

Examensarbete

WAP och röstbrevlådetjänster

av

Daniel Bratell

LiTH-IDA-Ex-00/20

2000-02-21

Examensarbete

WAP och röstbrevlådetjänster

av

Daniel Bratell

LiTH-IDA-Ex-00/20

2000-02-21

Handledare: Morgan Filipsson

Examinator: Lars Ahrenberg

Sammanfattning

WAP, Wireless Application Protocol, är en metod för att koppla ihop datornätverk och mobiltelefoni. Tanken bakom den ihopkopplingen är att kunna erbjuda användare av mobiltelefoner de tjänster som redan finns på Internet direkt in i mobiltelefonerna. Den fråga som Epacket Technology AB ville ha svar på var om det är möjligt att använda WAP till att förbättra röstbrevlådor såsom till exempel Telias MobilSvar.

Frågan om huruvida det är möjligt att använda WAP till en röstbrevlådetjänst visade sig kunna besvaras med både ja och nej beroende på tidsperspektivet och vilka krav som ställs på tjänsten. I den version av standarden som varit den aktuella under examensarbetet, WAP 1.1, saknas en del nyckelkomponenter för att möjliggöra växling mellan data och tal. Det innebär att en användare inte kan lyssna av meddelanden eller spela in hälsningsmeddelanden på något enkelt sätt och saknas de funktionerna är det inte heller möjligt att tillverka en WAP-röstbrevlåda. Det som kan byggas av de tillgängliga funktionerna är ett administrations- och övervakningsgränssnitt till en röstbrevlåda så att användarna kan göra inställningar och se vilka meddelanden som finns i röstbrevlådan. Det byggdes en prototyp-tjänst med de funktionerna under examensarbetets gång vilket identifierade en del andra problem såsom cachningsproblem, användaridentifieringsproblem med mera, men de fick alla sin lösning under arbetets gång.

Under arbetet med den här rapporten pågick utarbetandet av en ny version av standarden som skulle bli WAP 1.2. Det har funnits ett förslag till standard, vilket senare blev det som fastslogs, som jag har studerat under det här examensarbetet. I WAP 1.2 ingår de nyckeldelar som saknades i WAP 1.1 som bland annat gör det möjligt att komma åt telefonifunktioner från en WAP-sida. Med WAP 1.2 ska det i teorin vara fullt möjligt att bygga en tjänst som har samma funktionalitet som de traditionella röstbrevlådorna har idag, men först krävs telefoner och serversystem som har stöd för den nya versionen av standarden. Möjligtvis kommer vissa prestandakrav göra det intressant att vänta på GPRS som är den utvecklade formen av GSM som förväntas lanseras under år 2000 eller 2001.

Abstract

WAP, Wireless Application, Protocol, is a new system that has as one of its goals to connect mobile telephony and computer networks, particularly Internet. E pact Technology AB therefore wanted to study if it was possible to extend existing voice mail services, such as Telia's MobilSvar, with a WAP interface.

The study showed that it depended on the time perspective and ambition if it was possible to make a voice mail service with WAP. In the current specification, WAP 1.1, there are some key parts missing that are needed to switch between data connection and voice connection. Without that possibility it is very difficult to listen to recorded messages and to record a greeting message. It is still possible though to create an administration interface to the voice mail so that you can change your preferences and list messages in the message box. To explore those possibilities a prototype service was built with exactly those features, which identified some other problem areas that were solved during the work. That included problems with the phone's cache and authentication problems.

During the work on this report, WAP Forum was working on a new version of the standard and a proposed specification, which later has become the approved specification, was available during this time. This proposed specification was included in the study. In that proposition the exact key parts that were missing in WAP 1.1 have been included, so with WAP 1.2 it should be theoretically possible to build a service with at least the same functionality as the traditional voice mail services have today, if only the necessary hardware such as mobile phones and servers are built with WAP 1.2 support. It might be interesting to wait for GPRS, which is a new version of GSM that will be released in the year 2000 or 2001, in order to improve the performance of the service.

Förord

Jag vill tacka alla som på något sätt har bidragit till den här rapporten.

Ni vet själva vilka ni är.

Innehållsförteckning

1	INLEDNING	1
2	PROBLEMSTÄLLNING.....	3
2.1	EPACT TECHNOLOGY AB	3
2.2	MOTIVERING TILL EXAMENSARBETET	3
2.2.1	<i>Röstbrevlådeöversikt.....</i>	<i>3</i>
2.2.2	<i>Målsättning för en WAP-tjänst</i>	<i>5</i>
3	METOD	7
3.1	METODVAL.....	7
3.2	GENOMFÖRANDE.....	10
3.3	KÄLLOR.....	11
4	WAP-ÖVERSIKT.....	13
4.1	HISTORIK.....	13
4.2	SYFTE MED WAP.....	13
4.3	ÖVERFÖRINGSOPTIMERING.....	14
4.3.1	<i>Bytekodning</i>	<i>14</i>
4.3.2	<i>Kortlek</i>	<i>14</i>
4.4	WML	15
4.5	WMLSCRIPT	16
4.6	PROTOKOLLSTACKEN	18
4.6.1	<i>Bärare.....</i>	<i>18</i>
4.6.2	<i>WDP.....</i>	<i>19</i>
4.6.3	<i>WTLS</i>	<i>20</i>
4.6.4	<i>WTP.....</i>	<i>20</i>
4.6.5	<i>WSP</i>	<i>20</i>
4.6.6	<i>WAE.....</i>	<i>21</i>
4.7	INFRASTRUKTUR.....	21
4.8	MOBILTELEFONER OCH ANDRA WAP-KLIENTER	22
4.8.1	<i>Ericsson MC218 kopplad till en Ericssontelefon.....</i>	<i>23</i>
4.8.2	<i>Siemens S25</i>	<i>24</i>
4.8.3	<i>Nokia 7110.....</i>	<i>24</i>
4.8.4	<i>Ericsson R320s</i>	<i>25</i>
4.8.5	<i>Ericsson R380s</i>	<i>25</i>
4.8.6	<i>3Coms Palm Pilot.....</i>	<i>26</i>
4.8.7	<i>Simulatorer</i>	<i>26</i>
4.9	WAP-GATEWAY	28
4.10	WAP SOM FRAMTIDSTEKNIK.....	29
4.11	SAMMANFATTNING.....	29
5	RÖSTBREVLÅDA FÖR WAP-TELEFONER.....	31

5.1	KOPPLING TILL EXISTERANDE SYSTEM.....	31
5.1.1	<i>Kommunikation via ljudenheten</i>	32
5.1.2	<i>Kommunikation genom styrenheten</i>	32
5.1.3	<i>Styrning direkt genom databasen</i>	33
5.1.4	<i>Sammanfattning</i>	33
5.2	IDAG – WAP 1.1.....	34
5.2.1	<i>Systemförslag</i>	35
5.2.2	<i>Bärare</i>	37
5.2.3	<i>Upplevd prestanda</i>	38
5.2.4	<i>Inställningar</i>	38
5.2.5	<i>Meddelandelistning</i>	38
5.2.6	<i>Meddelandeavisering</i>	39
5.2.7	<i>Meddelandeavlyssning</i>	39
5.2.8	<i>Inspelning av hälsningsmeddelande</i>	39
5.2.9	<i>Identifiering</i>	40
5.2.10	<i>Cachningsproblem</i>	40
5.2.11	<i>Säkerhet</i>	41
5.2.12	<i>Sammanfattning</i>	42
5.3	FÖRÄNDRINGAR NÄR WAP 1.2 BLIR ALLMÄNT TILLGÄNGLIG.....	42
5.3.1	<i>Systemförslag</i>	43
5.3.2	<i>WTA/WTAI – Meddelandeavlyssning</i>	44
5.3.3	<i>”Push” – Meddelandeavisering</i>	47
5.3.4	<i>Säkerhet</i>	47
5.3.5	<i>Alternativa tekniker</i>	48
5.3.6	<i>Sammanfattning</i>	48
5.4	EFTER WAP 1.2.....	48
5.5	SAMMANFATTNING.....	49
6	KVARVARANDE PROBLEM.....	51
6.1	<i>STORLEKSBEGRÄNSNING AV KORTLEKAR</i>	51
6.2	<i>ANVÄNDARGRÄNSSNITT</i>	51
6.3	<i>STORDRIFT</i>	55
6.4	<i>MARKNADEN</i>	56
6.5	<i>SAMMANFATTNING</i>	56
7	ORDFÖRKLARINGAR OCH FÖRKORTNINGAR.....	57
8	REFERENSER.....	61
9	INDEX.....	64
10	BILAGA A – PROTOTYPEN.....	67
11	BILAGA B.....	71

Figur-, tabell- och exempelförteckning

Figurer

Figur 1.	Navigering i en traditionell röstbrevlåda.....	4
Figur 2	Menystruktur i en traditionell röstbrevlåda.....	5
Figur 3.	Vattenfallsmodellen.....	7
Figur 4.	Den iterativa vattenfallsmodellen.....	8
Figur 5.	Boehms spiralmodell.....	8
Figur 6.	WAP-stacken jämförd med TCP/IP-stacken.....	18
Figur 7.	Infrastruktur i WAP.....	21
Figur 8.	Ericsson MC218 tillsammans med en mobiltelefon Ericsson T28.....	23
Figur 9.	Siemens S25 (© Siemens AG).....	24
Figur 10.	Nokia 7110 med delvis frändragen lucka.....	24
Figur 11.	Ericsson R320s.....	25
Figur 12.	Ericsson R380s med öppnad lucka.....	25
Figur 13.	Nokia WAP SDK.....	26
Figur 14.	Nokia Toolkit 1.2.....	27
Figur 15.	Ericsson WAP IDE.....	27
Figur 16.	Phone.com:s UP.Simulator.....	28
Figur 17.	Möjliga kopplingar mellan WAP-system och existerande tjänst.....	32
Figur 18.	Systemförslag WAP-tjänst med WAP 1.1 över GSM-CSD.....	35
Figur 19.	Tidsdiagram för användning av WAP-röstbrevlåda.....	36
Figur 20.	Skiss av struktur för en WAP 1.2-baserad implementering.....	43
Figur 21.	Tidsdiagram för meddelandeuppspelning.....	46
Figur 22.	Nokia Toolkit i 6150-läge.....	53
Figur 23.	Nokia Toolkit i 6110-läge.....	53
Figur 24.	Nokia SDK (samma utseende som Nokia 7110).....	54
Figur 25.	Ericsson IDE i 320s-läge.....	54
Figur 26.	UP.simulator innan och efter att namnet skrivits in.....	55
Figur 27.	Sidstruktur i prototypen.....	68
Figur 28.	Klassdiagram för prototypen.....	69
Figur 29.	”Använder”-strukturen för en Servlet i prototypen.....	70

Tabeller

Tabell 1.	Prestanda hos några olika bärare.....	19
Tabell 2.	Ordförklaringar.....	57
Tabell 3.	Förkortningar.....	60

Kodexempel

Kodexempel 1.	Hello World i WML	15
Kodexempel 2.	Hello World i HTML	16
Kodexempel 3.	Kod i WMLScript som beräknar amorteringar på ett lån	17
Kodexempel 4.	WMLScriptkod som går till en ny unik URL givet en bas-URL....	41
Kodexempel 5.	URL som går till oldURL modifierad så att den är unik.	41
Kodexempel 6.	URL skapad av unikifieringsprogrammet	41
Kodexempel 7.	Exempelsida för klientjämförelser	52

1 Inledning

Den här rapporten består av två huvuddelar. Kapitlen 2 till 4 beskriver bakgrunds-information medan kapitlen 5 och 6 beskriver vad examensarbetet kommit fram till.

Kapitel 2 beskriver det problem som examensarbetet behandlar och förklarar kortfattat vad en röstbrevlåda är och hur den fungerar. Kapitel 3 redogör för den metod som nyttjats i arbetet. Kapitel 4 beskriver WAP på ett sätt som ger de förkunskaper som krävs för god förståelse av resten av rapporten.

Kapitel 5 innehåller det som examensarbetet lett fram till uppdelat i några olika delar beroende på ur vilket tidsperspektiv problemet studeras. Kapitel 6 beskriver några av de nya problem som upptäcktes men lämnades i stort sett obehandlade under arbetet.

För att tillgodogöra sig rapporten på ett bra sätt är det lämpligt att ha vissa grundläggande kunskaper om nätverk och då speciellt Internet. Viss webbkunskap som att känna till HTML och JavaScript kommer också väl till pass. Rapporten använder så långt som möjligt svenska tekniska termer, men för en del ord finns det inte någon lämplig svensk term och då används den engelska istället. Exempel på sådana ord är cache och gateway.

2 Problemställning

Det här examensarbetet föreslogs av Epact Technology AB och handlar kortfattat om att undersöka hur WAP, Wireless Application Protocol, kan användas i röstbrevlådetjänster. WAP-tekniken är helt ny och även om den vid en första anblick gör det mesta möjligt visste Epact inte riktigt vad som var möjligt att göra med den nya tekniken i praktiken. Epact, som underhåller och vidareutvecklar MobilSvar åt Telia, ville därför undersöka möjligheterna att integrera WAP med en röstbrevlådetjänst som till exempel MobilSvar.

Det här examensarbetet hade inte som mål att skapa en tjänst som skulle kunna användas, utan målet var att se vad som var möjligt att göra. Det sattes tidigt upp en lista på intressanta punkter i samarbete med Epact och den listan med kortfattade slutsatser finns i bilaga B.

2.1 Epact Technology AB

Epact Technology AB är ett konsultföretag som verkar från Mjärdevi Science Park i Linköping och från Kista i Stockholm. Företaget grundades 1992 och har sina rötter i Linköpings Tekniska Högskola. Huvudområden i Epacts verksamhet är mjukvaruteknik, telekom, signal- och bildbehandling, sensorteknik och elektronik. Telekom-avdelningen jobbar bland annat med vidareutveckling av Telias röstbrevlåda, MobilSvar, åt Telia Mobitel.

2.2 Motivering till examensarbetet

2.2.1 Röstbrevlådeöversikt

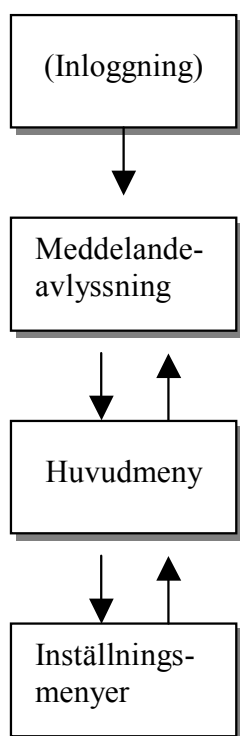
För att förstå varför Epact ville ha det här examensarbetet utfört är det bra att först känna till lite om vad en röstbrevlåda är och vad de har för egenskaper.

En ”röstbrevlåda” eller ibland bara ”röslåda” är ett system där en person via telefon eller eventuellt andra media kan lämna ett meddelande i talform. Den klassiska varianten påminner ganska mycket om en vanlig telefonsvarare och har ungefär samma funktioner. Dagens användare kräver mycket mer och de tjänster som erbjuds av teleoperatörerna i Sverige är alla mer eller mindre finessrika. Funktioner som förekommer är bland annat att röstbrevlådan själv kan kontakta sin ägare om det

kommer meddelanden. Det kan vara via personsökare, SMS-meddelanden eller till och med att röstbrevlådan ringer upp abonnenten.

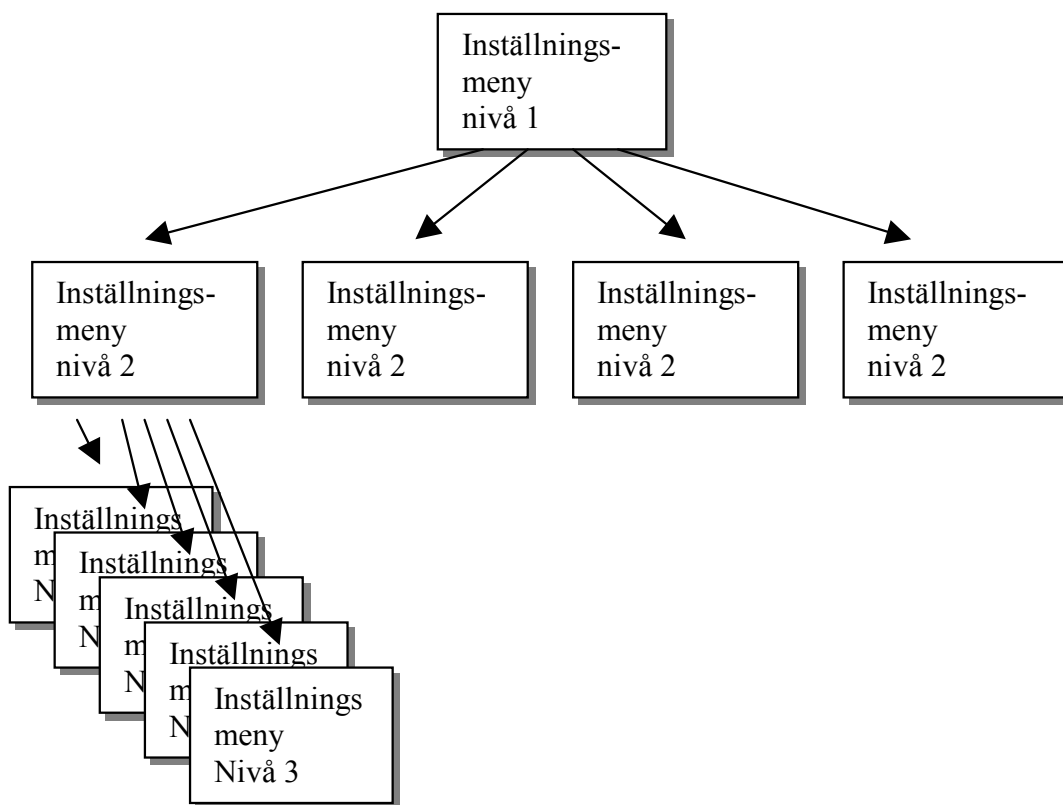
En annan sak som de flesta tar som självklar är att ägaren till röstbrevlådan får veta när ett meddelande talades in och att röstbrevlådan håller reda på vilka meddelanden som redan spelats upp. Röstbrevlådan flyter också i vissa system ihop med andra brevlådor såsom brevlådor för e-post och faxmeddelanden. Det gäller till exempel i Telias UM-DOF-abonnemang. Det brukar dessutom vara möjligt att lyssna av sin röstbrevlåda från andra ställen än den egna telefonen.

Idag styrs de flesta röstbrevlådor från knappsatsen på telefonen. En knapptryckning sänder en DTMF-signal som röstbrevlådan känner igen och reagerar på. Menyerna och menyalternativ läses upp av en inspelad röst från vilken användaren också får instruktioner och information.



Figur 1. Navigering i en traditionell röstbrevlåda

Den struktur som syns i figuren ovan är den vanligaste för röstbrevlådor och verkar enkel och bra tills lådan ”Inställningsmenyer” studeras närmare. Det är det som syns i figuren nedan. Den figuren visar hur menyer kan bli både många och djupt nästlade. Eftersom människans korttidsminne endast rymmer 7 ± 2 saker (Miller, 1956) så bör inte menyerna bli alltför långa vilket istället leder till det stora antalet. Dessutom kan systemet inte läsa upp hur mycket text som helst eftersom väntetiderna då skulle bli för långa. Nackdelen med att menyerna måste hållas korta blir alltså att de istället växer på djupet och ju fler funktioner en röstbrevlåda får, desto djupare blir menystrukturen och desto svårare blir det för användaren att få överblick och hitta det som söks.



Figur 2. Menystruktur i en traditionell röstbrevlåda

2.2.2 Målsättning för en WAP-tjänst

Ur användbarhetssynvinkel finns det stora möjligheter med att få tillgång till en liten skärm att visa text och bilder på, men naturligtvis vill varken användarna eller de som äger systemen samtidigt förlora delar av den funktionalitet som finns i systemen idag.

Det användaren vill kunna göra är att lyssna av meddelanden, se när de kom och spela in hälsningsmeddelanden. Användaren vill kunna komma åt sina meddelanden från andra ställen än sin egen telefon och dessutom bli informerad om att det kommit nya meddelanden.

Det nya medium som blir tillgängligt för de som utvecklar mobila tjänster i och med WAP är en skärm där användaren kan läsa och skriva text. Skärmen har funnits där tidigare men först nu har tjänstetillverkare kunnat påverka vad den visar och även om den oftast är liten är det ändå en skärm. Det betyder att WAP tycks kunna ge möjligheter att förbättra användargränssnitt som idag har vissa begränsningar. Samtidigt som tjänsterna har utvecklats har också antalet inställningar ökat vilket gör att användaren får navigera genom ganska omfattande menysystem vägled av muntliga instruktioner. Det är lätt att tappa bort sig på vägen eller att trycka fel siffra. Användaren kan också få sitta länge och vänta tills det önskade menyalternativet läses

upp. Det är inte heller smidigt att gå fram och tillbaka i menyer vilket en användare kan behöva göra om han eller hon inte är säker på var ett menyval finns.

