ingen bra lösning på problemet med inmatning av information utan det är något man för ta sig tid att göra för att komma åt innehållet på de olika webssiterna.

Oracles lösning

Oracles Portal-To-Go har en lösning på samtliga ovanstående problem och ger de mobila klienterna utökade möjligheter att komma åt webssiter. Portal-To-Go kan göra en dynamisk översättning av befintliga webssiter till ett generellt XML-format, och sedan generera en specifik version för varje klient. Detta angreppssätt gör det mycket enkelt att skapa tjänster för olika typer av klienter. Förutom att översätta innehåll utgör Portal-To-Go en personlig portal som är den naturliga startpunkten för mobila klienters internetanvändning.

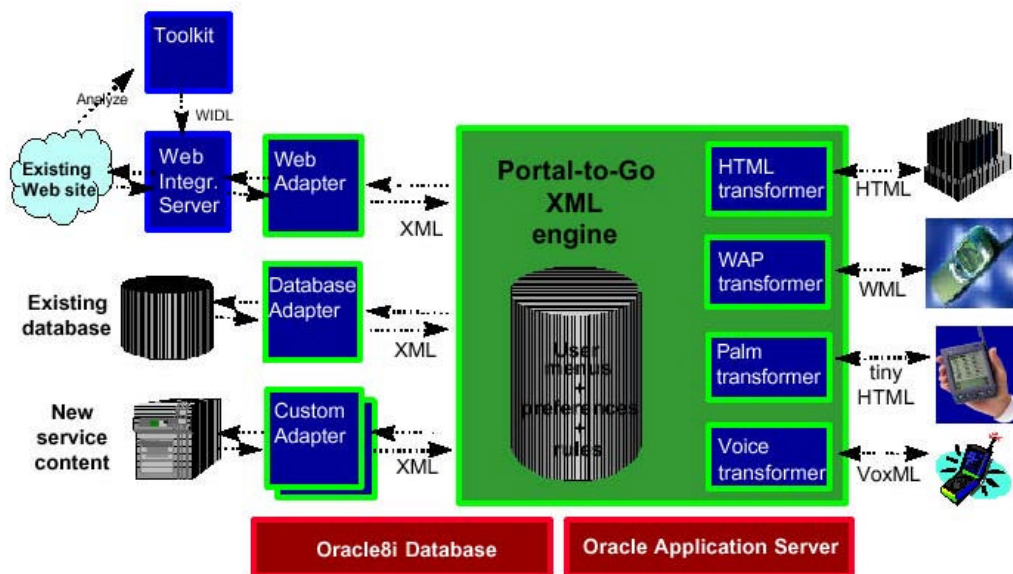
Att innehållet specificeras i XML gör det möjligt att översätta det till olika hypertextspråk från WML för mobila klienter via HTML för vanliga webläsare till VoXML för interaktiva rösttjänster. Detta innebär att när tjänsten väl har skapats spelar det ingen roll om användaren har en mobiltelefon, en PDA eller en PC. Skapandet av en tjänst är en enkel process som kan utföras av en systemadministratör för avancerade, transaktionsbaserade tjänster eller av slutanvändaren för enklare tjänster.

Portalfunktionen utgör en startsida som användaren kan anpassa med hjälp av en webläsare. En stor del av den information som matas in vid internetanvändning är personlig information som epostadresser, lösenord och liknande. Användaren kan välja vilken information som ska matas in med hjälp av den mobila klienten och vad som ska lagras på portalen för att användaren ska slippa skriva in det gång på gång.

Portal-To-Go innehåller också gränssnitt mot andra typer av informationstjänster som till exempel vanliga databaser. Eftersom Portal-To-Go innehåller information om de olika hypertextspråken behöver inte utvecklaren ha någon kunskap om till exempel WML. När nya typer av klienter eller hypertextspråk dyker upp laddar man bara hem information om dessa

Bilaga B

och laddar in dem i Portal-To-Go för att stödja dessa nya tekniker utan att skriva om befintliga tjänster eller portalsidor. Bilden nedan visar en översikt av Portal-To-Go.



Portal-to-Go Architecture

Källa: Oracle Portal-To-Go, An Oracle Business White Paper, October 1999,
<http://www.oracle.com>

Utvecklingsverktyg från Ericsson

Ericsson erbjuder en integrerad utvecklingsmiljö, en WAP Gateway och en applikationsserver för WAP-tjänster.

WapIDE (WAP Integrated Development Environment)

WapIDE är en fullständig utvecklingsmiljö som innehåller en WAP-klient, en HTTP-klient och en filklient. Detta innebär att utvecklaren kan testa applikationer på en webserver, WAP-gateway eller lokal hårddisk. WapIDE ger utvecklaren möjlighet att testa WML-kod på ett flertal simulerade klienter med olika möjligheter i form av grafisk kapacitet och skärmstorlek. WapIDE innehåller Device Designer som gör det möjligt att definiera egna klienter. Det kan vara användbart för att emulera nya typer av klienter. WapIDE innehåller också Application Designer för att utveckla WML och WMLScript. Utvecklingsmiljön integrerar valfri klient och WML-editor och kan växla mellan design- och testläge. WapIDE består av en WAP-läsare, Device Designer, Application Designer och serververktyg.

WAP-läsare

Testa och visa WML-kod och WMLScript med en simulerad WAP-klient. Innehåller:

- Grafiskt användargränssnitt.
- WML-tolk
- WMLScript-tolk
- WAP-klient för att ladda in kod från en WAP-gateway
- HTTP-klient för att ladda in kod från en webserver
- Filklient för att ladda in kod från lokal hårddisk

Device Designer

Ett verktyg för att simulera beteende och utseende hos en WAP-klient. Innehåller:

- Funktioner för att simulera utseende
 - Använd en bifogad bild på klienten eller importera en egen
 - Definiera knappar på klienten med olika lägen
 - Definiera typ av display och dimensioner
- Funktioner för att simulera beteende
 - Definiera vad olika knappar har för funktion
 - Definiera vad lysdioder har för funktion
 - Definiera hur text ska skrolla

Application Designer

Ett verktyg för att skapa WML-kod. Innehåller:

- Visar både WML-kod och tjänstens utseende
- Online hjälp
- Menyer för att infoga WML-element

Serververktyg

- Skapa dynamiska WAP-tjänster med hjälp av CGI-script.
- Kompilera WML-kod och WMLScript för snabbare åtkomst.
- Inkluderar CGI-script för att hantera HTTP-headers och parametrar.
- Stöd för Java-servlets.

WapIDE är en plattformsoberoende utvecklingsmiljö som kan användas på UNIX, Linux, Windows 95/98 och Windows NT.

WAP Gateway/Proxy

WAP Gateway/Proxy skapar en koppling till GSM-nätet med hjälp av WAP-protokollet. Den innehåller följande lager från WAP-stacken:

- Wireless Session Protocol (WSP)
- Wireless Transaction Protocol (WTP)
- Wireless Transport Layer Security (WTLS)
- Wireless Datagram Protocol (WDP)
- Bearer services (USSD, SMS; any IP bearer such as CSD, GPRS)

Mot intranät eller internet agerar WAP Gateway/Proxy som en klient med HTTP, SSL och TCP/IP. Förutom att skapa en koppling mellan internet och mobiltelefonnätet innehåller den också tjänster som:

- Autenticering av användare
- Kompilering av WMLScript
- Caching

WAP Gateway/Proxy säljs som en helhetslösning med både hård- och mjukvara. För att implementera Wap-tjänster finns också WAP Application Server som interagerar med WAP Gateway/Proxy. Applikationsservern innehåller funktioner för epost, konvertering av HTML till WML och API:s mot till exempel JDBC.

Källa: <http://www.ericsson.com/WAP/developer>

Utvecklingsverktyg från Nokia

Nokia erbjuder en integrerad utvecklingsmiljö och en WAP server.

Nokia WAP Toolkit

Nokia WAP Toolkit version 1.2 innehåller support för WAP specifikation 1.1. Denna version gör det möjligt att emulera en WAP telefons funktionalitet mot en Nokia WAP Server som förövrigt finns tillgänglig som en försöksversion på Nokias WAP Developer Forum.