I en framtid är det även tänkbart att röstbrevlådor kan göra omvandlingar från tal till text eller från text till tal. Det vill säga att tjänsten översätter det meddelande någon talar in till skriven text eller det omvända, att skriven text läses upp av tjänsten. Det skulle bland annat göra tjänsten tillgänglig för handikappade, både de med nedsatt syn och de med nedsatt hörsel, men även kunna fungera som ett komplement för vanliga användare. Det finns situationer när det inte är möjligt för ägaren till en röstbrevlåda att på ett smidigt sätt lyssna av ett meddelande, men där det skulle vara möjligt att läsa det, till exempel i en bullrig miljö eller om användaren har nedsatt hörsel. Även det motsatta är intressant, att det inte är möjligt att läsa ett meddelande för att den som tar emot meddelandet saknar skärm eller behöver hålla ögonen på någon annat, i vilket fall ett uppläst meddelande skulle kunna vara användbart.

På det hela taget är dagens tjänster användbara men det finns områden där de kan bli bättre. Att utnyttja de möjligheter som WAP tycks kunna ge vore ett steg i den ständiga förbättringsprocessen.

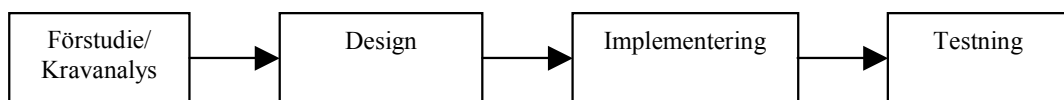
3 Metod

Det här kapitlet beskriver den metod som använts under examensarbetet, varför den valdes samt hur den tillämpades under arbetets gång.

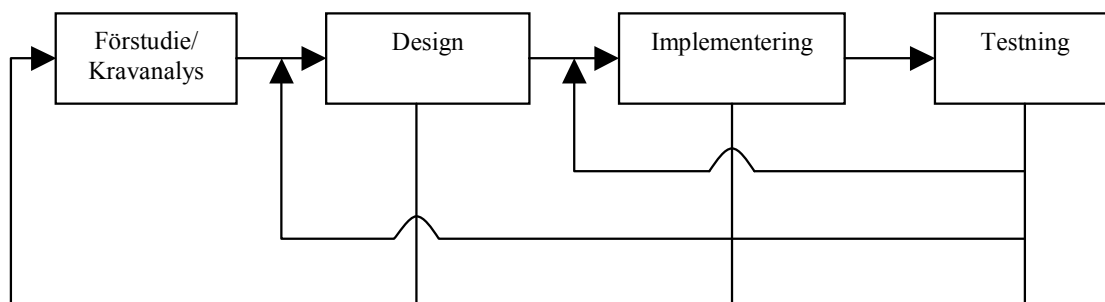
3.1 Metodval

Ett examensarbete ska vara vetenskapligt utfört. Det betyder att examensarbetaren ska vara strukturerad i sin arbetsmetod och dokumentera de metoder han eller hon använder. Examensarbetaren ska redan från början sätta upp mål och förutsättningar och sedan beskriva hur han eller hon under de förutsättningarna kan/ska nå målen. För min del innebar det bland annat att gå igenom punkterna i en lista med intressanta oklara områden och sedan ställa upp tester för att försöka verifiera var och en av dessa områden.

Vilken vetenskaplig metod som väljs i ett projekt är i mycket beroende av förutsättningarna och det är inte möjligt att finna en metod som passar för alla projekt. I en så oklar miljö som det här var, är det inte lämpligt att använda den vanligaste programutvecklingsmodellen som i stort sett ser ut så att de som utför ett projekt har ett antal steg som de tar ett i taget tills projektet är klart (vattenfallsmodellen), se figur nedan. Alternativt kan projektet, om det visar sig lämpligt, backa ett antal steg och sedan fortsätta därifrån, vilket kallas den iterativa vattenfallsmodellen se figur nedan (Sommerville, 1989).

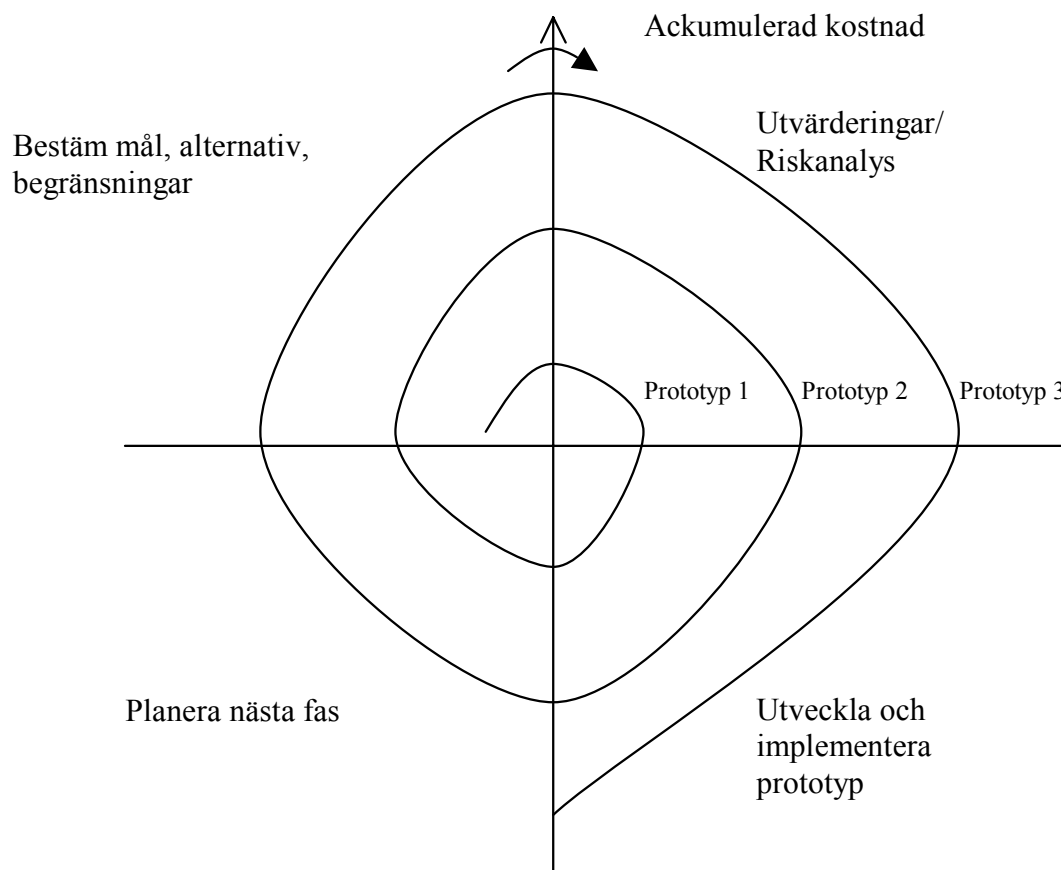


Figur 3. Vattenfallsmodellen



Figur 4. Den iterativa vattenfallsmodellen

I stället för de vanliga vattenfallsmodellerna var det i en så här osäker miljö lämpligt att använda inkrementell prototypning av vilket Boehms spiralmodell är en variant, se bild nedan. Inkrementell prototypning är synnerligen väl lämpat, jämfört med andra metoder, när de som ska utföra ett projekt har väldigt liten kunskap om området innan de börjar arbeta med det.



Figur 5. Boehms spiralmodell

I Boehms spiralmodell (Boehm, 1988) inleds arbetet med att projektgruppen försöker hitta de största riskerna i projektet. Med risker menas de områden som det råder stor osäkerhet om. När dessa områden identifierats görs en testimplementering, prototyp, av

de berörda bitarna för att på det sättet få erfarenhet av området. Efter att prototypen implementerats, testats och utvärderats görs en ny planering och nya riskområden identifieras varefter hela proceduren upprepas på nytt. Till sist kastas alla prototyper och projektarbetarna gör en fullständig nystart med de kunskaper de skapat sig. Det är det som är sista cykeln i spiralen. I figuren ovan finns tre cykler men modellen som sådan säger inget om hur många som är lämpliga. Allt arbete på prototyper behöver inte heller ske sekventiellt utan inom projektet kan en ny cykel påbörjas innan den föregående är avslutad och på det sättet fås en viss parallellism i utvecklandet.

Den stora skillnaden mot vattenfallsmodellen och dess iterativa variant är att projektgruppen vid inkrementell prototypning inte direkt försöker angripa hela problemet utan börjar med att ge sig på väl utvalda delar istället. På det sättet kastar sig inte de som jobbar med projektet in i något stort som de inte vet tillräckligt mycket om. Det finns också en metod som kallas inkrementell utveckling (Blum, 1992) där de som använder metoden istället börjar arbeta på de bitar de tror sig vara säkra på och som kan vara användbara för resten av programmet. Det är dock inget som varit aktuellt för det här examensarbetet.

Jämfört med vanliga vattenfallsmodellen där faserna förstudie, design, implementering och testning bara går igenom en gång har inkrementell prototypning en stor fördel. Det är att projektgruppen inte, på grund av okunskap, gör stora fel i en tidig fas som inte upptäcks förrän i en av de senare faserna. Fördelen med att upptäcka potentiella fel tidigt kan inte nog betonas. Grovt räknat så uppskattas kostnaden för att rätta ett fel att öka med en faktor tio för varje fas felet går igenom utan att det upptäcks och åtgärdas, till exempel kan ett fel som upptäcks vid sluttestningen och som kräver omdesign kosta uppemot 100 gånger så mycket (tid, pengar m.m.) i ett stort projekt som om det upptäckts och rättats redan i designfasen.

Andra fördelar, vilka tas upp i van Vliet et al (1993), är att inkrementell prototypning resulterar i en betydligt snabbare utveckling och betydligt mindre kod. I ett experiment som nämns tog ett projekt 40% mindre tid och den resulterande produkten hade 45% mindre kod jämfört med vad resultatet blev när projektet utfördes med en traditionell vattenfallsmodell. Tyvärr är det inte bara fördelar med inkrementell prototypning. Samma studie som visade på tidsvinster visade också att den slutliga produkten med den traditionella modellen blev stabilare och lättare att underhålla. Resultatet blev alltså en sämre kvalitet på slutprodukten när inkrementell prototypning användes. Detta är något som måste hållas i åtanke när denna metod väljs så att det åtminstone finns medvetenhet om riskerna.

I det här examensarbetet var inte målet att producera en färdig produkt utan spiralen kunde avslutas innan den egentligen har nått ändå fram enligt den fullständiga modellen. Min förhoppning var i alla fall att jag skulle kunna sätta ihop en form av slutprototyp som sammanfattade det som gjorts. Den tillsammans med den erfarenhet som byggts upp skulle då förhoppningsvis kunna användas om någon vill arbeta vidare med området.

3.2 Genomförande

Med Boehms spiralmodell som bakgrund inleddes arbetet med att identifiera risk-områden. Från början när det var mycket som var okänt togs några stora grepp på områden som var viktiga för att komma vidare. Senare när det var mer detaljer som var oklara blev det mindre "projekt" för att klarlägga dessa. Ju längre arbetet pågick och ju mindre detaljer som var oklara desto mer informell blev också arbetsgången.

Parallellt med det målinriktade arbetet på det aktuella problemet pågick hela tiden studier av specifikationer och nyheter inom WAP-området. Under första tredjedelen av tiden fick också mycket arbete göras rent teoretiskt innan Epect lyckades få tag på en WAP-telefon, Nokia 7110, som kunde användas för tester. Det är den telefonen som, trots sina brister, fungerat som referenstelefon under examensarbetet.

Det första steget (motsvarande första varvet i Boehms spiral) blev att se vad som krävdes för att få till en fungerande miljö för användning av WAP. I första hand var det intressant att se vad som krävdes för hårdvara och mjukvara på serversidan vilket blev arbetets första steg. Det visade sig ganska snart att mycket arbete som hade tänkts göras i olika varv i spiralen fick göras parallellt nästan helt och hållet. Det blev mer sicksackande fram och tillbaka mellan olika varv istället för att gå varven runt och runt. Detta berodde dels på att information om speciella ämnen var svår att få fram och det därför blev luckor i arbetet som behövde fyllas och dels på att de olika varven delvis var beroende av varandra.

Det blev till exempel svårt att testa servern utan att ha några genererade WML-sidor att testa med så ganska logiskt kom fasen "att generera WML-sidor" in tidigt.

Det som gjordes för att ändå få en bra struktur på arbetet var att ha återkommande möten med handledaren varje vecka. Dessa möten behandlade i huvudsak det arbete som genomförts senaste veckan samt planeringen av arbetet den följande veckan. På så sätt gick hela tiden arbetet vidare med såväl kortsiktiga som långsiktiga mål (de som beskrivs i den här rapporten).

I samband med studierna och försöken växte samtidigt en prototyp av en WAP-tjänst fram, även om centrala delar saknades för att göra den komplett. Prototypen hade sitt ursprung i metoden som valdes som är prototypcentrerad, och samtidigt ville Epect gärna ha något som gick att visa upp. Den ligger till grund för en hel del av de slutsatser som dras i den här rapporten. Hur den mer i detalj var uppbyggd kan ses i bilaga A, där prototypen förklaras.

För koden användes UML (Eriksson & Penker, 1998) till en del för att modellera prototyperna även om det användes mer för att analysera existerande kod än för att bygga nya modeller.

3.3 Källor

En stor svårighet med såpass ny teknik som WAP är att det finns väldigt få tryckta källor och de få som finns är inaktuella innan de lämnat tryckeriet. Det ställe den intresserade kan hitta aktuell dokumentation på är i första hand Internet vilket reflekteras ganska mycket i de källor som använts och listats i referenslistan. Ett stort problem med information på Internet är att det är svårt att bedöma trovärdigheten och jag har därför begränsat mig till källor som är av officiell natur, såsom standardiseringsorgan och specifikationer publicerade av tillverkare.

4 WAP-översikt

Det här kapitlet beskriver vad WAP är och hur det fungerar med tyngdpunkt på de delar av WAP som är relevanta för resten av rapporten.

4.1 Historik

WAP, Wireless Application Protocol, utvecklades av WAP Forum som är en sammanslutning av just nu ett hundratal företag med intressen inom mobil telefoni och data. I WAP White Paper (1999) beskrivs hur WAP Forum bildades i juni 1997 av Ericsson, Nokia, Motorola och Phone.com (som då hette Unwired Planet). 20 april, 1998 släppte de specifikationen till WAP 1.0. Den fick inte något större genomslag vad det gäller produkter och uppmärksamhet i media men arbetet bakom kulisserna tog fart på riktigt. När WAP 1.1 sedan släpptes i juni 1999 fick WAP ordentlig med publicitet i media. Delar av WAP 1.1 visade sig vara inkompatibla med WAP 1.0 men eftersom WAP ännu inte hade spridits till konsumenter så märktes inte det för gemene man.

Under hösten 1999 pågick arbete på WAP 1.2 och WAP Forum fastlade den standarden i januari 2000. Alla specifikationer som WAP Forum fastställer finns att läsa på Internet och kan hittas på deras hemsida som har adress <http://www.wapforum.org/>.

4.2 Syfte med WAP

Tanken bakom WAP är att göra datanätverk allmänt tillgängliga från mobila terminaler, till exempel mobiltelefoner som har begränsat minne, begränsat processor-kraft, liten skärm och begränsad bandbredd. På grund av de begränsningarna var det inte lämpligt att direkt koppla upp mot Internet med TCP/IP och använda HTML-sidor på det sätt som Internet-användare gör med datorer. Sammanfattat säger WAP Forum på <http://www.wapforum.org/what/index.htm> följande:

~ *What is the purpose of WAP?*

~ *To enable easy fast delivery of relevant information and services to mobile users.*

4.3 Överföringsoptimering

I det mobila nätet finns det mycket mindre bandbredd än vad som är fallet i de flesta vanliga datornät. Det var därför nödvändigt att använda speciella metoder för att göra det möjligt för WAP att klara sig med liten bandbredd. De två huvudmetoderna är beskrivna nedan

4.3.1 Bytekodning

Att skicka data som ren text vilket görs idag på webben är egentligen ett ganska stort slöseri med nätverksresurser. Detta eftersom mycket information i text är redundant. Enligt Shannon (1951) innehåller varje bokstav i engelsk text i genomsnitt endast 1-2 bitar information och eftersom varje bokstav typiskt skickas som 8 bitar slösas 70-80% av den tillgängliga nätverkskapaciteten bort i onödiga överföringar. Nu är det inte praktiskt möjligt att göra så bra kodningar på en normal mobiltelefon men det är åtminstone möjligt att minska mängden data som måste skickas över. Eftersom detta är ett av kraven för att göra WAP användbart har WAP-standarden infört något den kallar bytekodning (WBXML 1.1, 1999). Det innebär att vissa textbitar som förekommer ofta ersätts med speciella koder, normalt en kod på 8 bitar. Bytekodningen ger även möjligheten att bara skicka en textbit en enda gång över nätet även om den förekommer flera gånger i den ursprungliga texten. Allt som allt visar tester att den bytekodade texten ofta bara blir ca 50-60% så stor som den som inte blivit bytekodad.

Exempel:

Texten `href="http://` som normalt representeras av 8 bitar per tecken ersätts med den binära koden 01001011. Det blir en minskning från 104 bitar till 8, vilket är hela 93%.

4.3.2 Kortlek

Istället för att föra över en sida i taget för WAP över något som kallas kortlekar och som består av flera sidor på en gång, se avsnitt 4.4. På det sättet behöver inte WAP-klienten initiera nya överföringar för varje gång den lilla skärmen ska fyllas med ny sida. Om den som tillverkar sidor för WAP anpassar sig efter små skärmar blir varje sida bara några hundra bytes, så när en sida i taget förs över blir det procentuellt mycket extra data som tillkommer för administration av överföringen. Genom att föra över ett antal sidor på en gång blir det procentuellt mycket mindre extradata som förs över per sida. Dessutom leder det till att WAP-telefonen inte behöver göra uppkopplingar mot nätverket så länge användaren håller sig till sidor inom samma kortlek, till skillnad mot hur det är på webben där varje ny sida innebär en ny uppkoppling. Det är särskilt viktigt med WAP eftersom fördröjningen än så länge kan vara stor vid en uppkoppling. Utan den fördröjningen blir sidbyten istället mycket snabba. Därför kan WAP-tjänster i vissa sammanhang upplevas som betydligt snabbare än motsvarande webbtjänster.

4.4 WML

WML, "Wireless Markup Language" (WML 1.1, 1999), är ett sidbeskrivningsspråk som är baserat på XML, "eXtended Markup Language". XML är en fastslagen standard av W3C, World Wide Web Consortium (XML, 1999). Eftersom WML använder samma namn på identifierare som HTML är WML lätt att förstå för den som känner till HTML. Den största och ganska avgörande skillnaden är att WML beskriver kortlekar se avsnitt 4.3.2, istället för enskilda sidor.

En annan skillnad är mängden taggar, identifierare som beskriver texten. WML är designat för att kunna fungera på små terminaler med skärmar i storleksordningen 4 rader á 20 tecken och där både minne och processorkraft är kraftigt begränsade. Vad som finns är enkel uppmärkning av text som säger att den ska visas understruken, med fetstil eller kursiv. Det finns också enkla formulärmöjligheter så att användaren kan mata in text, lösenord eller välja alternativ från listor. Det är också möjligt att ha länkar till andra kort i samma kortlek eller kort i andra kortlekar.

En sak som inte är möjlig att göra direkt i HTML, utan som kräver att skriptspråk som JavaScript används, men som kan göras direkt i WML är att binda små program till vissa fördefinierade aktiviteter. Dessa program kan då aktiveras av speciella knappar på en mobiltelefon. Till exempel finns aktiviteten "prev" som normalt betyder att användaren går tillbaka till en föregående sida. Andra aktiviteter som finns fördefinierade är "accept", "options" och "help". Här finns dock ett litet problem. Vad den som gör sidan vill uppnå är att få en funktion lättillgänglig på en knapp men eftersom olika mobiltelefoner lägger olika aktiviteter på knapparna måste den som tillverkar sidorna specialanpassa sidan för varje mobiltelefon för att få funktionen på en lättillgänglig knapp.

Nedan finns exempel på hur en enkel sida som bara visar "Hello **World!**" kan beskrivas i WML respektive HTML.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="card1" title="WML Page">
<p>
Hello <b>World</b>!
</p>
</card>

</wml>
```

Kodexempel 1. Hello World i WML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">

<html>
<head><title>HTML page</title></head>
<body>
Hello <b>World</b>!
</body>
</html>
```

Kodexempel 2. Hello World i HTML

Vid en jämförelse av kodbitarna märks inga stora skillnader. WML-koden måste börja med en xml-deklarering eftersom det är ett XML-språk, men annars inleds båda av en "DOCTYPE" som talar om vad det är för sorts dokument. WML-dokumentet har sedan kortleksinnehållet innanför <wml> medan HTML har <html>. I WML måste ett kort skapas vilket görs med <card> medan det i HTML-koden finns en <body> som innehåller sidans innehåll. Slutligen måste texten i WML stå inom en styckemarkering <p> medan det kan lämnas som underförstått i HTML. Titeln på sidan har i WML också flyttat in till att vara ett attribut till själva kortet istället för att ligga i huvudet ("head").

4.5 WMLScript

För att kunna göra lite mer avancerade saker på klientsidan i WAP finns det ett skriptspråk, WMLScript (WMLScript 1.1, 1999). WMLScript är baserat på ECMAScript (ECMA-262, 1998) som även är känt som JavaScript. Den största skillnaden är att många avancerade konstruktioner i ECMAScript plockats bort just för att anpassa språket till den mer begränsade miljö en mobiltelefon är vilket betonas i kapitel 1 i *WMLScript 1.1* (1999). En annan skillnad mot hur ECMAScript används på webben är att all WMLScript-kod måste finnas i en annan fil än WML-koden. Den WMLScript-koden kan sedan anropas med hjälp av URL:er i WML-koden.

Med WMLScript är det möjligt att göra såväl ganska enkla saker som att multiplicera tal till mer avancerade saker som att göra om helt vanliga URL:er till unika sådana precis när en URL används. Just det senare gjordes för att komma ifrån en del problem under examensarbetet.

Följande är ett exempel på kod i WMLScript som beräknar amorteringar på ett lån.

```

/*
 * Calculate a mortgage's payment
 *
 *@param varname the variable name to store the result
 *@param principal the principal
 *@param interest the interest rate
 *@param num_payments the number of payments
 *@return the payment
 */
extern function payment(varname, principal, interest, num_payments)
{
    /*
     * Interest formulae:
     *
     * If (i != 0), then:
     *          pmt = principal * [i * (1+i)^n / ((1+i)^n - 1)]
     *
     * If (i == 0), then:
     *          pmt = principal / n
     */
    var mi = interest/1200; // monthly interest from annual
                        // percentage
    var payment = 0;
    if (mi != 0) {
        var tmp = Float.pow((1 + mi), num_payments);
        payment = principal * (mi * tmp / (tmp - 1));
    } else {
        if (num_payments != 0)
            payment = principal / num_payments;
    }
    var s;
    if (payment != 0)
        s = String.format("$%6.2f", payment);
    else
        s = "Missing data";

    /*
     * Send the result to the browser
     */
    WMLBrowser.setVar(varname, s);

    /*
     * Make sure the browser updates its current card
     */
    WMLBrowser.refresh();
};

```

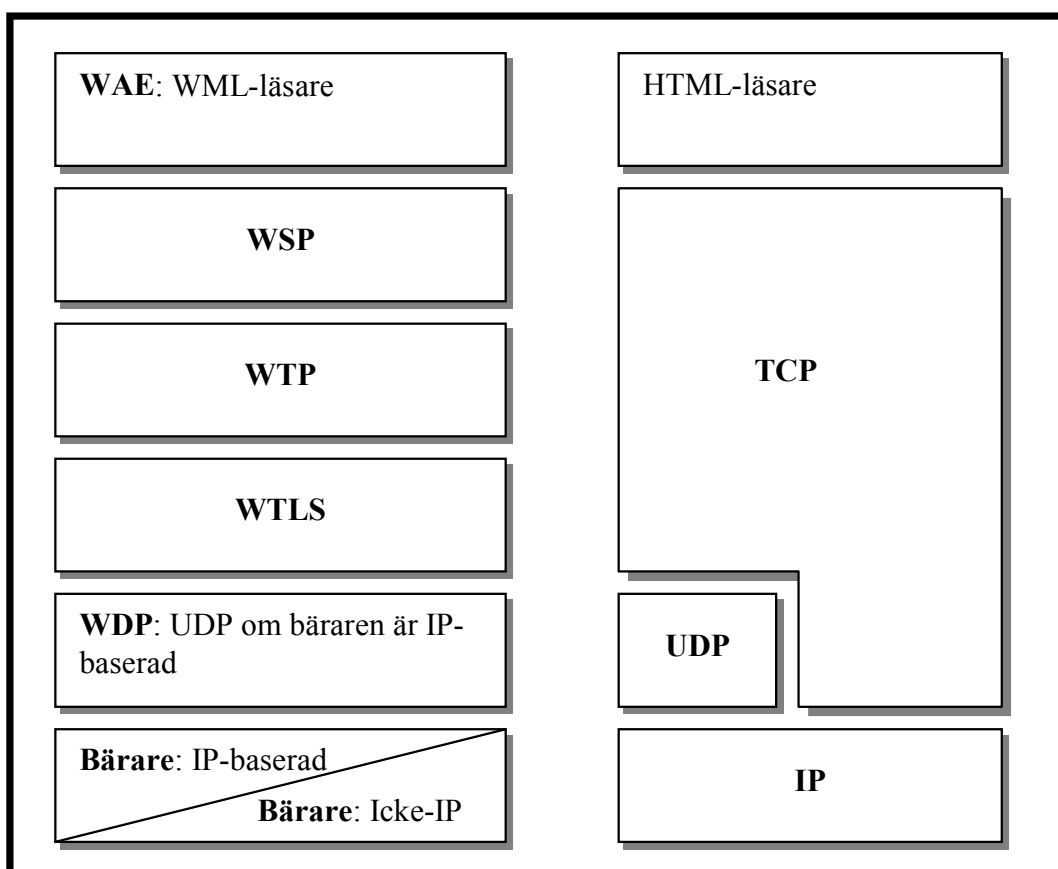
Kodexempel 3. Kod i WMLScript som beräknar amorteringar på ett lån

I koden refereras ett funktionsbibliotek `WMLBrowser` som innehåller funktioner för att manipulera utseendet på skärmen. I den här koden används `WMLBrowser` först för att sätta en lokal variabel i WML-läsaren till det beräknade värdet med hjälp av metoden `setVar` och sedan uppmanas WML-läsaren att rita om skärmen med det nya värdet genom att använda metoden `refresh`.

4.6 Protokollstacken

En av de centrala delarna av WAP är protokollstacken som översiktligt finns beskriven i WAP Architecture (1998). En ”stack” i det här fallet är en uppsättning protokoll som tillsammans beskriver hur kommunikationen sker på olika abstraktionsnivåer. Dessa protokoll jobbar tillsammans i olika lager där abstraktionen är minst längst ner och störst högst upp. Det skulle till exempel vara möjligt att ha en protokollstack där det längst ner finns ett protokoll som beskriver vilka spänningar det ska vara på olika ledningar vid olika tillfällen och högst upp något avancerat system som låter användaren styra ett kärnkraftverk.

I nedanstående figur jämförs WAP:s protokollstack med den TCP/IP-stack som används på Internet.



Figur 6. WAP-stacken jämförd med TCP/IP-stacken

4.6.1 Bärare

I WAP-stacken finns längst ner det som brukar kallas bärare. Bäraren är den redan existerande teknik som trafiken ska gå över. Detta kan vara SMS-meddelanden (korta meddelanden mellan GSM-telefoner), GSM-CSD (datauppkoppling på en GSM-telefon), GSM-USSD (trafik över GSM:s signaleringskanal), IS-136 (amerikansk

mobiltelefonstandard) eller något helt annat. I Europa är det egentligen bara de två förstnämnda, SMS och GSM-CSD, som är aktuella i dagsläget även om GSM-USSD också skulle kunna användas. Framöver kommer dock en del nya tekniker, såsom GPRS och UMTS tillkomma som bärare och därmed ge WAP både betydligt bättre prestanda och snabbare uppkopplingar.

I huvudsak delas bärarna upp i två klasser, de som kan köra IP-trafik och de som inte kan köra IP-trafik. Till de bärare där IP fungerar (och ska användas) hör GSM-CSD och framöver såväl GPRS som UMTS. Till kategorin som inte använder IP hör bland annat SMS och USSD.

Det är bäraren som till stor del avgör vilken prestanda en förbindelse får. Det finns två aspekter av prestanda. Den ena aspekten är fördröjning, vilket är den tid det tar för data att komma fram, och den andra är bandbredden. Exempelvis har en lastbil som kör 1 timme mellan två platser fullastad med databand (till exempel med 3600 GB data) väldigt hög bandbredd (1 GB/s) men stor fördröjning (1 timme).

Bärare	Fördröjning	Bandbredd
GSM-CSD	>30 s om uppkoppling saknas <1 s om uppkoppling finns	9600 bitar per sekund
SMS	>1minut	160 bytes per meddelande vilket beroende på hur fort meddelande sänds kan bli upp till 9600 bitar per sekund
GPRS (2000-2001)	<1 sekund	9600-13200 bitar per sekund och kanal där det är möjligt att ha mellan 1 och 8 kanaler
UMTS (ca 2003)	<1 sekund	Upp till 2 Mbit per sekund

Tabell 1. Prestanda hos några olika bärare

I ovanstående tabell finns en jämförelse mellan olika bärarens prestanda. Som synes kan det variera inom ett mycket stort intervall även om GPRS och UMTS fortfarande inte finns på marknaden utan är framtidstekniker. Siffrorna om fördröjning är inte entydiga utan reflekterar den typiska bärarrelaterade fördröjningen. Det är till exempel fullt möjligt för ett SMS-meddelande att komma fram snabbare än på 1 minut, även om det inte händer ofta.

4.6.2 WDP

Ovanpå bärarlagret finns ett transportlager, WDP, "Wireless Datagram Protocol" (WDP 1.1, 1999), och det är det lägsta lagret som WAP specificerar i detalj. Syftet med transportlagret är att ge alla möjliga bärare ett gemensamt gränssnitt uppåt så att de övre lagren i stort sett kan bortse från skillnader mellan olika bärare. Speciellt säger standarden att om det är möjligt ska WDP vara exakt samma sak som Internetprotokollet UDP. Det gäller speciellt om bäraren är baserad på, eller klarar av, IP-trafik.

Det finns dock möjligheter att till exempel fråga transportlagret om egenskaper hos bäraren även om transportlagret utåt ska se likadant ut oavsett bärare. Det är för att lager högre upp ska kunna göra anpassningar efter bäraren om det är lämpligt. Det kan till exempel vara ett annat lager som vill ha olika maximala tider för ett meddelande beroende på hur snabb bäraren är. Om bäraren är SMS som normalt kan ha väntetider på någon eller ett par minuter är det inte meningsfullt att göra nya sändningsförsök efter ett par sekunder.

4.6.3 WTLS

Fortsätter man uppåt i protokollstacken så finns ovanpå transportlagret, WDP, ett säkerhetslager, WTLS, ”Wireless Transport Layer Security” (WTLS 1.1, 1999). Det här lagret används inte alltid men om det används så finns i stort sett samma möjligheter som i den existerande Internetstandarden TSL (RFC2246, 1999) som i sin tur är baserad på SSL 3.0. SSL används på Internet bland annat på webben vid säkra förbindelser till t.ex. webb-banker. För den som använder WAP innebär WTLS att det finns skydd som försvårar för obehöriga att avlyssna eller ändra på data. Eftersom data är krypterat krävs det omfattande arbete för att komma åt informationen. Det finns också stöd som ska förenkla säker identifiering av motparten (gäller både klienten och servern).

Idag finns det dock en hel del frågetecken kring hur WTLS ska användas. Ett av problemen är att det endast är trafik mellan själva telefonen och operatören som skyddas. Om trafiken går vidare från operatören ut på Internet så är den delen helt oskyddad. För det andra leder det begränsade minnet, den begränsade processorkapaciteten och vissa exportlagrar till att de krypteringsalgoritmer och nycklar som kommer att användas i ett första skede antagligen blir ganska svaga, det vill säga lättknäckta.

4.6.4 WTP

Vidare uppåt i protokollstacken finns WTP, ”Wireless Transaction Protocol”, (WTP 1.1, 1999). Detta är den del av protokollet som ansvarar för att all data kommer fram till mottagaren och gör omsändningar om det behövs. I sin funktionalitet påminner det ganska mycket om TCP i ett TCP/IP-nätverk.

4.6.5 WSP

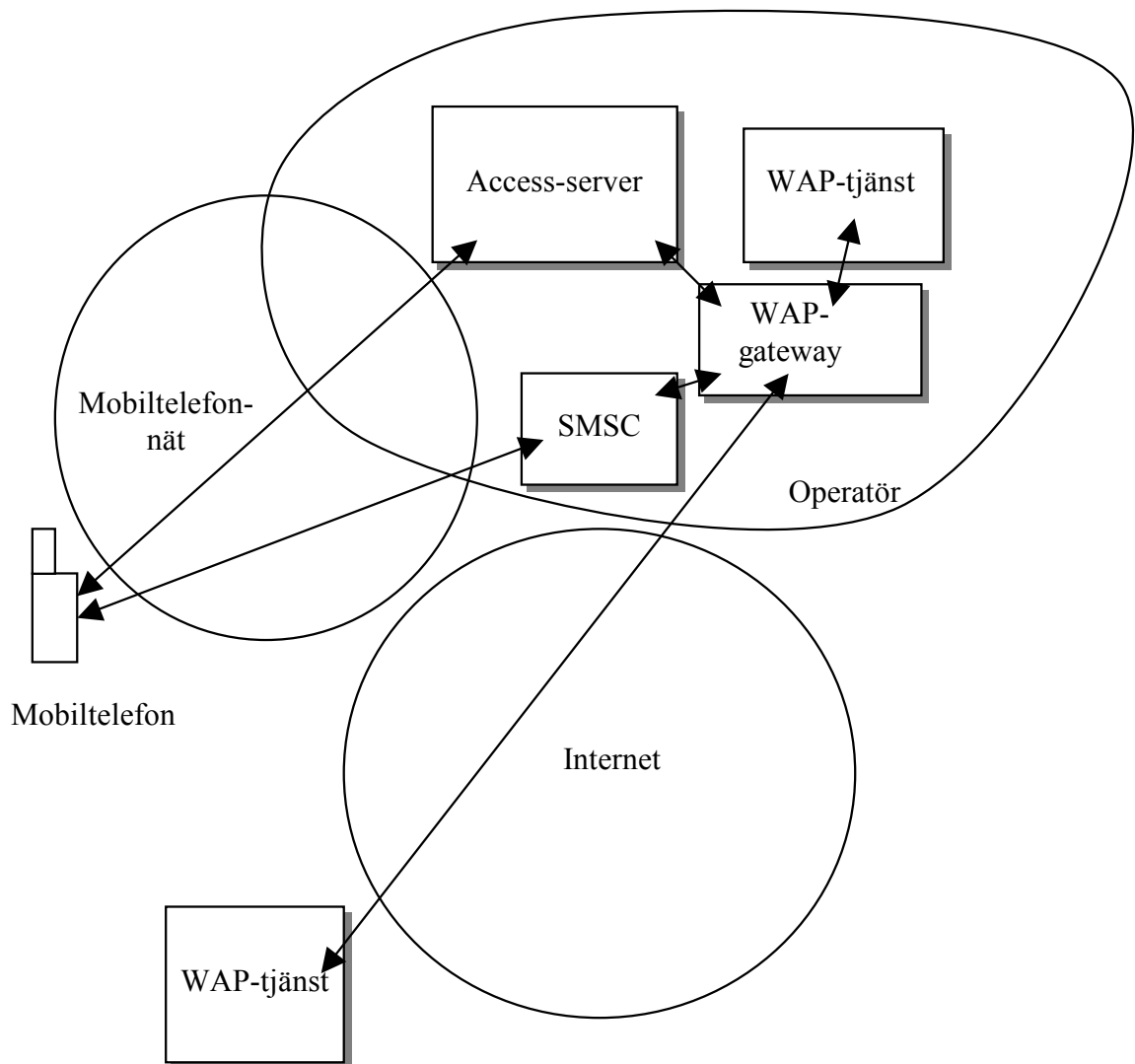
Det översta lagret som behandlar rena kommunikationsaspekter är WSP, ”Wireless Session Protocol” (WSP 1.1, 1999), som är det lager som innehåller uppkopplingsfunktionalitet. I WSP finns också HTTP/1.1 inbyggd i en variant som kan användas i WAP. WAP använder nämligen HTTP för att hämta WML-sidor och annan data. Mellan klienten och gatewayen används WAP:s egna specialanpassade variant av HTTP som bland annat är bytekodad. Gatewayen översätter sedan till standardiserad HTTP/1.1 för användning på Internet.

4.6.6 WAE

Protokollstackens översta del är WAE, ”Wireless Application Environment” (WAE 1.1, 1999), som egentligen inte är ett kommunikationslager utan en omgivning för program att köra i. WML-läsaren finns i WAE och i senare versioner av WAP finns där även andra program.

4.7 Infrastruktur

För ett WAP-system behövs ett flertal olika delsystem (WAP Architecture, 1998). Dessa delar finns åskådliggjorda i figuren nedan.



Figur 7. *Infrastruktur i WAP*

Pilarna i figuren beskriver hur WAP-trafik går i det skandinaviska fallet med GSM-nät. De olika delarna beskrivs överskådligt nedan.

- **Mobiltelefon** – En mobiltelefon som har WAP inbyggt, men det skulle också kunna vara en dator eller en handdator (PDA). Från mobiltelefonen går trafiken till en access-server eller en SMSC. Se avsnitt 4.8.
- **Access-server** – Den dator som är kopplad till telefonledningen. Det är vanligtvis en dator med ett normalt modem för konsumentmarknaden eller motsvarande. Om meningen med accessservern är att koppla upp sig med ett TCP/IP-nät tilldelar den även ett IP-nummer och annan nödvändig information till den uppringande. Det är precis vad som händer om GSM-CSD används som bärare.
- **SMSC** – Om WAP-trafiken ska gå över SMS måste det finnas en SMSC, vilket är en centraldator för SMS.
- **WAP-gateway** – Det här en maskin som gör om WAP-trafiken, i det här fallet WSP, som kommer in från Access-servern eller SMSC:n till normal Internet-trafik. I det vanliga fallet med ”surfning” blir det HTTP-förfrågningar av det som ursprungligen kommer från mobiltelefonen. Se avsnitt 4.9.
- **WAP-tjänst**, även kallad applikationsserver – Gränssnittet utåt hos WAP-tjänsten är en vanlig webbserver, men bakom webbservern kan det finnas mycket avancerade system. Det viktiga är att det genereras WML-sidor som kan levereras till mobiltelefonen via WAP-gatewayen. En WAP-tjänst kan finnas både hos WAP-operatören och på Internet.
- **Operatör** – Operatören syftar på den teleoperatör som har ansvar för WAP-gatewayen, access-servern och mobiltelefonnätet. Det skulle kunna vara möjligt att en annan organisation än mobiltelefonoperatören driver access-server och WAP-gateway men med största sannolikhet kommer mobiltelefonoperatören att vara inblandad, även om det inte är ett krav.
- **Internet** – Inget hindrar WAP-trafiken från att gå över Internet så WAP-tjänsten skulle fysiskt kunna finnas var som helst på Internet.
- **Mobiltelefonnät** – Eftersom WAP är designat för mobila klienter så antas i figuren att trafiken kommer att gå från mobiltelefonen via det mobila telefonnätet till access-servern. Det finns dock inget i WAP i sig som kräver det utan det skulle kunna vara det fasta telenätet eller till och med ett fast nät även om WAP inte är optimerat för sådana fall.

4.8 Mobiltelefoner och andra WAP-klienter

Idag finns det inte så många klienter som stöder WAP, eller klarar av att visa WML-sidor. Den klassiska WAP-klienten är en mobiltelefon men det är också möjligt att använda en handdator eller en vanlig dator. Är det tillräckligt att bara kunna se WML-sidor är den enklaste metoden att använda den webbtjänst som finns på <URL:http://gelon.net/>, vilken kan visa WML-sidor i en Java-applet.

De klienter som finns att tillgå idag (december 1999) är inte många och alla har sina problem.

4.8.1 Ericsson MC218 kopplad till en Ericssontelefon

Ericsson MC218 är egentligen inte en mobiltelefon utan det är en handdator med inbyggd WAP-mjukvara. Den är gjord så att vem som helst enkelt kan koppla den till en Ericssontelefon och därmed få samma möjligheter som en vanlig WAP-telefon.

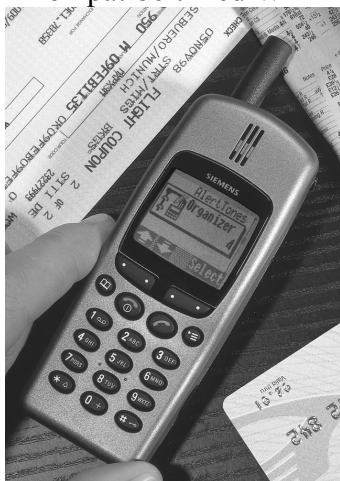


Figur 8. Ericsson MC218 tillsammans med en mobiltelefon Ericsson T28.

I figuren ovan används en Ericsson T28 för att kommunicera med mobiltelefonnätet. Mobiltelefonen har en Bluetoothenhet ansluten på undersidan. Bluetooth är ett system för radiokommunikation mycket korta sträckor och används ovan för att kommunicera med handdatorn.

4.8.2 Siemens S25

Siemens S25 var den första riktiga WAP-telefonen men har idag endast akademiskt värde eftersom den implementerar WAP 1.0 som idag inte används och dessutom är inkompatibelt med WAP 1.1.



Figur 9. Siemens S25 (© Siemens AG)

4.8.3 Nokia 7110

Nokia 7110 är den första WAP 1.1-telefonen som allmänheten kunde använda. Den har under oktober-december 1999 släppts i mindre serier, främst för utvecklare, men det är osäkert om den kan anses vara tillgänglig för allmänheten.



Figur 10. Nokia 7110 med delvis från dragen lucka.

4.8.4 Ericsson R320s

Ericsson har meddelat att de ska släppa sin mobiltelefon R320s under första kvartalet år 2000. Eftersom den ännu inte är släppt finns inte så mycket officiell information om den, mer än att det blir Ericssons allra första WAP-telefon, med R380s som nummer två.