Nokia WAP Toolkit 1.2 Specifikationer

WAP Toolkit komponenter

Kompilatorer

- WML-kompilator
- WMLScript-kompilator

Simulatorer

Micro-browser

- Script tolkare och run-time bibliotek
- En modul för telefonens gränssnitt

Debugger verktyg

On-line dokumentation

- Nokia WAP Toolkit Getting Started
- Nokia WAP Toolkit Developer's Guide
- WML Reference
- WMLScript Reference

Exempel applikationer

System krav

Applikationer kan utvecklas på en PC plattform som har följande utrustning:

- Pentium 266 MHz eller snabbare
- 64 MB RAM
- Windows NT 4.0 med SP3, Windows 95, eller Windows 98
- 16 bitars färg
- 1024*768 upplösning
- 20 MB ledigt hårddiskutrymme
- Java™ Run Time Environment 1.2.2 (eller senare)

Nokia WAP Server

Nokias WAP Server Trial version 1.0 finns just nu tillgänglig att hämta gratis för alla som är intresserade av att utveckla WAP applikationer och WAP tjänster.

Utveckla ditt företags trådlösa data tjänster med Nokias WAP Server. Nokias WAP Server kan erbjuda en trådlös förbindelse för existerande Intra- eller Extranet så väl som för företagets egna system. Man kan kryptera företagets data med Nokias WAP Server Security Pack, och du kan behålla den konfidentialitet som du och dina kunder har och även skapa nya nivåer av kundintimitet i företaget.

Nokias WAP Server följer strikt den öppna industri standarden, WAP specifikation 1.1 och dess tjänster som exempelvis en mängd olika trådlösa bärare, uppkopplingen mot

Bilaga B

internetservrar och dessutom företags applikationer genom Application Programming Interface (API).

Tjänster och systemkrav

Öppenhet och lätt att använda karakteriserar Nokias WAP Server. Tjänsterna är listade nedan.

- Anslutning:
 - WAP 1.1 compliant

- UDP/IP för:
 - GSM Circuit Switched Data

- SMS Center:
 - Nokia SMSC via CIMD 2.0 protocol
 - CMG SMSC via UCP 2.4 protocol

- Serverfunktionalitet:
 - Användaridentifikation och tillgänglighetskontroll
 - Lagring och exekvering av applikationer
 - Öppet programmeringsgränssnitt (API) för back-end integration

- Innehållsfunktioner:
 - WML and WMLScript text eller binärt
 - HTML och text omvandling till WML
 - Access till HTTP 1.1 och HTTP 1.0 servrar
 - Access till lokala filsystem
 - Access till en tredje parts system genom Nokias WAP Server API
 - Nokia WAP Server API
 - Java™ Servlet API
 - WAP Service API för WAP protokoll data
 - Server Extension API för att konfigurera WAP server
 - Data push med SMS

- Krav på datormiljön:
 - Pentium Processor, minst 266 MHz
 - 128 MB RAM, 256 MB RAM rekommenderas
 - Minst 100 MB ledigt hårddiskutrymme
 - Windows NT™ 4.0 Service pack 5
 - Java™ Runtime Environment 1.2.2
 - Java™ HotSpot 1.0.1

- Wireless Transport Layer Security:
 - Två versioner finns tillgängliga som separata produktval till Nokias WAP Server: 56-bitars och 128-bitars nyckel längds krypteri