Figur 11. Ericsson R320s

4.8.5 Ericsson R380s

Ericsson R380s är det som ska bli Ericssons andra WAP-telefon och skiljer sig från tidigare WAP-telefoner med sin stora skärm av "touchscreen"-typ. "Touchscreen" innebär att användaren kan använda skärmen för att skriva saker eller kanske peta på för att följa länkar. Den här telefonen är tänkt att komma under första halvåret år 2000.



Figur 12. Ericsson R380s med öppnad lucka

4.8.6 3Coms Palm Pilot

Det är möjligt att använda Palm Pilot för att läsa WAP-sidor. Exempelvis kan en Palm Pilot-innehavare använda programmet ”WAPman” (<http://virtuacom.com/wap/>) på sin Palm Pilot och sedan använda en kabel för att ansluta den till en mobiltelefon. I den Palm Piloten behöver användaren då köra ett mjukvarumodem, t.ex. GlobalPulse från TDK (<http://www.tdkgreycell.com/Products/softmob/>).

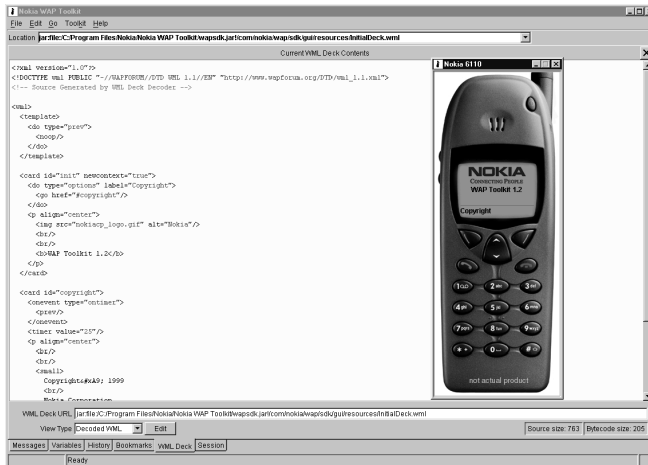
4.8.7 Simulatorer

Det finns också en serie simulatorer som gör att det blir möjligt att köra WAP på en vanlig dator. Såväl Ericsson, Nokia som Phone.com har tillverkat simulatorer som är fria att ladda ner och använda.



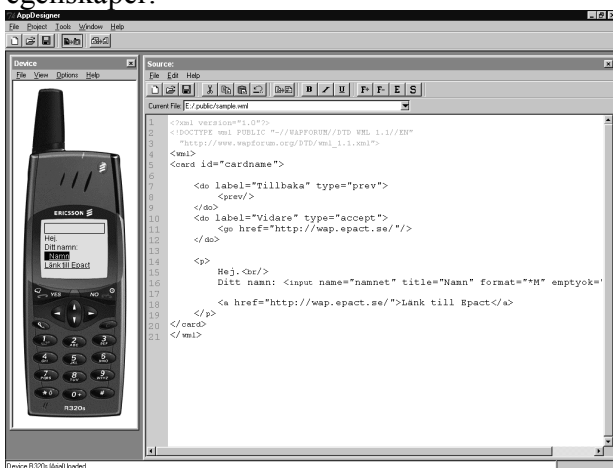
Figur 13. Nokia WAP SDK

Nokia SDK har enligt dokumentationen samma programvara som Nokia 7110 men den saknar de minnesbegränsningar som den riktiga mobiltelefonen har på grund av alltför lite minne i mobiltelefonen.



Figur 14. Nokia Toolkit 1.2

Nokia Toolkit innehåller redigeringsprogram för WML-kod och WMLScript-kod såväl som WBMP-bilder. Dessutom innehåller det en simulator där både 6110 och 6150 används som modeller trots att ingen av de telefonmodellerna har några WAP-egenskaper.



Figur 15. Ericsson WAP IDE

Ericssons WAP IDE har liksom Nokias Toolkit möjlighet att använda olika modeller som exempeltelefon. Dock så följer endast en telefonmodell, Ericsson R320s, med även om själva modellsystemet är mer flexibelt än det i Nokia Toolkit. I likhet med Nokias Toolkit finns även en utvecklingsmiljö med redigeringsprogram och hjälpmedel med i Ericsson WAP IDE.



Figur 16. Phone.com:s UP.Simulator

Phone.com:s simulator, UP.Simulator, är den enklaste och består förutom av själva telefonbilden endast av ett textgränssnitt. Till simulatormen följer modeller av ett halvt dussin olika telefonmodeller av olika tillverkare som licensierat Phone.com:s WML-läsare.

Eftersom det finns så få telefoner och det dessutom kan vara svårt att ens få tag på någon är det främst simulatorerna den intresserade kan använda och får förlita sig på vid testning och utveckling. En sak som märks redan nu, och framöver kommer att bli ännu tydligare är de stora skillnaderna i bildskärmsstorlek hos de olika klienterna. Det kan variera från ett fåtal rader till skärmar som kan jämföras med normala datorskärmar.

4.9 WAP-gateway

En central del i ett WAP-system är WAP-gatewayen (WAP Architecture, 1998). Den är porten mellan den mobila världen och Internet eller något annat datanät. En viktig uppgift som WAP-gatewayen har är att bytekoda data. När data kommer utifrån och den är okodad bytekodas den innan den skickas till mobiltelefonen och när data kommer bytekodad från mobiltelefonen avkodas den och skickas vidare.

För ögonblicket finns det inte så många gatewayar som blivit lanserade men det är rimligt att anta att de flesta, om inte alla, tillverkare av mobiltelefonsystem kommer att ha sin egen till att börja med. Under det här examensarbetet användes en betaversion av "Nokia WAP Server" vilket är en kombinerad webbserver och gateway avsedd för internt företagsbruk. Nokia har även en gateway avsedd för operatörer i sin "Nokia Artus Messaging Platform" (NAMP, 1999). Det finns även en helt fri gateway,

”Kannel”, under utveckling som enligt Kannel (2000) ska kunna användas i januari 2000.

4.10 WAP som framtidsteknik

Eftersom WAP är en såpass ny teknik är det fortfarande osäkert om det kommer att bli något stort av det. Som kommer att framgå i resten av rapporten finns det en del brister som kommer att behöva diskuteras.

En av osäkerheterna just nu är kostnaden för att använda WAP. Om surfning med WAP kommer att kosta lika mycket som vanliga samtal kommer det att bli för dyrt för de flesta att använda de tjänster som utvecklas, men eftersom operatörerna ännu inte lanserat några stora och omfattande WAP-satsningar är inget bestämt ännu.

Redan vid testning med simulatorer är det uppenbart att det inte är möjligt att få en WML-sida att se ut på liknande sätt i alla möjliga klienter. Inte ens om målgruppen begränsas till ett fåtal klienter blir det någon större förbättring utan det ser olika ut i varje WML-läsare. Den enda ljusglimten är att många telefoner använder samma WML-läsare och särskilt Phone.com:s WML-läsare tycks ha fått stor spridning, även om spridningen är delvis begränsad till Amerika och de mobiltelefonsystem de har där.

4.11 Sammanfattning

WAP har varit under utveckling sedan 1997 och är en teknik som syftar till att göra det möjligt att koppla ihop mobiltelefoner och datornätverk. För att använda WAP krävs det att användaren har en telefon som har WAP-funktioner och att det finns en möjlighet att ringa upp en dator (WAP-gateway) som kopplar mobiltelefonen till ett datornätverk.

För att göra en WAP-tjänst krävs det att den som gör tjänsten har tillgång till en webbserver. Tjänsteutvecklaren skriver sedan sidor i WML som är ett språk som liknar HTML. Det finns också möjligheter att göra skript genom att använda WMLScript vilket liknar JavaScript/ECMAScript.

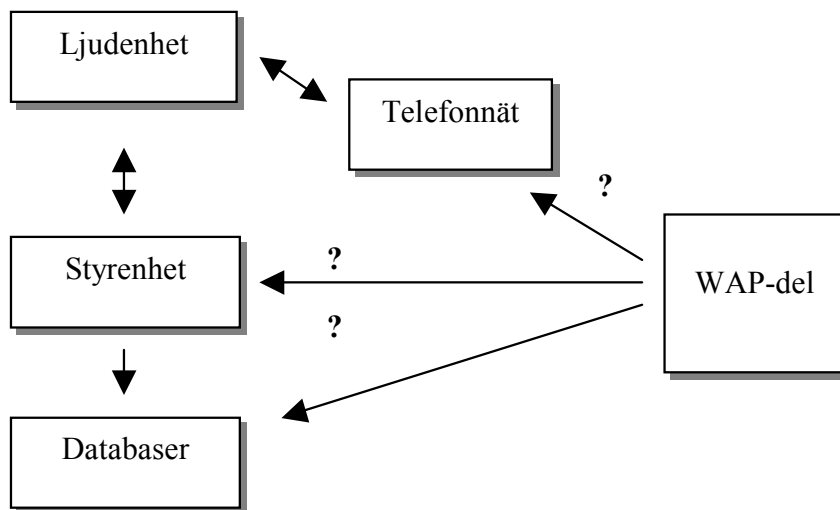
WAP är under ständig utveckling och det tillkommer funktioner för varje ny version av standarden. Det mesta talar dock för att WAP så småningom kommer att inlemmas i Internet på något sätt.

5 Röstbrevlåda för WAP-telefoner

Eftersom det redan finns väl utbyggda röstbrevlådesystem är det logiskt att utgå från dessa även om det skulle vara fullt möjligt att tänka sig ett röstbrevlådesystem med endast WAP-gränssnitt. Om planerna är att kombinera ett existerande system med ett traditionellt gränssnitt så finns det två sidor av det. Dels måste WAP-delen fungera som önskat och dels måste systemet fungera i kombination med det existerande röstbrevlådesystemet. I det här kapitlet tittar jag såväl på de olika sidorna (WAP respektive internt) som hur WAP skulle kunna användas nu jämfört med vilka möjligheter som finns om man ser framåt i tiden med ett ganska kort perspektiv (cirka 1-2 år).

5.1 Koppling till existerande system

På insidan av en röstbrevlåda är systemkonstruktören inte alls lika begränsad av standarder eftersom det är något som han eller hon till stora delar kan påverka själv. Att det inte är standardiserat innebär också att utseendet på röstbrevlådesystem som regel inte är något som finns som öppen information. Av den anledningen studerades främst hur MobilSvar såg ut eftersom MobilSvar är något Epact känner väl till genom sitt samarbete med Telia. Utan att gå ned på för låg nivå kan en röstbrevlåda sägas bestå av åtminstone tre logiska delar även om de fysiskt kan vara integrerade med varandra. Det är en ljudenhet kopplad till telefonnätet, en styrenhet samt en stor databas med abonnentinformation, brevlådeinformation, ljuddata och andra mer systemnära strukturer.



Figur 17. Möjliga kopplingar mellan WAP-system och existerande tjänst.

I den strukturen, vilken syns i figuren ovan, finns det tre olika ställen att koppla in sig, i figuren markerade med pilar med frågetecken. Det är möjligt att koppla in sig via telefonnätet mot ljudenheten, mot styrenheten eller mot databaserna. Det är inte säkert att det räcker att koppla in sig på ett ställe utan det kan behövas flera olika sätt att kommunicera med röstbrevlådan för att nå en önskad effekt, men nedanstående avsnitt behandlar frågan om vad som bör vara huvudangreppssätt.

5.1.1 Kommunikation via ljudenheten

Att koppla upp sig mot ljudenheten via telefonnätet har en stor fördel, nämligen att den som gör WAP-tjänsten inte behöver ha tillgång till de bakomliggande systemen i den existerande tjänsten eftersom vem som helst kan komma in den vägen genom att använda telefonnätet. Dock skulle det bli både svårt och långsamt. Det blir svårt eftersom WAP-systemet på något sätt måste tolka det som sägs och dessutom blir WAP-tjänsten beroende av underliggande underförstådd information som kan ändras när den ursprungliga tjänsten utvecklas. Det blir långsamt eftersom informationsöverföring med tal är betydligt långsammare än att överföra något digitalt eller specialkodat. Om den som utvecklar WAP-tjänsten samarbetar med ägaren av röstbrevlådesystemet är det möjligt att komma runt en del av de problemen genom att det i så fall blir möjligt att använda annan, effektivare ljudsignalering än tal för att överföra informationen, fast även med den förändringen är det en komplicerad och ineffektiv metod.

5.1.2 Kommunikation genom styrenheten

För att koppla upp sig mot styrenheten krävs i de flesta fall att det görs utökningar av den, såvida det inte redan finns ett API, programmeringsgränssnitt, som tillåter externa program att utföra de funktioner som behövs. Det finns flera stora fördelar med att koppla sig mot just styrenheten varav en är att det blir möjligt att använda en stor del av den logik som redan finns där och som systembyggaren annars hade behövt återskapa i

det externa programmet. En annan fördel är att det blir möjligt att återanvända API:et om andra program har liknande behov och vill koppla ihop sig med röstbrevlådetjänsten. Det kan till exempel vara en Internetversion av röstbrevlådan.

Nackdelen är att det kan vara svårt att göra de ändringar som krävs i styrenheten för att tillåta högre nivåers funktioner utifrån. Det beror helt på hur det är utformat från början. Den lösning som utvecklas blir dessutom väldigt beroende av just det icke-standardiserade gränssnittet.

5.1.3 Styrning direkt genom databasen

Att koppla upp sig direkt mot databasen har sina poänger. Om vi antar, vilket med bakgrund av känd information om MobilSvar¹ verkar troligt, att det finns en SQL-databas i botten så är det möjligt att använda standardiserade metoder för att komma åt den. Till exempel är det troligt att databasen kan kommunicera med protokollen ODBC eller JDBC eftersom en stor majoritet av alla kommersiella databaser idag har stöd för ODBC, JDBC eller båda två. Om så är fallet är det möjligt att ställa helt vanliga SQL-frågor och kommandon mot databasen för att få ut eller ändra på den information som WAP-tjänsten behöver.

Ett problem som kan uppstå är att få information från databasen vid förändringar, till exempel nyinkomna meddelanden. Många moderna databaser innehåller trigger- eller alert-funktioner som tillåter att program körs när vissa villkor uppfylls men det är inte alla som klarar det och det är inte säkert att det är lämpligt ur prestandasynvinkel om databasen är stor och har många transaktioner.

Ett annat potentiellt problem är om databasen är distribuerad och den saken inte är transparent för den som använder databashanteraren. I så fall måste det system som används för distribueringen i det existerande systemet dupliceras i WAP-systemet vilket kan vara ett svårt och avancerat jobb om inte resten av det existerande systemet kan användas.

En nackdel med hela angreppssättet är att även om SQL är standardiserat så är inte själva databaserna som hör till tjänsten det. Hur informationen om användare, meddelande och brevlådor är uppdelad och exakt vilken information som behövs eller används skulle kunna variera mycket mellan olika system. En förutsättning för att det ska vara möjligt att återanvända samma eller likartad kod på många ställen är därför att tabellerna som finns i databasen påminner om varandra mellan olika röstbrevlådesystem. Det kan till exempel vara fallet om olika operatörer bygger sina röstbrevlådor på samma grundplattform.

5.1.4 Sammanfattning

Med bakgrund av vad som sägs ovan rekommenderar jag att WAP-tjänsten i första hand kopplar upp sig genom styrenheten. Den innehåller mycket av den kod som WAP-

¹ MobilSvar använder en Oracledatabas.

tjänsten behöver, den har kontakt och kontroll över de andra enheterna och det blir möjligt att nå hög effektivitet och integrering genom att använda den delen.

Går det inte, eller är det inte lämpligt att använda styrenheten så är databaserna nästa logiska alternativ. Genom att manipulera dessa kan WAP-tjänsten få hög effektivitet och försäkra sig om att det finns tillgång till samma information som den ursprungliga tjänsten har. Enda poängen med att använda ljudenheterna direkt är att det inte krävs några ändringar hos det existerande systemet och att de i så fall kan användas utan direkt samverkan med ägaren till ljudenheterna. En intressant tanke är att göra ett system som samordnar olika operatörers röstbrevlådesystem och förser dem med ett gemensamt användargränssnitt.

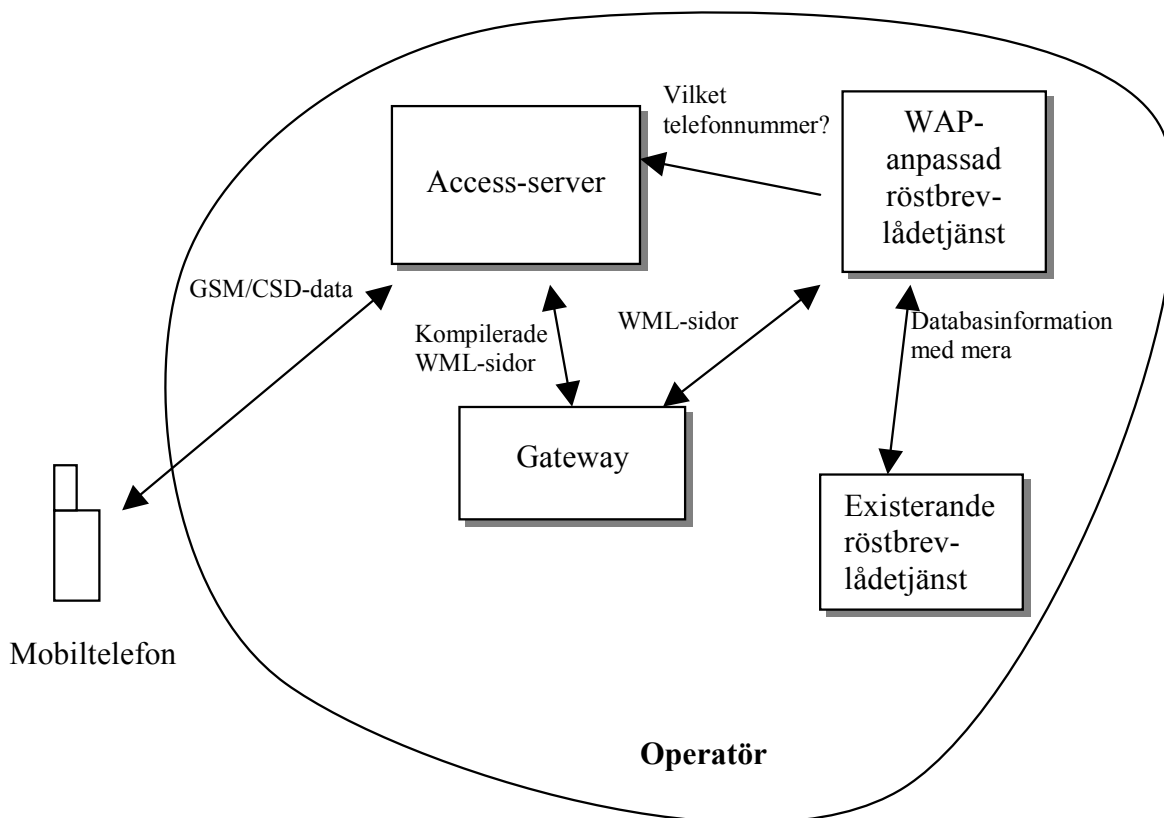
5.2 Idag – WAP 1.1

Det här kapitlet beskriver hur en WAP-lösning för en röstbrevlåda skulle kunna se ut om bara teknologi som finns tillgänglig just nu används. Problem och osäkerheter tas upp och erfarenheter från prototypen som tillverkats under examensarbetet redovisas.

För ögonblicket, är det version 1.1 av WAP-standarden som är den som används. Den allra senaste versionen, WAP 1.2, saknar än så länge såväl telefoner som serversystem. Just nu befinner sig WAP-tekniken i ett ganska omoget skede men med potential att utvecklas. Det är möjligt att det är ett medvetet val av standardiseringsorganet, WAP Forum, eftersom de måste ge tid för hårdvaru- och systemutvecklarna att bygga upp anpassade telefoner, mobilsystem och tjänster.

5.2.1 Systemförslag

Ett förslag till systemlösning syns i figuren nedan.



Figur 18. Systemförslag WAP-tjänst med WAP 1.1 över GSM-CSD

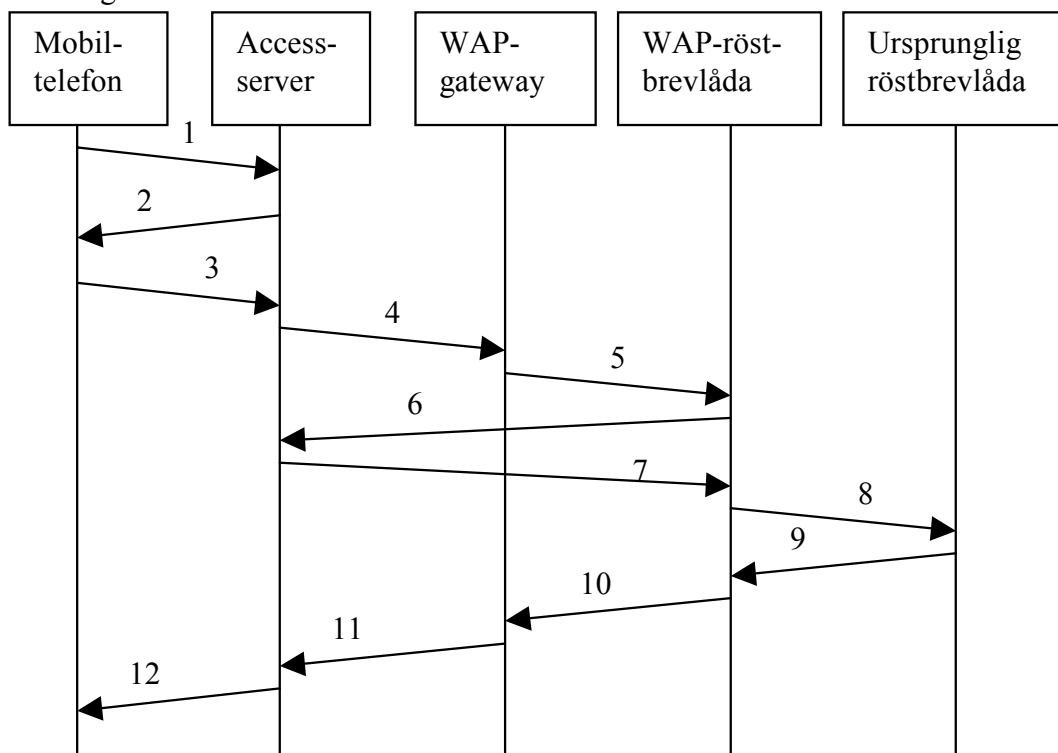
Figuren består av ett antal enheter som känns igen från kapitlet om WAP.

- Mobiltelefon – Det är en WAP-kapabel mobiltelefon, t.ex. Nokia 7110, Ericsson R320s eller R380s, eller motsvarande från andra tillverkare. Det kan också vara en dator eller handdator (PDA)
- Access-server – Det här är ett vanligt modem eller i fallet med en operatör snarare en stor modempool som dessutom redan finns hos de operatörer som har datatjänster för mobiltelefoner. När någon ringer upp görs normala behörighetskontroller genom till exempel RADIUS (RFC 2138, 1997), som är en standardiserad metod att hantera behörigheter när ett system har stora mängder användare. Efter det tilldelas mobiltelefonen ett IP-nummer och lite annan information som behövs för att använda TCP/IP-nätverk. Eftersom mobiltelefonen från och med den här punkten kommunicerar med hjälp av IP måste all telefoningssignalering tas om hand här. Telefoningssignaleringen innehåller information om varifrån samtalet kommer, vilket finns tillgängligt med hjälp av enkel nummerpresentation, men signaleringen innehåller också annan information

såsom vidarekopplingar och liknande som operatören vill ha tillgång till. Den information sparas hos accessservern tillsammans med det IP-nummer som användaren blivit tilldelad.

- Gateway – Den här maskinen kompilerar WML-sidorna till bytekod (se avsnitt 4.3.1) om de inte redan är kompilerade. Dessutom gör den om WAP-trafiken från WSP-trafik som i WAP-standarden till normal Internet-trafik och tvärtom i andra riktningen. I det vanliga fallet med ”surfning” blir det HTTP-förfrågningar av det som kommer in från mobiltelefonen.
- WAP-anpassad röstbrevlåda – Det här är själva hjärtat i tjänsten. Det skulle utåt kunna använda en vanlig webbserver.
- Röstbrevlådesystem – Ett normalt konventionellt röstbrevlådesystem som det ser ut idag fast eventuellt modifierat för att samarbeta med resten av WAP-systemet.

Kommunikationen mellan enheterna kan enklast visas med ett tidsdiagram. Tiden går nedåt i figuren.



Figur 19. Tidsdiagram för användning av WAP-röstbrevlåda

De numererade pilarna står för följande händelser.

1. Mobiltelefonen ringer upp accessservern.
2. Efter att ha kontrollerat behörigheter tilldelas mobiltelefonen ett IP-nummer som mobiltelefonen sedan använder för resten av kommunikationen.
3. Användaren önskar se vilka meddelanden som finns i sin röstbrevlåda och mobiltelefonen skickar därför en bytekodad begäran om en URL som motsvarar den tjänsten. Den kommer via modemmet först till accessservern som ett IP-paket.

4. Accessservern kollar vad IP-paketet som kom hade för adress och skickar vidare det till lämplig annan dator. I det här fallet är paketet adresserat till datorn som har WAP-gatewayen och därför skickas paketet till den datorn.
5. WAP-gatewayen får in IP-paketet och läser vad det är för något. Bytekoden packas upp till normal HTTP och URL:en tolkas. WAP-gatewayen noterar att det handlar om en förfrågan till WAP-röstbrevlådan och HTTP-förfrågan skickas därför vidare till det systemet.
6. WAP-röstbrevlådan noterar att det kommer en förfrågan från IP-nummer x.y.z.w och frågar därför Accessservern vad det IP-numret motsvarar för telefonnummer och eventuellt även annan signaleringsinformation.
7. Accessservern slår upp informationen om från vilket nummer personen med IP-nummer x.y.z.w ringer i den databas den skapat och skickar det tillbaka till WAP-tjänsten.
8. När numret och eventuellt annan signaleringsinformation är känd kan WAP-tjänsten kontakta den ursprungliga röstbrevlådan och fråga efter den information som WAP-tjänsten vill ha.
9. Den ursprungliga röstbrevlådan svarar med lämplig information.
10. WAP-tjänsten sammanställer WML-sidor som ger den information som användaren ska ha och skickar det till WAP-gatewayen.
11. WAP-gatewayen kompilerar WML-koden till bytekodad WML och skickar till Accessservern för vidare transport.
12. Accessservern tar det som kom från WAP-gatewayen och skickar över telenätet till mobiltelefonen.

5.2.2 Bärare

Det finns i dagsläget två möjliga bärare som det i Skandinavien finns både telefoner och servrar som klarar.

En möjlig bärare är GSM-CSD som är en datauppkoppling med hjälp av PPP, "Point-to-Point Protocol" (RFC 1661, 1994) mot en accessserver. PPP är en standard för att tunnla trafik över en förbindelse mellan två datorer. Efter att datauppkopplingen är gjord används UDP över IP-läget i WAP för kommunikation med gatewayen. För att använda den här bäraren idag krävs inga ändringar hos operatören så länge som det är möjligt att göra vanliga datauppkopplingar. Det ingår inte i alla abonnemang men finns åtminstone som tillvalstjänst hos alla operatörer. Det är den bäraren som antas överallt i den här rapporten där inget annat sägs.

Den andra bäraren som kan vara aktuell är SMS. Det har inte varit möjligt att testa SMS som bärare under utredningen eftersom det inte har funnits en SMSC att tillgå. Eftersom SMS dessutom har låg prestanda är det ändå inte en bra lösning. Det finns dock inget som hindrar att systemutvecklaren kompletterar Access-servern med en SMSC som klarar WAP så länge som det finns en WAP-gateway som klarar det. Den modell som används för att ta reda på den uppringandes nummer ovan behöver då inte användas eftersom den informationen följer med ett SMS-meddelande och därför kan tas om hand

av WAP-gatewayen. Information som WAP-gatewayen har tillgång till kan lätt skickas med till WAP-tjänsten som extra information i HTTP-huvudet.

5.2.3 Upplevd prestanda

Att det i figuren ovan antas att användaren kommer att köra WAP över GSM-CSD, det vill säga vad som i vanligt tal kallas för datauppkoppling, verkar rimligt med tanke på att det enda alternativet som finns idag är att kommunicera över SMS vilket kan ge stora fördröjningar. Att använda GSM-CSD innebär att förbindelsen får en överförings-hastighet på 9.6 kbit/s och uppkopplingstider på åtminstone 30 sekunder, vilket betyder att om användaren väljer att kolla sina meddelanden och inte redan är uppkopplad mot rätt WAP-gateway får han eller hon vänta minst ca 30 sekunder innan det kommer någon respons från röstbrevlådan.

Informella jämförelser med ett existerande system ger vid hand att det är längre tid än i existerande röstbrevlådetjänster. Enkla mätningar jag gjort mot Telias MobilSvar visar att det i MobilSvarsystemet tar ca 10 sekunder tills svarsrösten hörs och ca 15 sekunder tills första meddelandet börjar spelas upp.

Det är dock bara uppkopplingsfördröjningen som är stor hos GSM-CSD. De 9.6 kbit/s förbindelsen har (ca 1000 tecken per sekund) räcker gott och väl till att föra över varje kortlek på någon sekund. Den risk som finns i det här avseendet är att telefonen behöver göra en ny uppkoppling av någon anledning. Under tester med Nokia 7110 visade det sig att telefonen kopplar ned efter en viss tids inaktivitet på nätet vilket kan förekomma både om användaren är inaktiv eller om användaren håller sig till sidor inom samma kortlek.

5.2.4 Inställningar

Idag har ofta röstbrevlådesystem många inställningsmöjligheter. Det kan vara allt från hur många signaler som ska gå fram innan röstbrevlådan tar över, till när PIN-koden måste användas eller hur användaren vill bli informerad om nya meddelanden. Det gör att den gamla metoden att navigera sig genom menyer med knappatsen kan bli omständlig eftersom användaren måste hålla i minnet hela tiden var han eller hon befinner sig och vad de olika knapparna gör, samtidigt som varje knapp hela tiden byter betydelse. Här finns stora möjligheter att kraftigt förbättra användbarheten även om mobiltelefonen bara har en skärm med ett fåtal rader eller tecken.

Att ha ett WAP-system som låter användaren ändra inställningar är även med dagens teknologi en någorlunda lätt uppgift eftersom det i WML finns stöd för att göra formulär med till exempel menyalternativ.

5.2.5 Meddelandelisting

Det är relativt enkelt att använda WAP för att göra en lista på meddelanden och presentera den listan på ett tilltalande sätt. Att skärmarna är såpass små kan göra det svårt att få överblick över alla meddelanden men fortfarande är det lättare än med dagens talsystem, där till exempel MobilSvar inte ger användaren någon lista alls, utan

hela meddelandena efter varandra. Det problem systemet kan få är att inte hela listan får plats i kortleken på grund av den storleksbegränsning på ca 1400 bytes som finns i praktiken. Se avsnitt 6.1 för mer information om storleksbegränsningen.

5.2.6 Meddelandeavisering

Meddelandeavisering, att meddela abonnenten att det kommit ett meddelande, är något systemet inte kan utföra med version 1.1 av WAP om man begränsar sig till ren WAP-teknik. För att avisera ett meddelande måste servern på något sätt initiera en uppkoppling med abonnenten. Det kallas med WAP-terminologi för "Push" (Push, 1999) och är en del av WAP 1.2. Se avsnitt 5.3.3 för hur det kommer att påverka möjligheterna till meddelandeavisering.

Om man ser utanför WAP-området så finns möjligheter att utnyttja mer traditionella tekniker, såsom SMS, och där uppmana folk att gå till WAP-tjänsten. Det är inte lika smidigt som att direkt komma in i tjänstens WAP-del men det är inte ett uppenbart funktionshinder.

5.2.7 Meddelandeavlyssning

Meddelandeavlyssning, att lyssna på vad ett meddelande innehåller, blir även det svårt med nuvarande teknik. Det finns två stora skäl till detta.

1. Det finns inget sätt att med WAP 1.1 från mobiltelefonen i WAP-läge öppna en ljudförbindelse med en annan part.
2. Det finns bara en kanal mellan mobiltelefonen och mobilnätet och den behövs för dataöverföringar. Skulle SMS användas som bärare skulle det gå att klara sig utan att låsa kanalen.

Den möjliga metod som finns är alltså att SMS används som bärare och när en viss sida besöks på servern ringer servern upp telefonen. Användaren måste då svara och då läses meddelandet upp. När meddelandet är slut återvänder användaren till WAP-surfandet. Troligtvis har dock telefonen lämnat WAP-läge för samtalet även om specifikationen inte kräver något speciellt beteende. Ett annat problem är att det inte finns något säkert telefonnummer som kan användas för att nå den som använder tjänsten. Mer information om de svårigheterna finns i avsnitt 5.2.9.

Det är möjligt att komma runt hela den här problematiken om WAP-klienten visar ett meddelande, som från början var ett röstmeddelande, översatt till text, men det ligger utanför den här rapporten och möjligtvis ganska många år fram i tiden.

5.2.8 Inspelning av hälsningsmeddelande

Att spela in hälsningsmeddelande är lika svårt som att lyssna av ett meddelande. I båda fallen måste en talförbindelse kopplas upp mot en server, så det som sägs i avsnitt 5.2.7 gäller även här.

5.2.9 Identifiering

I traditionella röstbrevlådesystem används oftast signaleringsinformation eventuellt kombinerat med en kod för att identifiera och autentisera användare. I ett WAP-system har tjänsten inte direkt samma möjligheter. Som syns i systemskissen ovan försvinner signaleringsinformation ett par steg innan applikationsservern så den informationen kan inte användas direkt.

Direkt med WAP kan systemet använda ett inloggningsförfarande genom att kräva att användaren ska mata in ett användarnamn/telefonnummer och kod i ett vanligt formulär. Den informationen skickas sedan till servern som kontrollerar mot en lämplig databas och antingen godkänner eller nekar tillgång till tjänsten. Även om det skulle fungera kan det upplevas som jobbigt för användaren att behöva mata in såpass mycket information varje gång han eller hon ska använda tjänsten, så andra metoder bör övervägas. En metod som användes för den prototyp som gjordes för utredningen var att bygga en tjänst på accessservern som använder dess loggar för att matcha telefonnummer och IP-nummer.

På den accessserver som användes på Epect under utredningen används nummerpresentation när någon ringer upp och de nummer som rapporteras sparas i logfiler. För att få veta vilket telefonnummer ett visst IP-nummer motsvarar kontaktar WAP-tjänsten en tjänst på accessservern som tar ett IP-nummer som indata och efter att ha tolkat logfilerna rapporterar det telefonnummer som användaren ringde ifrån. Nackdelar med den här lösningen är att det inte fungerar när inte nummerpresentation fungerar vilket till exempel gäller för vissa växelsystem och hemligt nummer. Det är dock inget problem för en operatör som ändå har tillgång till den informationen genom signaleringen.

5.2.10 Cachningsproblem

Cachning innebär att någon, till exempel en klient, sparar undan sådant som den har hämtat så att den slipper hämta det igen om informationen behövs vid ett senare tillfälle.

Eftersom bandbredden är såpass liten och fördröjningarna kan bli långa vid uppkoppling i WAP idag är det viktigt att utnyttja cachen så mycket som möjligt. Tyvärr bidrar det till att information inte nyhämtas fast den kanske ändrats. I standarden (Caching, 1999) sägs att de vanliga HTTP-flaggorna för att stänga av cachning (RFC 2616, 1999) ska användas men försök med Nokia 7110 och Nokia WAP Server Beta 3 visar att så inte är fallet. För att undvika cachningen kan ett antal olika metoder användas. En metod som förkastades var att utnyttja att Nokia 7110 inte cachar sidor vars adress (URL) är längre än 255 tecken. Nu har dagens WAP-system en praktisk begränsning på ca 1400 tecken per kortlek så då skulle inte mer än cirka 5 sådana URL:er få plats i en kortlek. Mer information om den begränsningen finns i avsnitt 6.1. Istället för att använda extremt långa URL:er användes metoden att generera nya unika URL:er för varje ny sida. Det är inte möjligt att generera de unika URL:erna på serversidan eftersom de i så fall skulle kunna återanvändas då de hamnar i telefonens cache. Istället löstes det i prototypen med

följande WMLScriptfunktion som la till ett slumpmässigt 6-siffrigt tal till en dummyparameter för varje URL. Dummyargumentet ignorerades helt av servern.

```
extern function gotoUniqueURL(oldurl)
{
    /* The oldurl have to end with a ? and a parameter for
       this to work. */
    var newurl;
    newurl = oldurl+"&dummy=";
    newurl += Lang.random(999999);
    WMLBrowser.go(newurl);
}
```

Kodexempel 4. WMLScriptkod som går till en ny unik URL givet en bas-URL

Resultatet av WMLScriptanrop cachas inte och eftersom det finns 1 000 000 olika 6-siffriga tal blev det en ny unik URL med sannolikheten $999\,999/1\,000\,000 = 99,9999\%$. Det är fullt möjligt att öka antalet siffror ytterligare om risken för att få samma nummer som innan bedöms som för stor. Det finns ingen direkt maxgräns förutom att ju längre URL:erna blir, desto större plats tar de i kortleken. Det är möjligt att WAP-klienterna kommer att ha slumpvalsgeneratorer som är av en typ som inte kan ge samma tal två gånger i rad och i så fall är risken ännu mindre att ett nummer som finns på en sida i cachen ska återkomma.

URL:en som användes för att koden ovan skulle anropas såg då ut som

```
http://w/file_with_scripts.wmls#gotoUniqueURL("http://w/login?sid=12131433")
```

Kodexempel 5. URL som går till oldURL modifierad så att den är unik.

Resultatet skulle bli att man hämtade en sida med en URL liknande den nedan.

```
http://w/login?sid=12131433&dummy=43784
```

Kodexempel 6. URL skapad av unikifieringsprogrammet

5.2.11 Säkerhet

I traditionella röstbrevlådesystem bygger säkerheten normalt på två saker, vilket nummer användaren ringer från och/eller kunskapen om tjänstens kod. Eftersom det i GSM-nätet är mycket svårt att förfalska uppringande nummer så får den biten anses vara säker. Att koden är hemlig är i första hand upp till abonnenten men risker finns att någon avlyssnar koden när den sänds. Enligt en nyligen gjord upptäckt är nämligen det krypto som säkerheten i GSM bygger på inte kryptologiskt säker (Biryukov & Shamir, 1999). Det är dock inget stort problem för den vanliga användaren då det krävs att någon kan snappa upp en sändning för att överhuvudtaget kunna knäcka den och det är inte helt trivialt.

Så länge trafiken går inom en operatörs nät är det samma säkerhet för ett WAP-system som det är för ett traditionellt system. Det säkerhetslager som finns, WTLS, inför visserligen extra skydd men även utan det räcker det som finns gott och väl.

Saken kommer i en annan dager om en del av trafiken skulle gå på Internet. Då krävs det att WAP-tjänsten själv inför något speciellt skydd eftersom WAP-standarden inte specificerar något om hur data ska skyddas mellan tjänsten och gatewayen. Den kryptering som finns specificerad gäller endast mellan klienten och gatewayen.

5.2.12 Sammanfattning

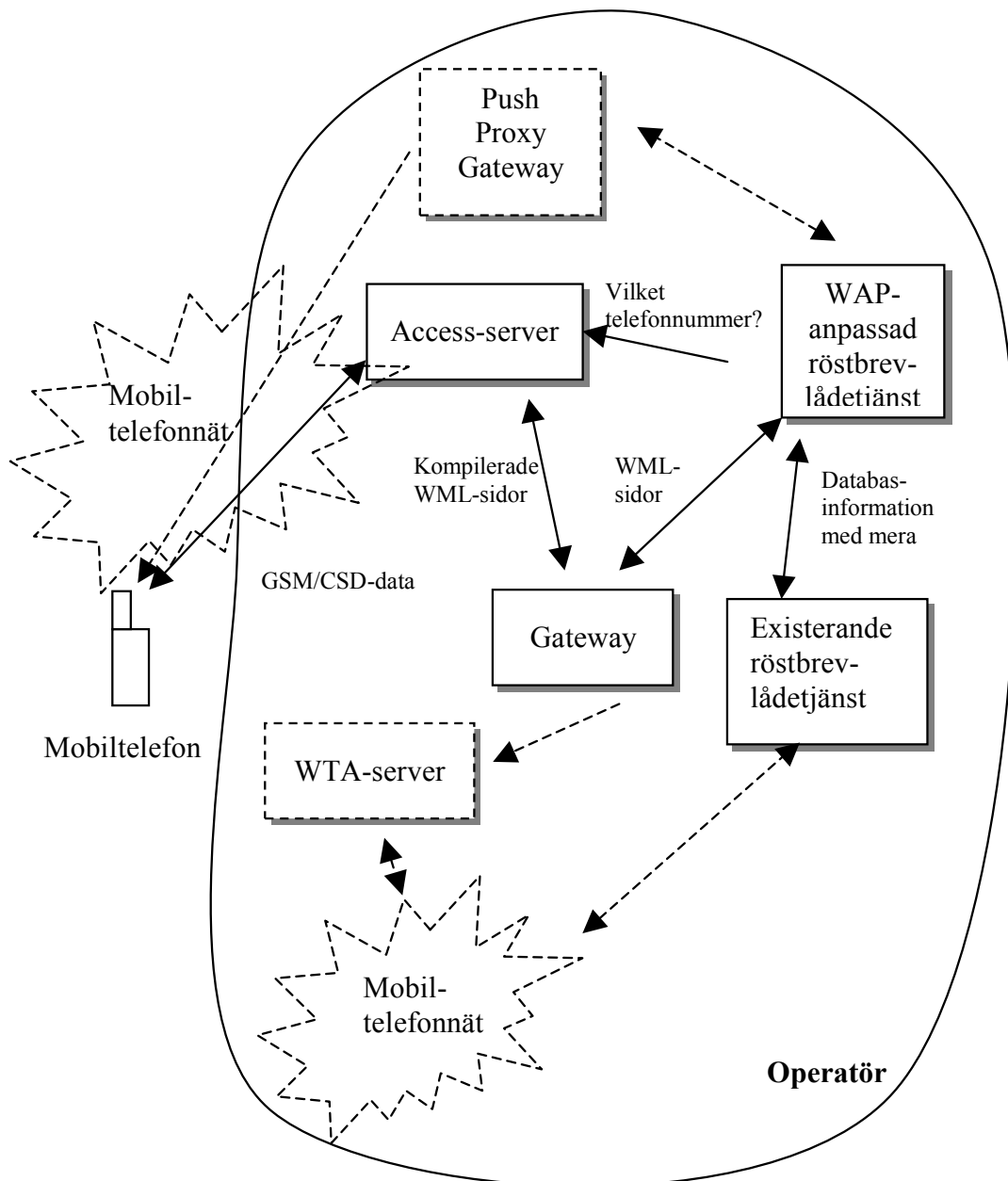
Med WAP 1.1 är det inte möjligt att göra ett tillräckligt användbart röstbrevlådesystem. Det finns möjlighet att göra allt som behövs utom just en ljuduppkoppling vilket tyvärr är ett mycket stort problem om någon vill göra en röstbrevlåda. Vad som är möjligt att göra med WAP är att tillverka ett sorts administrations- och övervakningsgränssnitt till en röstbrevlåda så att en användare kan se vad det finns för meddelanden och ändra sina inställningar. Det skulle kunna vara ett första steg i väntan på vidare teknikutveckling, men det kan inte på något sätt ersätta de existerande systemen.