Bilaga C. Enkät

Fråga	Person1	Person2	Person3	Person4	
I vilken utsträckning är du bekant med tur och retur?		80	30	70	100
I vilken utsträckning är du bekant med webversionen av tur och retur?		0	0	10	100
Anser du att prototypen ger ett bra helhetsintryck?		40	30	70	90
Anser du att tjänsten utlägg fungerar på ett likvärdigt sätt i prototypen som i den ursprungliga applikationen?		50	40	70	70
Tycker du att det framgår vilken inmatning som förväntas i de olika fälten? Motivera	Enheten kunde visa default text "Bilersättning [km]"	Vet ej	Ja, om man kan tur och retur	Ja, ledtext. Eventuellt bör man förtydliga vilken typ av data som förväntas	
Ger prototypen tillräcklig feedback på utförda operationer? Motivera	Ja	Vet ej		Ja, bekräftelse på inmatning	
Har prototypen en logisk arbetsgång? Motivera	Ja	Vet ej	Ja, utifrån förutsättningarna	Någotsånär, liten skärmyta	
Är det lätt att veta var i applikationen man befinner sig? Motivera	Ja, om man är bekant. En sitemap kanske kunde passa	Vet ej		Verkar lite stökigt, och mycket knapptryck	
Är det för mycket eller för litet information på varje skärmbild? Motivera	Nej	Vet ej	Nej	Nej	
Är det lätt att ångra en felaktig inmatning? Motivera	Vet ej	Vet ej		Vet ej	

Övriga åsikter

Bilaga C

Fråga	Person5	Person6	Person7	Person8	
I vilken utsträckning är du bekant med tur och retur?		75	40	80	100
I vilken utsträckning är du bekant med webversionen av tur och retur?		65	0	0	100
Anser du att prototypen ger ett bra helhetsintryck?		75	70	50	100
Anser du att tjänsten utlägg fungerar på ett likvärdigt sätt i prototypen som i den ursprungliga applikationen?		65	70	50	70
Tycker du att det framgår vilken inmatning som förväntas i de olika fälten? Motivera	Ja, enkla "talande" ledtexter	Det står ju	Ja	Ja, vet ej om det kan stödja rimlighetskontroller etc. Mycket bra	
Ger prototypen tillräcklig feedback på utförda operationer? Motivera	Ja	Ja, allt avslutas med en total sammanställning	Ja, men tidsfördröjningarna är långa. Dock ej prototypens fel	Ja, kontrollfrågor finns om inmatad info är korrekt	
Har prototypen en logisk arbetsgång? Motivera	Ja	Ja, frågor kommer i rätt ordning	Ja	Ja, enkel och strukturerad arbetsgång	
Är det lätt att veta var i applikationen man befinner sig? Motivera	Så där, beror på om man kan webb-versionen	Nja, displayen begränsar överblick	Ja	Ja, eftersom den är logisk	
Är det för mycket eller för litet information på varje skärmbild? Motivera	Nej	Det kunde vara mer	Nja, lite rörigt	Precis det som krävs, varken mer eller mindre	
Är det lätt att ångra en felaktig inmatning? Motivera	Nej	Ja, det är lätt	Ja	Nej, inte i prototypen. Men det är fullt möjligt att hantera	
Övriga åsikter	Bra utifrån de möjligheter som WAP ger	Intressant teknik. Början på något nytt		Sammanfattningsvis ett intressant arbete som går att vidareutveckla	

Bilaga C

Fråga	Person9	Person10	Person11
I vilken utsträckning är du bekant med tur och retur?		100	80 100
I vilken utsträckning är du bekant med webversionen av tur och retur?		20	0 100
Anser du att prototypen ger ett bra helhetsintryck?		80	60 100
Anser du att tjänsten utlägg fungerar på ett likvärdigt sätt i prototypen som i den ursprungliga applikationen?		80	60 70
Tycker du att det framgår vilken inmatning som förväntas i de olika fälten? Motivera		Ja	Ja
Ger prototypen tillräcklig feedback på utförda operationer? Motivera	Ja	Ja	Ja
Har prototypen en logisk arbetsgång? Motivera	Ja	Ja	Ja
Är det lätt att veta var i applikationen man befinner sig? Motivera	Ja	Hade varit bra med en rubrik	Ja
Är det för mycket eller för litet information på varje skärmbild? Motivera	Ja	Nej, men bilden hade kunnat vara större	Nej
Är det lätt att ångra en felaktig inmatning? Motivera	Nej	Ja	Nja, bättre att välja ja eller nej för godkännande eller underkännande
Övriga åsikter	Intressant och spännande	Demon visar på begränsningarna i WAP-standarden och den begränsade bildskärmsstorleken	