5.3 *Förändringar när WAP 1.2 blir allmänt tillgänglig*

Som redan noterats ovan saknas idag möjligheter att göra en del av det som skulle behövas för att få till en WAP-tjänst med samma funktionalitet som de traditionella tjänsterna har idag. I det här avsnittet försöker jag reda ut vilka skillnader som är väsentliga mellan WAP 1.1 och WAP 1.2. Hur WAP 1.2 påverkar möjligheterna att lösa en del problem tas också upp.

5.3.1 Systemförslag

Det här är en skiss av hur det skulle kunna se ut i ett WAP 1.2-system.



Figur 20. Skiss av struktur för en WAP 1.2-baserad implementering

De nya delarna i figuren är streckade för att enkelt kunna skilja den från den tidigare skissen. Det finns två nya lådor i skissen, en "Push Proxy Gateway" och en WTA-

server. En "Push Proxy Gateway" är en dator som behövs för att meddela en mobiltelefon att det kommit nya meddelanden eller liknande information. Se avsnitt 5.3.3 för mer information om "Push". WTA-servern är en dator som kopplar ihop mobiltelefonnätets styrande delar med WAP-systemet. För mer information om den, se avsnitt 5.3.2.

Det finns också två bubblor som symboliserar (samma) mobiltelefonnät vilket naturligtvis funnits underförstått och fortfarande finns mellan mobiltelefonen och accesservern i den tidigare systemskissen. Det nya som gör det intressant att rita ut dem explicit är att man i och med de utökningar som finns i WAP 1.2 inte bara har mobiltelefonnätet som något att kommunicera över utan nu kan en WAP-tjänst även kommunicera *med* mobiltelefonsystemet. Till exempel kan en tjänst genom att använda WMLScript fråga ett GSM-nät i vilken "cell" mobiltelefonen befinner sig i, se avsnitt 5.3.2 för mer information.

5.3.2 WTA/WTAI – Meddelandeavlyssning

Med i WAP 1.2 följer delstandarderna WTA, "Wireless Telephony Application" (WTA, 1999), som behandlar hur telefonfunktioner ska kunna utnyttjas från WML och WMLScript. Att använda telefonfunktioner, såsom att ringa upp någon eller att svara på ett samtal, är något som inte alls går i WAP 1.1. Det är också det som ger de problem som nämns i tidigare avsnitt om meddelandeavlyssning och meddelandeinspelning.

Det finns flera delar av WTA som är intressanta. Det är till att börja med möjligt att aktivt initiera telefonhändelser, till exempel ringa upp någon, skicka DTMF-toner, svara eller avsluta ett samtal. Det finns också utökningar som är bärarberoende i högre grad än normalpaketet. När GSM är bärare ingår i WTA möjligheten att få ut vilken GSM-cell telefonen för tillfället befinner sig i. Just denna funktion är dock inte intressant såvida inte information om var olika celler befinner sig finns tillgänglig. Utan den informationen blir det på sin höjd möjligt att konstatera att användare finns i samma cell.

En annan minst lika intressanta del, särskilt om den kombineras med de vanliga telefonfunktionerna, är att det är möjligt att stoppa in händelsehanterare. Händelsehanterarna är små program som körs när vissa händelser inträffar. Händelser som inkommande telefonsamtal eller att ett samtal slutar ingår bland de händelser som tjänsteutvecklaren kan definiera egna program att ta hand om.

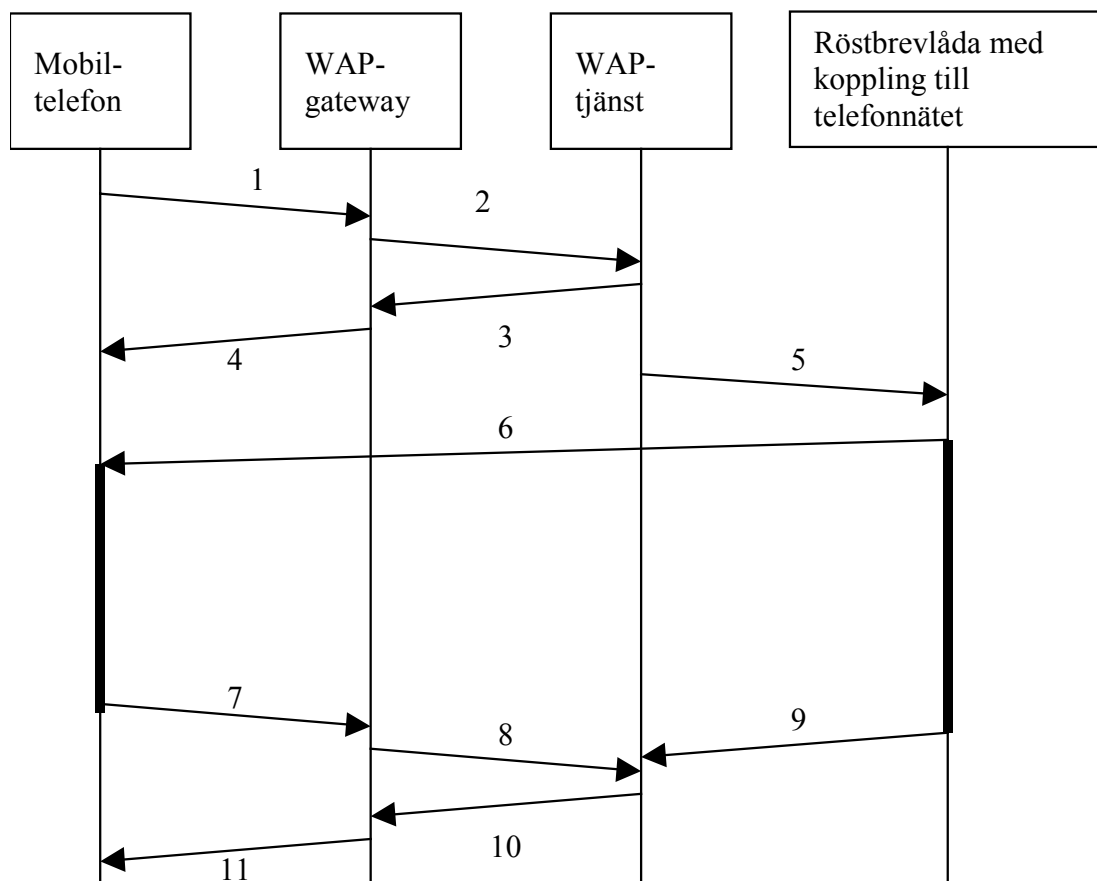
WTA påverkar flera delar av WAP-systemet. Det behövs en utökad WML-läsare i telefonen, vilket innebär antingen nya telefoner eller uppdaterad programvara i existerande telefoner. WTA inför möjligheten att hämta flera kortlekar och WMLScript-filer på en gång och lagra dessa i mobiltelefonen för senare användning. Det gör att tjänsten i det allmänna fallet förlorar mindre när mobiltelefonen måste stänga ned dataförbindelsen för att kunna använda telefonfunktioner, men om sidorna är dynamiska fungerar naturligtvis inte det som en lösning.

På serversidan behövs det en ny del i form av en WTA-server. Det är den som svarar på mobiltelefonens frågor om telefonnätet.

Det finns två sätt att komma åt funktionaliteten i WTA, det vill säga de funktioner som ligger i WTAI, vilket är namnet på själva gränssnittet mellan WML-delen och telefonerna. Alla funktioner kan skrivas som URL:er, vilket innebär att det är möjligt att ha dem överallt i WML-koden där det är möjligt att ha URL:er. Den mest aktuella platsen är länkar så att det är möjligt att ha en länk som utför något, t.ex. ringer upp någon. Det andra sättet är att på ett mer traditionellt sätt för en programmerare anropa funktionerna i WTAI på normalt sätt från WMLScript.

De två egenskaperna är mycket användbara om en tjänst vill koppla upp en ljudförbindelse från WAP. Det finns flera olika sätt som alla har sina för- och nackdelar. Ett sätt är att informera servern när användaren vill lyssna på ett meddelande. Servern plockar då fram meddelandet och ringer upp telefonen (förutsätter att servern känner till telefonnumret på telefonen) som då automatsvarar. Givetvis kontrolleras först att det är servern som ringer och inte någon annan som råkat ringa i samma ögonblick. Det för oss till ett av problemen med den metoden. Om telefonen redan används kommer det att vara upptaget och det är inte möjligt att höra meddelandet. Till exempel kan datauppkopplingen göra att telefonlinan (bildligt talat) är upptagen. Även om datauppkopplingen kan brytas innan servern ringer så är inte det en perfekt lösning eftersom mobiltelefonen då måste göra en ny uppkoppling vilket redan konstaterats är en långsam process som man inte vill göra särskilt ofta. En förutsättning för att det här ska fungera bra är därför att det finns tillgång till en metod för att göra snabba upp- och nedkopplingar, vilket inte finns med nuvarande system men väl kommer med framtida generationer av mobiltelefonsystem, med början med GPRS.

Ett annat sätt att lösa det är att systemet låter telefonen ringa upp servern när användaren vill avlyssna ett meddelande. Då skulle det vara tänkbart att ta bort kravet att servern måste känna till telefonnumret till den som använder WAP-tjänsten. WAP-telefonen ringer bara upp servern och antingen genom att slå ett speciellt nummer eller genom att skicka information med DTMF-signaler informerar den om vilket meddelande som önskas höras. Frågan är dock om systemet inte fortfarande vill veta vem ("vem" som i vilken telefon) som får höra meddelandet för att inte obehöriga ska kunna ringa och lyssna av andras meddelanden. Som alternativ eller kompletterande metod skulle klienten kunna autentisera sig med en hemlig kod som mobiltelefonen skickar som DTMF-signaler, men det skulle kunna verka störande på användaren att höra en serie pip före varje meddelande.



Figur 21. Tidsdiagram för meddelandeuppspelning

I figuren ovan beskrivs ett möjligt händelseförlopp för att lyssna av ett meddelande.

1. Användaren av WAP-tjänsten markerar att han/hon vill lyssna av ett meddelande och den uppmaningen skickas till gatewayen.
2. Gatewayen vidarebefordrar uppmaningen till WAP-tjänsten.
3. WAP-tjänsten skickar en bekräftelse i form av en WML-sida och installerar, alternativt aktiverar, samtidigt händelsehanterare som dels kommer att svara när röstbrevlådan ringer och dels kommer att koppla upp mot WAP-tjänsten igen när samtalet är slut.
4. Gatewayen vidarebefordrar bekräftelsen till mobiltelefonen.
5. När bekräftelsen är skickad och kvitterad av mobiltelefonen instrueras röstbrevlådan att ringa upp mobiltelefonen.
6. Röstbrevlådan ringer upp mobiltelefonen som automatsvarar, helst utan att användaren märker att någon ringer upp. Så fort mobiltelefonen svarat spelas meddelandet upp. Lämpligen fungerar DTMF-signaler här så att användaren kan spola fram och tillbaka i meddelandet samt avbryta meddelandet.
7. När meddelandet är slut vilket märks genom att förbindelsen bryts skickas ett informationsmeddelande om det till gatewayen.

8. Gatewayen vidarebefordrar informationsmeddelandet till WAP-tjänsten.
9. Röstbrevlådan informerar också WAP-tjänsten om att förbindelsen brutits.
10. En ny WML-sida skickas till gatewayen.
11. Gatewayen vidarebefordrar WML-sidan till mobiltelefonen.

5.3.3 "Push" – Meddelandeavisering

Som noterades i avsnitt 5.2.6 kan man inte göra meddelandeavisering med WAP-teknik som det ser ut idag. Med meddelandeavisering menas alltså att informera abonnenten att det kommit nya meddelanden och se till att abonnenten smidigt och enkelt kan få upp en lista i WAP med de olika meddelandena.

Den bristen i WAP 1.1 är åtgärdad i WAP 1.2 som tillåter "push", det vill säga att initiera en uppkoppling från servern istället för från WAP-klienten som är den enda möjlighet som finns idag. Det är inte riktigt helt och hållet "push" i den klassiska bemärkelsen eftersom inte själva sidan skickas till klienten. Istället skickas en uppmaning ("Service Indication") till klienten som har ett speciellt program för att hantera sådana uppmaningar. Det programmet studerar vad uppmaningen vill ska hända, kontrollerar det mot en del interna regler som säger vad programmet ska lyssna på för uppmaningar och, om allt är korrekt, utför uppmaningen. Det normala fallet är att det är en uppmaning att hämta en viss sida, men även WTA använder sådana uppmaningar för att få information om vissa händelser. Det kan krävas ganska mycket för att följa en uppmaning. I värsta fall är WAP-systemet nedkopplat från gatewayen och mobiltelefonnätet och i så fall måste klienten först koppla upp innan något annat kan göras.

Liksom WTA kräver "push" en del utökningar i själva infrastrukturen. Förutom de nya programmen i mobiltelefonen behövs det åtminstone en PPG ("Push Proxy Gateway") hos operatören. Det är en maskin som ansvarar för att meddelanden som kommer från Internet eller lokala nät når mobiltelefonen. Den håller även koll på vad som händer med alla försök att kontakta mobiltelefonen och meddelar den som initierade operationen om resultatet. Alla "push"-operationer måste gå genom en PPG som mobiltelefonen litar på.

I fallet med en WAP-röstbrevlåda är det enkelt att hänvisa en person till en viss WML-sida när någon spelar in ett meddelande i personens röstbrevlåda eller om något annat händer som röstbrevlådeanvändaren vill bli informerad om, så snart det finns mobiltelefoner och serversystem som har stöd för "Push" i den form den beskrivs i WAP 1.2-specifikationen.

5.3.4 Säkerhet

I och med WTA och "push" kommer det in en del nya områden vars säkerhet behöver studeras. Med WTA är det känsligt ur flera synvinklar att det fungerar som det är tänkt så att inte telefonsamtal rings automatiskt eller inkommande telefonsamtal skickas vidare. Vad förslaget till WTA-specifikation gör för att behandla sådana frågor är att de kräver att WTLS (säkerhetslagret) används så att inga uppkopplingar kan komma från otillåtna gatewayar. Vidare kräver specifikationen att telefonanvändaren själv

godkänner (lämpligen med en ja/nej-fråga på skärmen) vissa operationer. Vad som måste godkännas beror delvis på vem som initierar kommandot. WAP Forum skiljer i WTAI-specifikationen på kommandon som initieras inifrån "nätverket" och kommandon som initieras från en tredje part.

När det gäller "push" finns det ett par olika säkerhetsdelar. Först och främst accepterar bara klienten uppmaningar som kommer från godkända PPG:er. Sedan är det upp till varje klient om den vill följa den uppmaningen eller inte. Slutligen finns det även filtermöjligheter i PPG:en. Om den tillhör en operatör skulle den (som ett exempel) filtrera bort sådana uppmaningar som kommer från en konkurrent.

5.3.5 Alternativa tekniker

Om en tjänst vill föra ut information till skärmarna på mobiltelefoner på något sätt finns det idag inte så många olika metoder att välja på. Redan nu används SMS för att meddela om nya inkomna meddelanden men utöver det finns det inget som fungerar på ett flertal olika telefoner.

Det pratas en del om att använda SIM Toolkit, vilket är ett programmeringsgränssnitt mot SIM-kortet för att distribuera program. Det skulle eventuellt kunna vara möjligt men verkar innebära minst lika många problem som WAP.

5.3.6 Sammanfattning

I WAP 1.2 ingår det som saknades för att göra ett bra röstbrevlådesystem med WAP 1.1. Med hjälp av WTA ("Wireless Telephony Application") blir det möjligt att göra talkopplingar så att meddelanden både kan spelas in och lyssnas av. Med hjälp av de "push"-funktioner som finns i WAP 1.2 blir det möjligt att koppla upp mot en WAP-tjänst automatiskt om det kommer meddelanden.

5.4 Efter WAP 1.2

Det tempo som WAP Forum haft senaste året talar för att det kommer en uppdaterad WAP-standard ungefär en gång i halvåret, men det är osäkert hur det verkligen blir. En sak som i alla fall verkar klar är att mycket arbete kommer att läggas ned för att göra WAP alltmer Internetkompatibelt. WAP är i många avseenden utvecklat för de begränsningar som finns hos mobiltelefoner och mobilnät idag men många av de begränsningarna kommer så småningom att försvinna. Telefoner får både mer minne och mer processorkraft, skärmarna blir åtminstone lite större och det är inte otänkbart att de får högre upplösning eller att helt nya tekniker såsom projektion på näthinnan används istället för traditionella skärmar. De mobila näten kommer också att få allt större bandbredd. Redan med GPRS blir det möjligt att öka bandbredden och med UMTS som ska ersätta GPRS kan datauppkopplingar komma upp i 2 Mbit/s vilket är betydligt mycket bättre än vad många människor som använder Internet idag har tillgång till. Den största fördelen med GPRS är dock att de långa uppkopplingstiderna kan undvikas. Se avsnitt 4.6.1 för mer detaljer.

Så småningom kan man med största sannolikhet räkna med att gränserna mellan WAP och Internet flyter ut och försvinner i och med att mobiltelefonerna får kapacitet att hantera de mer komplexa standarder som används på Internet idag, till exempel HTML. Det är dock inte troligt att funktionalitet som kommit med WAP tills dess och som används försvinner helt utan troligen kommer de istället att vidareutvecklas och anpassas för Internet och därefter bli upphöjda till Internetstandarder. Det kan till exempel gälla WTA som inte har någon motsvarighet på Internet. Redan nu har W3C som fastställer HTML-standarder och WAP Forum meddelat att de tänker samarbeta för att utveckla standarder för webben som ska passa mobila terminaler (W3C, 1999)

5.5 Sammanfattning

Det är inte idag möjligt att göra en WAP-tjänst för röstbrevlådor med full funktionalitet eftersom det saknas möjlighet att koppla ihop telefonfunktioner och klassiska WAP-funktioner i WAP 1.1. Det innebär att användarna inte enkelt kan lyssna av ett meddelande eller spela in ett hälsningsmeddelande.

Det förändras teoretiskt med WAP 1.2 där det ingår funktioner som ska möjliggöra just en koppling mellan en mobiltelefons telefondel och WAP-delen. Dock måste det lanseras mobiltelefoner och WAP-servrar med stöd för WAP 1.2 innan de nya funktionerna kan användas.

6 Kvarvarande problem

Det här kapitlet tar lite översiktligt upp möjliga problem som upptäckts under arbetets gång eller som medvetet lämnats utan behandling.

6.1 Storleksbegränsning av kortlekar

Även om inte WAP-standarden säger något om det så är det just nu de facto-standard att telefonerna klarar kortlekar och bilder som är maximalt cirka 1400 bytes stora (Nokia 7110 klarar till exempel 1397 bytes). Den storleken är efter att kortlekarna bytekodats, så det är lite svårt att utifrån WML-koden manuellt eller maskinellt bestämma hur stor bytekoden blir. Detta eftersom en WML-tillverkare i allmänhet inte känner till vilken bytekodare som kommer att användas vid kodningen. Det är möjligt att sända kortlekar som är kodade från början men det är endast praktiskt vid statiska sidor som kan kodas en gång och sändas många gånger och inte dynamiska sidor som måste kodas varje gång.

När man tittar på röstbrevlådefallet är det listan på meddelanden som kan växa och ganska lätt bli för stor för en kortlek. En naiv lösning är att granska alla kortlekar innan de skickas och om de skulle kunna vara för stora så delas de upp och det skapas länkar mellan dem. Det finns en hel del problem med det och frågeställningar om hur ett WAP-system ska hantera länkar in till mitten av en delad kortlek. Systemtillverkaren blir antagligen tvungen att redan vid designen av det grundläggande WML-genererande systemet på något sätt få in lösningar på de problemen.

6.2 Användargränssnitt

Som nämnts i texten om WAP och WML är målsättningen med WML inte att beskriva hur det ska se ut på skärmen utan att beskriva hur sidan ska fungera och sedan låta telefонтillverkaren själv välja hur det ska presenteras. Det innebär att det kan vara stora skillnader mellan olika WML-läsare. Under examensarbetet har följande olika WML-läsare undersökts.

- Nokia 7110 (firmware 04.69)
- Nokia SDK 3.18 (med samma programvara som Nokia 7110)
- Nokia Toolkit 1.1 och 1.2 med 6150-beteende.
- Nokia Toolkit 1.1 och 1.2 med 6110-beteende.

- Ericssons WAP-IDE
- Phone.com:s UP.Simulator 3.2 (Som har samma programvara som ett tjugotal telefoner)

Som testkod har jag använt följande kod som har en länk, ett fält för textinmatning och lite text med fet stil. Koden innehåller även en "accept"-knapp som ska märkas "Vidare" och en "prev"-knapp som ska märkas "Tillbaka". Det finns inget i specifikationen som garanterar att de funktionerna kommer att hamna på lättillgängliga knappar egentligen, men de är de som är mest troliga att göra det.

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="cardname">

  <do label="Tillbaka" type="prev">
    <prev/>
  </do>
  <do label="Vidare" type="accept">
    <go href="http://wap.epact.se/" />
  </do>

  <p>
    <b>Hej!</b><br/>
    Ditt namn:
    <input name="namnet" title="Namn"
    format="*M" emptyok="true"/>

    <a href="http://wap.epact.se/">Länk till Epact</a>
  </p>
</card>
</wml>

```

Kodexempel 7. Exempelsida för klientjämförelser



Figur 22. Nokia Toolkit i 6150-läge

I Nokia Toolkit i 6150-läge får själva sidan det utseende som kan förväntas men knapparna "Vidare" och "Tillbaka" syns inte till. Däremot finns en "Back" som motsvarar "Tillbaka", men av någon anledning har den inte döpts om. "Options"-valet ger en meny där man kan hitta "Vidare".



Figur 23. Nokia Toolkit i 6110-läge

Samma Toolkit i 6110-läge har inte alls samma problem. Här syns såväl "Vidare" som "Tillbaka" och den meny som i 6150-läget fanns under "Options" verkar ha försvunnit.



Figur 24. Nokia SDK (samma utseende som Nokia 7110)

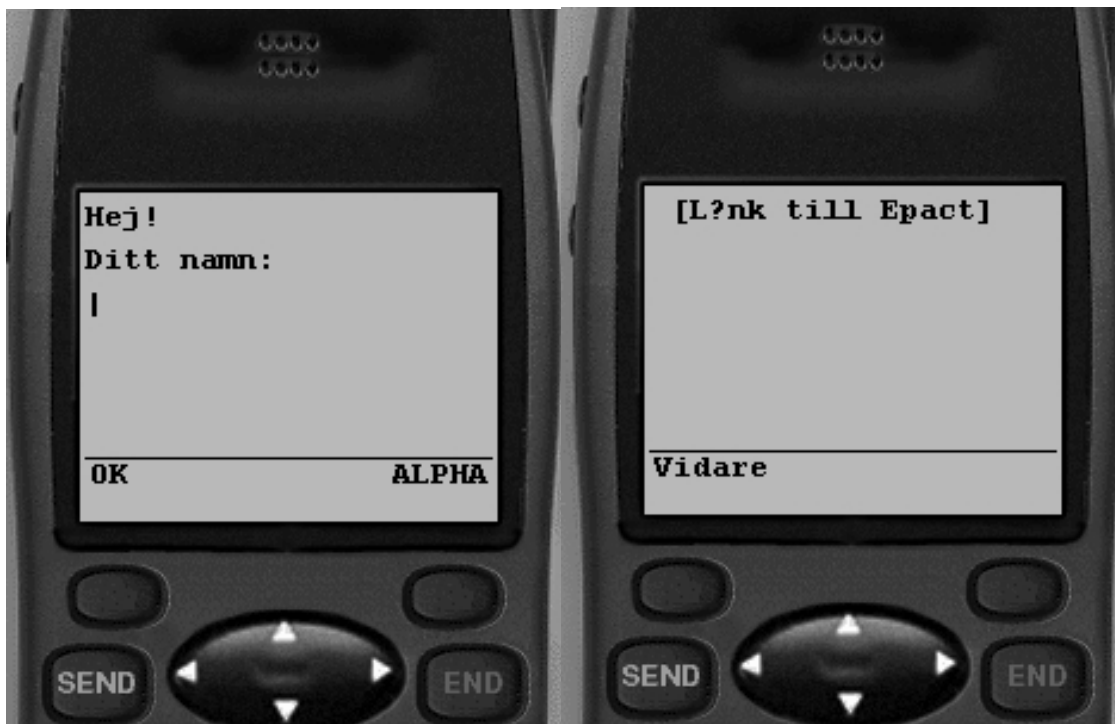
På en riktig Nokia 7110 saknas först och främst fetstil. "Hej!" visas med samma typsnitt som alla andra ord. Vidare så sätter telefonen själv in en radbrytning innan textfältet vilket verkar onödigt. Slutligen så syns varken "Vidare" eller "Tillbaka" till. Hade inte "prev"-aktiviteten döpts om så hade det funnits en "back" till höger om "Options" men nu finner man både "Tillbaka" och "Vidare" först om man ser efter vad som finns i "Options"-menyn.



Figur 25. Ericsson IDE i 320s-läge

Ericssons IDE i R320s-läge (det enda läget än så länge) har ytterligare ett eget utseende. Här visas datornamnet som WML-sidan hämtades ifrån som titel längst upp.

Formulärfältet hamnar på en egen rad men det kan bero på radbredden. Här visas också namnet på textfältet, "Namn", vilket inte har skett i någon av de andra WML-läsarna. Den här simulatoren/telefonen visar inte knappbeskrivningar alls utan knapparna förutsätts göra något vettigt utan att man ska behöva läsa vad de gör.



Figur 26. UP.simulator innan och efter att namnet skrivits in

UP.Simulator har sin alldeles egna stil. Den stannar upp när den kommer till ett textfält och väntar på inmatning (vänstra bilden). Inget spår syns av resten av sidan eller "Tillbaka" och "Vidare"-knapparna. Dessutom så visar inte heller den här WML-läsaren text med fetstil.

Först när något matats in i textfältet (högra bilden) visas resten av sidan men då har den övre halvan försvunnit. Man ser också hur WML-läsare misslyckas med att visa tecknet "ä" i "länk" och istället visar ett frågetecken.

Sammanfattningsvis så är det uppenbart att det är stora skillnader mellan olika miljöer och detta kan helt klart bli ett problem.

6.3 Stordrift

När en tjänst designas är antalet möjliga användare och vilka önskemål som finns på tillgänglighet aspekter som spelar in. Om målet är att bygga en tjänst med potential för flera miljoner användare krävs det att det redan från början skapas möjligheter för att distribuera och sprida den belastning så många användare skulle innebära. Det finns inget direkt i de föreslagna modellerna som skulle fungera som flaskhals som allt måste

gå genom, utan de delarna ligger i så fall hos de existerande systemen, såsom databaser med mera.

Trots att det inte verkar finnas några direkta problem med att parallellisera WAP-delar som gateway, access-servrar med mera, så måste systemutvecklaren ha det i åtanke när han eller hon försöker designa ett större system än det prototypsystem som utvecklats under examensarbetet.

6.4 Marknaden

För att det ska bli meningsfullt att göra en sådan här tjänst måste det också finnas intresse från marknaden, det vill säga de som använder tjänsterna och de som tillhandahåller tjänsterna. Inte heller det har varit i fokus under utredningen även om det verkar som om de möjligheter som WAP ger skulle kunna vara väldigt attraktiva för såväl operatörer som operatörernas kunder.

6.5 Sammanfattning

Det finns fortfarande en del saker som måste utredas för en slutgiltig implementering. Det finns till exempel problem med att kortlekar endast kan vara ca 1400 bytes stora och hur system för riktigt stora operatörer ska byggas.

7 Ordförklaringar och förkortningar

I följande tabeller finns ord och förkortningar som används i rapporten tillsammans med en förklaring av vad de betyder.

Ord	Förklaring
Bandbredd	Den maximala mängd data som kan föras över per tidsenhet.
Fördröjning	I samband med nätverk betyder fördröjning den tid det tar för data att komma från en punkt till en annan.
Gateway	I WAP-sammanhang den dator som sitter mellan WAP-telefonen och det riktiga datornätverket.
Klient	När ordet klient använts i rapporten syftas, om inget annat sägs, på mobiltelefonen.
Kortlek	En samling sidor, kort, i en WML-fil.
”Push”	”Push”, knuffa, trycka, innebär att information påtvingas klienten istället för att klienten hämtar information själv vilket brukar kallas ”pull”, dra.
Röstbrevlåda	Tjänst som låter uppringande lämna meddelanden och ägaren att lyssna av dessa. Ordet kan också syfta på en specifik persons meddelandelåda i tjänsten.
Signalering	Med en samtalsuppkoppling följer information om vidarekopplingar, den uppringande och den uppringdas telefonnummer, med mera. Det kallas signalering.
Taggar	Identifierare mellan vinkelparenteser (”större än” och ”mindre än”-tecken) som används i WML, HTML och i många andra uppmärkningsspråk.
WML-läsare	Det program som används för att visa WML-sidor.

Tabell 2. Ordförklaringar

Förkortning	Utläses	Betydelse
API	Application Program Interface	De funktioner och metoder en programmerare kan använda för att komma åt ett system eller objekt.
CSD	Circuit Switched Data	Kretskopplad data till skillnad mot paketbaserad data. En fast ledning kopplas upp mellan två parter.
DTMF	Dual Tone Multi Frequency	De signaler som en tonvalstelefon eller mobiltelefon skickar när användaren trycker på knapparna.
GPRS	General Packet Radio Service	Efterföljare till GSM som har möjlighet att använda paketbaserad överföring.
GSM	Ursprungligen från arbetsgruppen Groupe Spéciale Mobile, men numera Global System for Mobile communication	Digital mobiltelefonstandard som används i större delen av världen utanför USA och Japan.
HTML	HyperText Markup Language	Språk för att beskriva den logiska och layoutmässiga strukturen i ett dokument.
HTTP	HyperText Transfer Protocol	Den metod som används för att föra över filer mellan två datorer på WWW. Ursprungligen utvecklat för HTML.
IETF	Internet Engineering Task Force	En organisation som arbetar med utvecklingen av Internet. Det är de som fastställer så kallade Internetstandarder.
IP	Internet Protocol	Det grundläggande protokoll som används på Internet. Det är ansvarigt för att transportera enkla paket av data mellan två datorer.
JDBC	Java Database Connectivity	Ett gränssnitt som kan användas för att kommunicera med databaser från Java.
MSISDN	Mobile Subscriber ISDN number	Det nummer som är unikt för alla användare av GSM. Det består av telefonnumret med vissa landskoder framför och kan vara upp till 15 siffror långt.
ODBC	Open Database Connectivity	Ett gränssnitt för att kommunicera med databaser som Microsoft utvecklat men som finns till de flesta databaser och datorsystem.
PPP	The Point-to-Point Protocol	PPP är en metod att för kapsla in data som hör till ett nät i en förbindelse mellan två datorer. Typiskt används det för att köra TCP/IP över en modemförbindelse för att kunna använda Internet.

RADIUS	Remote Authentication Dial In User Service	En standardiserad metod för att hantera behörighetsinformation hos stora mängder uppringande användare hos till exempel en Internetleverantör. (RFC 2138 1997)
RFC	Request For Comments	Ett dokument fastställt av IETF. Det kan vara rekommendationer, information eller standarddokument. Såväl utkast, föreslagna som fastställda standarder förekommer bland standarddokumenten.
SIM	Subscriber Identity Module	Det personliga kort som hör till ett mobiltelefonabonnemang och som sätts in i mobiltelefonen.
SMS	Short Message Service	En tjänst som gör det möjligt att skicka korta meddelanden till och från mobiltelefoner.
SMSC	Short Message Service Centre	Den dator hos operatören som tar emot och levererar SMS.
SQL	Structured Query Language	Ett standardiserat frågespråk som används för att hämta och ändra data i en databas.
TCP	Transmission Control Protocol	Ett protokoll som ger två datorer en till synes fast förbindelse med varandra trots att trafiken går i små IP-paket (RFC 793 1981).
TLS	Transport Layer Security	En version av det säkerhetssystem som används på webben.
UDP	User Datagram Protocol	Ett protokoll som låter en dator skicka ett datapakete utan någon kontroll om det kommer fram eller när det i så fall kommer fram. (RFC 768 1980)
UMTS	Universal Mobile Telecommunications System	Tredje generationens mobiltelefonsystem som med avsikt att lanseras inom 2-3 år som en påbyggnad och ersättare till GSM. (Att förstå telekommunikation 2 1998)
URL	Uniform Resource Locator	En "adress" till ett objekt, till exempel en HTML-sida på en annan dator.
W3C	World Wide Web Consortium	Den organisation som fastställer standarder för webben. Det är W3C som fastslagit HTML-standarderna.
WAP	Wireless Application Protocol	Det system som ska göra Internet tillgängligt från mobila enheter, till exempel mobiltelefoner.
WML	Wireless Markup Language	Ett uppmärkningspråk liknande HTML men baserat på XML som används för att visa information på till exempel mobiltelefon-skärmar i WAP.
WTA	Wireless Telephony Application	Ett system som kombinerar WAP:s datadelar med mobiltelefonens telefondelar.

WTAI	Wireless Telephony Application Interface	Det API som ingår i WTA.
WWW	World Wide Web	Det nät av hoplänkade hemsidor bestående av HTML-sidor och motsvarande som finns på Internet
XML	eXtended Markup Language	Ett väldefinierat språk som det är möjligt att bygga uppmärkningspråk på. Alla språk baserade på XML har samma struktur (XML 1998)

Tabell 3. Förkortningar

8 Referenser

Att förstå telekommunikation 2 (1998), Studentlitteratur, Lund. ISBN 91-44-37811-4

Biryukov, Alex & Shamir, Adi (1999) *Real Time Cryptanalysis of the Alleged A5/1 on a PC* <http://cryptome.org/a5.ps> (Acc. 1999-12-15)

Blum, Bruce I. (1992) *Software Engineering, A Holistic View* Oxford University Press, New York ISBN: 0-19-507159-X

Boehm, B. W. (1988) *A Spiral Model of Software Development and Enhancement* Computer 21,5 sid 61-72

Caching (1999) *Wireless Application Protocol, Cache Model Specification* <http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WAPCachingMod-19990211.pdf> (Acc. 1999-12-16)

ECMA-262 (1998) *ECMAScript Language Specification* <http://www.ecma.ch/stand/ecma-262.htm> (Acc. 1999-12-09)

Eriksson, Hans-Erik & Penker, Magnus (1998) *UML Toolkit* John Wiley & Sons Inc. New York ISBN: 0-471-19161-2

Miller, G.A. (1956) *The magical number seven, plus or minus two: Some limits on our capacity for processing information* Psychological Review 63 sid. 81-97

NAMP (1999), *Nokia Artus Messaging Platform* <http://www.nokia.com/networks/17/namp.html> (Acc. 1999-12-16)

Push (1999) *WAP Push Architectural Overview* <http://www1.wapforum.org/tech/terms.asp?doc=PROP-PushArchOver-19991108.pdf> (Acc. 1999-12-15)

RFC 768 (1980) *User Datagram Protocol* <http://www.cis.ohio-state.edu/htbin/rfc/rfc0768.html> (Acc. 1999-12-09)

RFC 793 (1981) *Transmission Control Protocol, DARPA Internet Program Protocol Specification* <http://www.cis.ohio-state.edu/htbin/rfc/rfc0793.html> (Acc. 1999-12-09)

RFC 1661 (1994) *The Point-to-Point Protocol (PPP)*
<http://www.cis.ohio-state.edu/htbin/rfc/rfc1661.html> (Acc. 1999-12-09)

RFC 2138 (1997) *Remote Authentication Dial In User Service (RADIUS)*
<http://www.cis.ohio-state.edu/htbin/rfc/rfc2138.html> (Acc. 1999-12-09)

RFC 2246 (1999) *The TLS Protocol Version 1.0*
<http://www.cis.ohio-state.edu/htbin/rfc/rfc2246.html> (Acc. 1999-12-09)

RFC 2616 (1999) *Hypertext Transfer Protocol -- HTTP/1.1*
<http://www.cis.ohio-state.edu/htbin/rfc/rfc2616.html> (Acc. 1999-12-09)

Shannon, C.E. (1951) *Prediction and Entropy in Printed English* Bell System Technical Journal v. 30 n. 1, 1951 sid. 50-64

Sommerville, Ian (1989) *Software engineering* 3:e upplagan, Addison-Wesley Publishers Ltd, Wokingham, ISBN 0-201-17568-1

van Vliet, Hans, Van Vliet Vrije Van Vliet, J.C. (1993) *Software Engineering Principles and Practice*, John Wiley & Sons Ltd, Chicester. ISBN 0-471-93611-1

WAP White Paper (1999) *WAP White Paper*
<http://www.wapforum.org/what/whitepapers.htm> (Acc. 1999-12-11)

W3C (1999) *World Wide Web Consortium and Wireless Application Protocol Forum Establish Formal Liaison Relationship*
<http://www.w3.org/1999/12/W3C-WAP-CG-pr.html.en> (Acc. 1999-12-10)

WAP Architecture (1998), *Wireless Application Protocol, Architecture Specification*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WAPArch-19980430.pdf> (Acc. 1999-12-09)

WAE 1.1 (1999) *Wireless Application Protocol, Wireless Application Environment Specification, Version 1.1*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WAESpec-19990524.pdf> (Acc. 1999-12-09)

WBXML 1.1 (1999), *Wireless Application Protocol, Binary XML Content Format Specification, Version 1.1*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WBXML-19990616.pdf> (Acc. 1999-12-09)

WDP 1.1 1999, *Wireless Application Protocol, Wireless Datagram Protocol Specification*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WDP-19990514.pdf> (Acc. 1999-12-09)

WML 1.1 (1999), *Wireless Application Protocol, Wireless Markup Language Specification, Version 1.1*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WML-19990616.pdf> (Acc. 1999-12-09)

WMLScript 1.1 (1999), *Wireless Application Protocol, WMLScript Language Specification, Version 1.1*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WMLScript-19990617.pdf> (Acc. 1999-12-09)

WSP 1.1 (1999) *Wireless Application Protocol, Wireless Session Protocol Specification*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WSP-19990528.pdf> (Acc. 1999-12-09)

WTA (1999), *Wireless Application Protocol, Wireless Telephony Application Specification* (förslag till WAP 1.2-standard)
<http://www1.wapforum.org/tech/terms.asp?doc=PROP-WTA-19991108.pdf> (Acc. 1999-12-10)

WTLS 1.1 (1999), *Wireless Application Protocol, Wireless Transport Layer Security Specification*
<http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WTLS-19990211.pdf> (Acc. 1999-12-09)

WTP 1.1 (1999), *Wireless Application Protocol, Wireless Transaction Protocol Specification* <http://www1.wapforum.org/tech/terms.asp?doc=SPEC-WTP-19990611.pdf> (Acc. 1999-12-09)

Kannel (2000) www.kannel.org <http://www.kannel.org/> (Acc. 2000-01-21)

XML (1999) *Extensible Markup Language (XML)* <http://www.w3.org/XML/> (Acc. 1999-12-09)

9 Index

A

access-server, 22
aktivitet, 15
användargränssnitt, 51

B

bärare, 18, 37
 bandbredd, 19
 fördröjning, 19
Boehms spiralmodell, 8
bytekod, 36
bytekodning, 14

C

cachning, 40

D

databas, 31, 33
DTMF-signaler, 45

E

ECMAScript, 16
Ericsson, 13
Ericsson MC218, 23
Ericsson R320s, 25
Ericsson R380s, 25
Ericsson WAP IDE, 27, 52

F

förkortningar, 57

G

gateway, 22, 28
 Kannel, 29
GPRS, 19, 48
GSM-CSD, 18, 37, 38
GSM-USSD, 18

H

hälsningsmeddelande, 5
inspelning, 39

I

identifiering, 40
inkrementell prototypning, 8
innehållsförteckning, v
Internet, 22, 49
IP, 19
IS-136, 18

J

JavaScript, 16

K

källor, 11
Kannel, 29
kortlek, 14, 15
 storleksbegränsning, 51
kryptering, 20

M

meddelandeavvisering, 39, 47
meddelandeavlyssning, 39, 44
meddelandelistning, 38
metod, 7
Miller, 4
MobilSvar, 31
Motorola, 13

N

Nokia, 13
Nokia 7110, 10, 24, 38, 40, 51
Nokia SDK, 26, 51
Nokia Toolkit, 27, 51
Nokia WAP Server, 40
nummerpresentation, 40

O

ordförklaringar, 57

P

Palm Pilot, 26
Phone.com, 13
PPP, 37
prestanda, 38
push, 47

R

referenser, 61
röstbrevlåda, 3
 inställningar, 38
 WAP 1.1, 34
 WAP 1.2, 42

S

säkerhet, 41
Shannon, 14
Siemens S25, 24
simulator, 26
SMS, 18, 37, 38, 39
SMSC, 37
SSL, 20

T

Telia, 31
TSL, 20

U

UMTS, 19, 48
Unwired Planet, 13
UP.Simulator, 28, 52

V

vattenfallsmodellen, 7
 iterativa, 7

W

W3C, 15, 49
WAE, 21
WAP, 13
 bärare, 18
 gateway, 22, 28
 historik, 13
 infrastruktur, 21
 klient, 22
 protokollstack, 18
 simulator, 26
 syfte, 13
WAP Forum, 13
WBXML, 14
WDP, 19
WML, 15
 exempel, 15
 läsare, 51
WMLScript, 16
 exempel, 17
WSP, 20
WTA, 44
 server, 45
WTAI, 44
WTLS, 20
WTP, 20

X

XML, 15

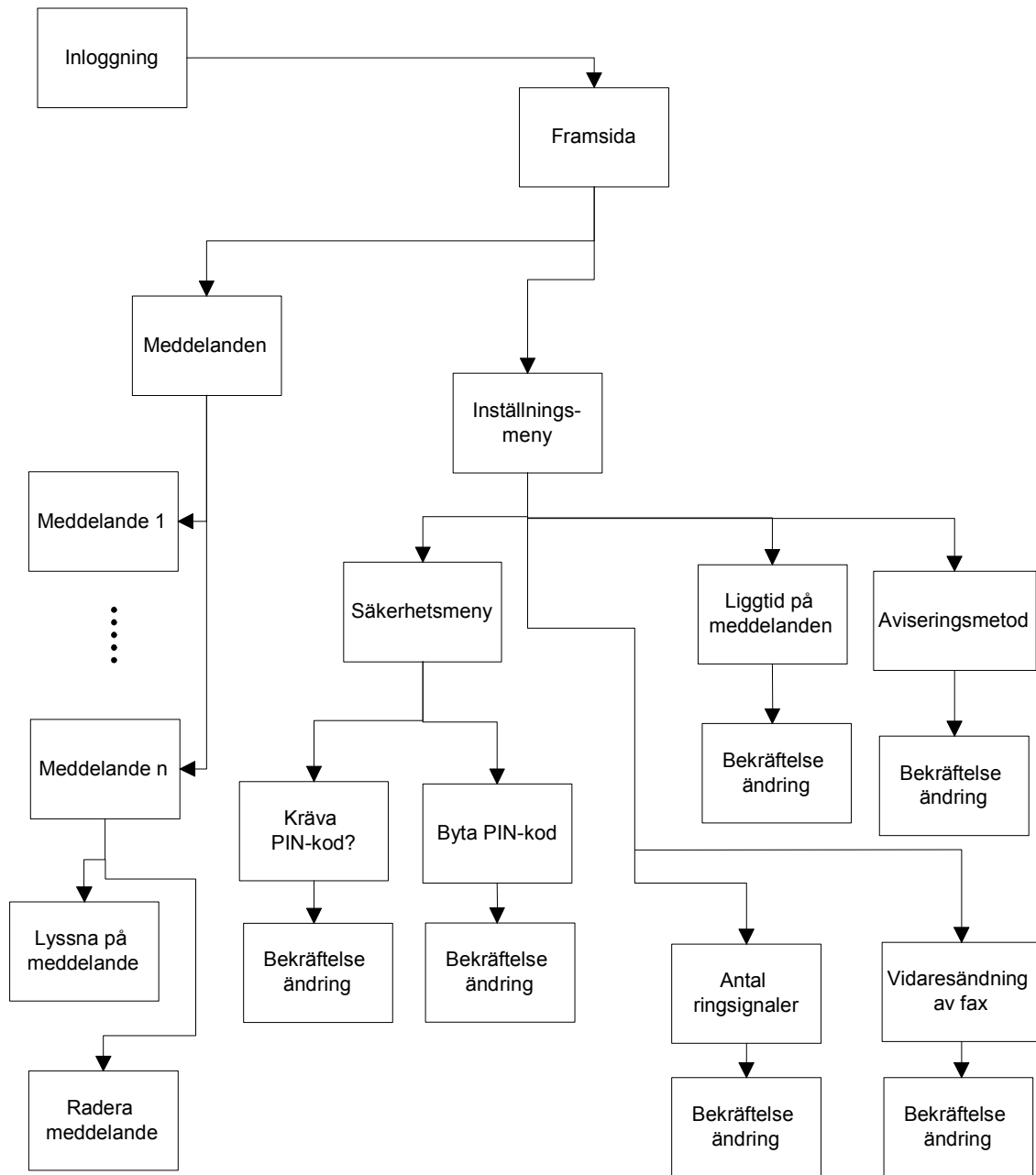
10 Bilaga A – prototypen

Det som till slut blev ett demoprogram körde på en Nokia WAP Server Beta 3. Programmet bestod av ett antal servlets och några fixa filer. En servlet är ett Java-program som är specialanpassat för att generera HTML-, eller, som i det här fallet, WML-sidor. Typiskt initieras servleten när WWW-servern startar och sedan anropas en metod i servleten när det kommer ett besök till en adress som servleten ansvarar för.

Servlet	Beskrivning
Login	Hanterar inloggningar och autenticierar de som ger rätt information.
Woris	Den servlet som listar och visar de olika meddelandena.
Options	Inställningsservleten. Visar menyer och låter användaren pilla på inställningar.
Options2	Ännu fler inställningar. Om de var i Options skulle den WML-sidan bli för stor.
Delete	Raderar ett meddelande.
Listen	Lyssna av ett meddelande. Just nu läsmarkeras meddelandet och det visas ett meddelande om att det inte går att höra det.
*change	Servlet vars namn slutar på change är ansvariga för att ändra någon inställning.

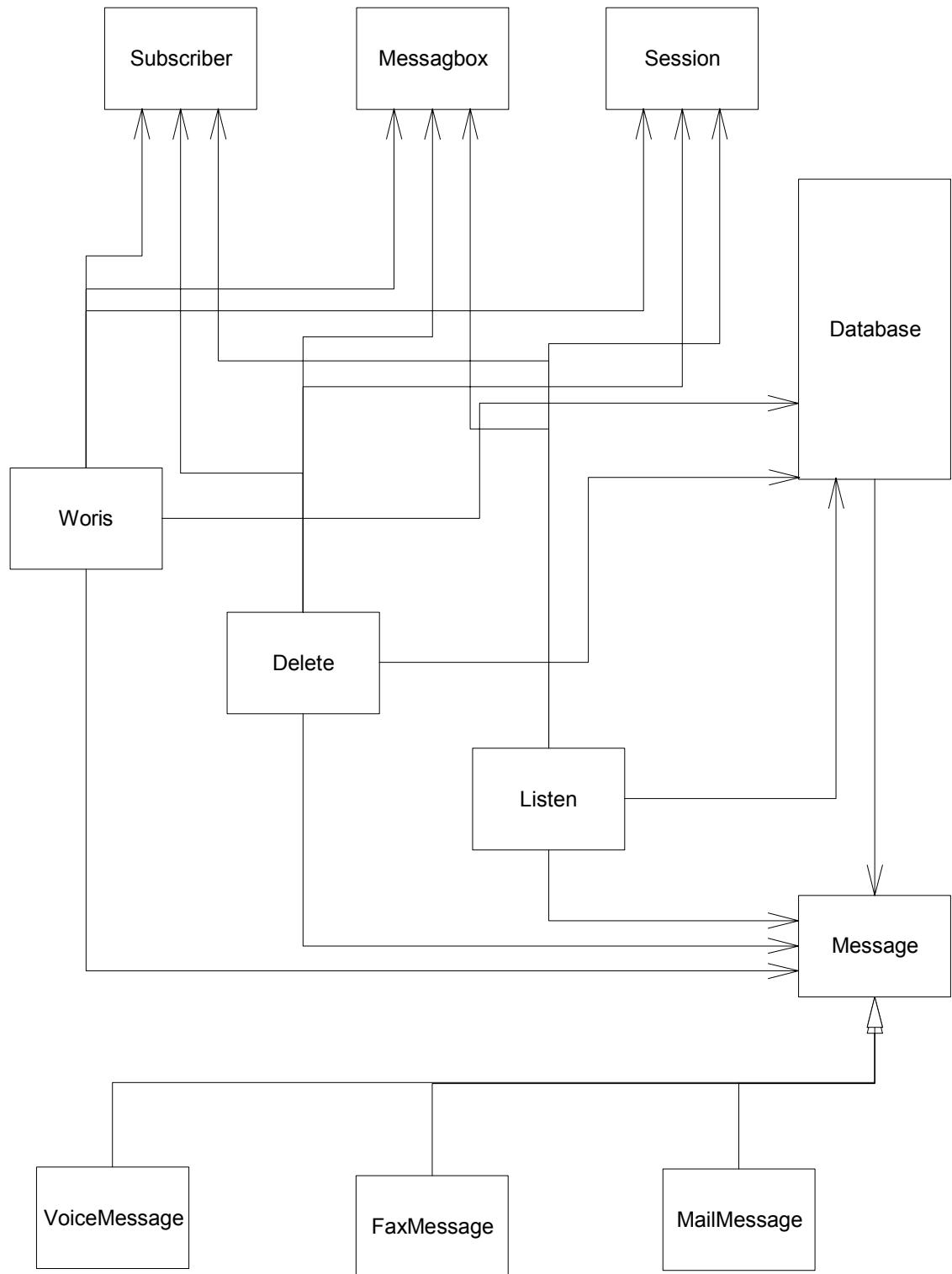
Det finns också några statiska sidor, bland annat en som innehåller WMLScript som används av de WML-sidor som servletarna genererar.

Eftersom varje servlet är en egen enhet så finns det ingen fysisk anropsstruktur mellan dem förutom de URL:er som finns inne i den WML-kod servleten genererar, men det finns en speciell ordning som det är meningen att de ska användas i. Nedan finns en figur som visar de olika sidor som finns och hur de länkar till varandra.

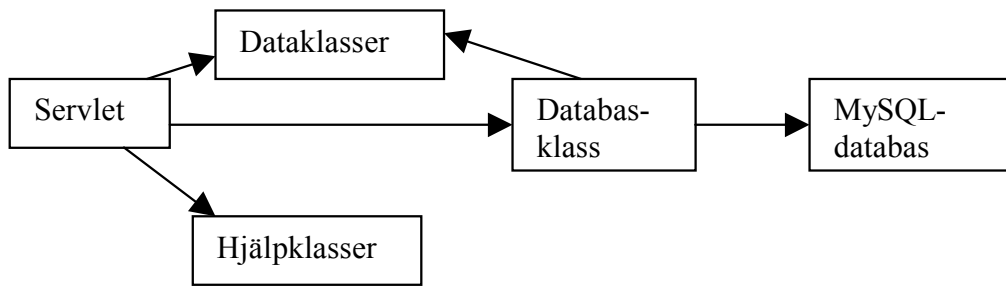


Figur 27. Sidstruktur i prototypen

Ett klassdiagram i UML ser ut som i nedanstående figur.



Figur 28. Klassdiagram för prototypen



Figur 29. "Använder"-strukturen för en Servlet i prototypen

11 Bilaga B

Den här bilagan listar de punkter som sattes upp innan arbetet började och hur testen för de olika testen utföll.

Ordförklaring för resultat:

FUNGERAR	Testet uppfylldes.
FUNGERAR INTE	Testet försökte uppfyllas men det misslyckades på ett sätt som var icke-trivialt att lösa.
EJ TESTBAR	Punkten var inte objektivt testbar.
EJ TESTAD	Punkten testades inte

Punkt	Resultat	Kommentar
Ta emot uppkopplingar från en WAP-telefon.	FUNGERAR	Inga problem.
Meddela person med WAP-telefon att det kommit meddelanden då personen kopplar upp mot tjänsten.	FUNGERAR	Inga problem.
Meddela person med WAP-telefon att det kommit meddelanden då personen inte är uppkopplad mot tjänsten.	FUNGERAR INTE	Det saknas push-funktionalitet i WAP 1.1.
Lista tillgängliga meddelanden på skärmen.	FUNGERAR	Inga problem.
Låta person med WAP-telefon lyssna av meddelanden valda från en meny på skärmen.	FUNGERAR INTE	Saknas koppling mellan tal och data i WAP 1.1.
Navigera mellan meddelanden.	FUNGERAR	Inga problem.
Radera meddelande från en meny på skärmen.	FUNGERAR	Inga problem.

Göra de inställningar användaren kan göra idag i ett telesvarssystem.	FUNGERAR	Inga problem.
Presentera omfattande hjälp när det kan behövas.	FUNGERAR	Kräver att mer data hämtas till telefonen men skapar inga egentliga problem.
Tjänsten ska kunna hållas i synk med en extern databas med lämpligt API.	FUNGERAR	Inga problem.
Klara flera användare samtidigt.	EJ TESTAD	Det beror till en stor del på hur det underliggande röstbrevlåde-systemet ser ut.
Loggning av information som önskas sparas.	FUNGERAR	Inga problem.
Tillförlitlighet.	EJ TESTAD	Det fanns många buggar i de program som användes så användningen av prototypen hade en del svårigheter. Det var dock inga problem med serverprogramvaran som inte krånglade fast den gick veckor i sträck.
Att lösningen blir resurssnål	EJ TESTBAR	Prototypen och serverprogrammen var skrivna i Java och använde mycket minne.
Tillåta och klara av andra meddelandetyper än röstmeddelanden, till exempel fax eller SMS.	EJ TESTAD	Det bör inte vara något problem
Säker mot "elaka" användare.	EJ TESTBAR	Det blir i stort sett samma säkerhetskrav som en tjänst på Internet.
Ha ett lättanvänt användargränssnitt.	EJ TESTAD	Problemen med att anpassa WML-koden så att den fungerade bra på Nokia 7110 gjorde att användargränssnittet inte blev testat.
Kunna demonstreras på en WAP-telefon.	FUNGERAR	Fungerade bra på en Nokia 7110.
Välstrukturerade designlösningar.	EJ TESTBAR	Prototypen växte fram steg för steg, så designen blev som den blev.
Den får gärna vara plattformsoberoende för att förenkla byte av plattform.	EJ TESTBAR	Nu är prototypen skriven i Java vilket var vad Nokia WAP Server ville ha.
Den ska vara miljövänlig.	EJ TESTBAR	Jag hoppas det.

Testerna som sattes upp var följande:

Ta emot uppkopplingar från en WAP-telefon.

Det ska anses verifierat när en person kan få upp en WML-sida som kommer från en egen server på en telefon.

Meddela person med WAP-telefon att det kommit meddelanden då personen kopplar upp mot tjänsten.

Det ska anses verifierat när en person med en WAP-telefon eller lämplig simulator kan få upp ett meddelande som är beroende på antalet meddelanden i meddelandelådan.

Meddela person med WAP-telefon att det kommit meddelanden då personen inte är uppkopplad mot tjänsten.

Det ska anses verifierat när en person med en WAP-telefon eller lämplig simulator kan få upp ett meddelande utan att vara uppkopplad mot tjänsten.

Lista tillgängliga meddelanden på skärmen.

Det ska anses verifierat när en person med en WAP-telefon eller lämplig simulator kan få upp en lista av de olika meddelande som finns i meddelandelådan. Det ska tydligt framgå vilka meddelanden som är nya och vilka som är "lästa".

Låta person med WAP-telefon lyssna av meddelanden valda från en meny på skärmen.

Det ska anses verifierat när en person med WAP-telefon eller lämplig simulator kan lyssna av ett meddelande valt från den lista som nämns i föregående test.

Navigera mellan meddelanden.

Det ska anses verifierat när en person med WAP-telefon eller lämplig simulator kan markera och få upp information om olika meddelanden. Det ska vara möjligt att på ett enkelt sätt gå fram och tillbaka mellan meddelanden.

Radera meddelande från meny på skärmen.

Det ska anses verifierat om en person med WAP-telefon eller lämplig simulator kan utföra en operation som gör att ett meddelande inte dyker upp på några listor nästa gång personen använder tjänsten. Direkt efter radera-operationen räcker det om meddelandet på ett tydligt sätt är markerat för radering. I så fall ska det också vara möjligt att ta bort raderamarkeringen i vilket fall meddelandet ska vara kvar nästa gång personen använder tjänsten.

Göra de inställningar användaren kan göra idag i ett telesvarssystem.

Det ska anses verifierat när en person med WAP-telefon eller lämplig simulator kan byta PIN-kod.

Presentera omfattande hjälp när det kan behövas.

Det ska anses verifierat när en person med WAP-telefon eller lämplig simulator kan få upp mer information om en operation och sedan enkelt kan komma tillbaka till den situation personen var i innan.

Tjänsten ska kunna hållas i synk med en extern databas med lämpligt API.

Det ska anses verifierat när en person med WAP-telefon eller lämplig simulator kan göra ändringar på sin telefon eller i simulatorm som sedan avspeglas i en meddelandedatabas som ligger externt från serverprogramvaran eller i en modul separerad från den övriga servern med ett tydligt och väl definierat API.

Klara flera användare samtidigt.

Det ska anses vara verifierat när två personer med WAP-telefoner eller lämpliga simulatorer kan utföra operationer mot sina respektive meddelandelådor på ett sätt som för användarna upplevs som samtidigt.

Loggning av information som önskas sparas.

Det ska anses verifierat när händelserna inloggning och radering av meddelande finns loggade i någon form på servern.

Tillförlitlighet.

Det ska anses verifierat när en server med programvara kan köra flera dagar med många (>100 i testfallet) operationer utan att servern kraschar, ökar sitt minnesbehov annat än temporärt eller på annat sätt uppvisar beteenden som kan tyda på att längre drift skulle resultera i problem.

Att lösningen blir resurssnål

Det är ett så kallat ickefunktionellt krav och kommer inte att verifieras med några testfall.

Tillåta och klara av andra meddelandetyper än röst-meddelanden, till exempel fax eller SMS.

Det ska anses verifierat när meddelanden av andra typer än röstmeddelanden presenteras på WAP-telefonen eller simulatorm och det är möjligt att se information om dessa meddelanden, enkelt avgöra deras typ (fax, SMS, m.m.) och radera dem.

Säker mot ”elaka” användare.

Det är till viss del ett så kallat ickefunktionellt krav och kommer inte att verifieras fullständigt, men åtminstone ska systemet klara av att någon med insikt i hur systemet fungerar försöker störa dess funktion. Med klara av innebär att inga andra användare ska påverkas.

Ha ett lättanvänt användargränssnitt.

Det är ett så kallat ickefunktionellt krav och kommer inte att verifieras med några testfall.

Kunna demonstreras på en WAP-telefon.

Det ska anses verifierat när en person med en WAP-telefon kan stå utan fysisk eller IR-kontakt med någon annan maskin och använda systemet därifrån.

Välstrukturerade designlösningar.

Det är ett så kallat ickefunktionellt krav och kommer inte att verifieras med några testfall.

Den får gärna ha vara plattformsoberoende för att förenkla byte av plattform.

Det är ett så kallat ickefunktionellt krav och kommer inte att verifieras med några testfall.

Den ska vara miljövänlig.

Det är ett så kallat ickefunktionellt krav och kommer inte att verifieras med några testfall.



LINKÖPINGS UNIVERSITET

Avdelning, institution
Division, department

Institutionen för datavetenskap

Department of Computer
and Information Science

Datum
Date

2000-02-21

Språk
Language

Svenska/Swedish
 Engelska/English

Rapporttyp
Report category

Licentiatavhandling
 Examensarbete
 C-uppsats
 D-uppsats
 Övrig rapport

ISBN —

ISRN —

Serietitel och serienummer
Title of series, numbering

ISSN —

LiTH-IDA-Ex- 00/20

URL för elektronisk version

Titel
Title

WAP och röstbrevlådetjänster

Författare
Author

Daniel Bratell

Sammanfattning
Abstract

WAP, Wireless Application Protocol, är en metod för att koppla ihop datornätverk och mobiltelefoni. Tanken bakom den ihopkopplingen är att kunna erbjuda användare av mobiltelefoner de tjänster som redan finns på Internet direkt in i mobiltelefonerna. Den fråga som E pact Technology AB ville ha svar på var om det är möjligt att använda WAP till att förbättra röstbrevlådar såsom till exempel Telias MobilSvar.

Frågan om huruvida det är möjligt att använda WAP till en röstbrevlådetjänst visade sig kunna besvaras med både ja och nej beroende på tidsperspektivet och vilka krav som ställs på tjänsten. I den version av standarden som varit den aktuella under examensarbetet, WAP 1.1, saknas en del nyckelkomponenter för att möjliggöra växling mellan data och tal. Det innebär att en användare inte kan lyssna av meddelanden eller spela in hälsningsmeddelanden på något enkelt sätt och saknas de funktionerna är det inte heller möjligt att tillverka en WAP-röstbrevlåda. Det som kan byggas av de tillgängliga funktionerna är ett administrations- och övervakningsgränssnitt till en röstbrevlåda så att användarna kan göra inställningar och se vilka meddelanden som finns i röstbrevlådan.

Under arbetet med den här rapporten pågick utarbetandet av en ny version av standarden som skulle bli WAP 1.2. Det har funnits ett förslag till standard, vilket senare blev det som fastslogs, som jag har studerat under det här examensarbetet. I WAP 1.2 ingår de nyckeldelar som saknades i WAP 1.1 som bland annat gör det möjligt att komma åt telefonifunktioner från en WAP-sida. Med WAP 1.2 ska det i teorin vara fullt möjligt att bygga en tjänst som har samma funktionalitet som de traditionella röstbrevlådorna har idag, men först krävs telefoner och serversystem som har stöd för den nya versionen av standarden. Möjligtvis kommer vissa prestandakrav göra det intressant att vänta på GPRS som är den utvecklade formen av GSM som förväntas lanseras under år 2000 eller 2001.

Nyckelord
Keywords

WAP, mobiltelefoni, röstbrevlåda, GPRS