

Synkronisering med SyncML

Arash Afsarinejad

LITH-ISY-EX-3189-2002

Titel

Synkronisering med SyncML

Examensarbete utfört i Kommunikationssystem
vid Linköpings tekniska högskola

av

Arash Afsarinejad

LiTH-ISY-EX-3189-2002

Handledare: Niclas Persson

Examinator: Fredrik Gunnarsson

Linköping: 2002-04-18

**Avdelning, Institution**

Division, Department

Institutionen för Systemteknik
581 83 LINKÖPING**Datum**

Date

2002-04-23

Språk

Language

 Svenska/Swedish
 Engelska/English**Rapporttyp**

Report category

 Licentiatavhandling
 Examensarbete
 C-uppsats
 D-uppsatsÖvrig rapport
—**ISBN****ISRN** LITH-ISY-EX-3189-2002**Serietitel och serienummer**

Title of series, numbering

ISSN**URL för elektronisk version**<http://www.ep.liu.se/exjobb/isy/2002/3189/>**Titel**

Synkronisering med SyncML

Title

Synchronisation with SyncML

Författare

Arash Afsarinejad

Author

Sammanfattning

Abstract

The last couple of years the use of mobile devices such as mobile phones and PDAs has increased tremendously. Most of these devices have their own protocols for synchronising data and this has given rise to a need for a standard synchronisation protocol, SyncML. This thesis compares this protocol against the existing ones. The comparison shows that the preferred choice is SyncML.

Also an application using SyncML has been developed. The application's task is to synchronise the calendar on a mobile phone with a database on a computer.

Nyckelord

Keyword

Synkronisering, SyncML, IntelliSync, ActiveSync, ScoutSync, uSync

Sammanfattning

Detta examensarbete behandlar synkronisering av data mellan en mobiltelefon och en dator. Detta görs genom att utreda vilka olika synkroniseringsprotokoll som används idag samt hur dessa uppfyller de krav och kriterier som ställs på dagens synkroniseringsprotokoll. Fyra olika protokoll jämfördes och av dem väljs ett protokoll, SyncML, som uppfyller de uppsatta kriterierna på bästa sätt.

Vidare byggdes den systemlösning som krävs för att utveckla en applikation som med hjälp av SyncML ska synkronisera data via en extern server, uSync servern. Applikationen har till uppgift att synkronisera kalendern på en mobiltelefon med en databas på dator. Mellan mobiltelefon och dator finns det en uSync server som används för att genomföra synkroniseringen.

Förord

Det är många jag skulle vilja tacka för deras hjälp och stöd för fullbordandet av detta arbete.

Framförallt vill jag tacka alla på TietoEnator som har ställt upp och hjälpt mig då jag har stött på svårigheter. Jag skulle vilja tacka min handledare Jonas Brodd. I synnerlighet vill jag rikta ett stort tack till Lars Blomberg och Per Borg som båda två alltid har tagit sig tid att hjälpa mig. Ett tack går också till Jonas Brandt på Ericsson som bibestod mig med uSync servern så jag skulle kunna testa applikationen.

Jag vill också tacka min handledare Niclas Persson och min examinator Fredrik Gunnarsson för deras hjälp och stöd.

Sist men inte minst vill jag tacka mina föräldrar och vänner som har varit ett orubbligt stöd genom hela detta arbete.

Arash Afsarinejad
Linköpings Tekniska Högskola
April 2002

Innehållsförteckning

1 INLEDNING	1
1.1 BAKGRUND	1
1.2 TIETOENATOR CARLSYS AB	2
1.3 SYFTE	2
1.4 UPPDRAGSBESKRIVNING	2
1.5 AVGRÄNSNINGAR	3
2 SYNKRONISERINGSBEGREPPET	4
2.1 SYNKRONISERINGSSCENARIOS	4
2.2 VARFÖR SYNKRONISERING	4
2.3 DAGENS SYNKRONISERINGSPROBLEM	8
2.4 ÜRSKILJNINGSKRITERIER	8
2.4.1 INNEHÅLLSKÄNNEDOM	9
2.4.2 SIMULTAN SYNKRONISERING	9
2.4.3 DIREKT SYNKRONISERING	10
2.4.4 PLATTFORMSOBEROENDE	10
2.4.5 EFFEKTIVITETSKRAV	10
2.4.6 ÖNSKVÄRDA EGENSKAPER HOS SYNKRONISERINGSPROTOKOLLET	11
2.5 UTVÄRDERING AV SYNKRONISERINGSSALTERNATIV	11
2.5.1 INTELLISYNC	12
2.5.2 ACTIVESYNC	12
2.5.3 SCOUTSYNC	13
2.5.4 SYNCML	14
2.6 MOTIVERING AV VAL	15
2.6.1 VAL AV SYNKRONISERINGSPROTOKOLL	15
3 SYSTEMBESKRIVNING	17
3.1 SYSTEMLÖSNING	17
3.2 USYNC SERVERN	18
3.3 SOAP	19
3.4 TOMCAT 4.0	20
3.5 APPLIKATIONEN	21
3.6 JDBC	22
3.7 MYSQL	23
4 RESULTAT AV APPLIKATIONSUTVECKLING	25
5 SLUTSATS OCH FRAMTIDSUTSIKTER	26
6 REFERENSER	27

1 Inledning

I detta inledande kapitel presenteras såväl en bakgrund till ämnesområdet som företaget. Sedan följer en problemdiskussion vilken sedermera leder fram till studiens syfte. Därefter följer en mer detaljerad uppdragsbeskrivning och slutligen tas de avgränsningar som gjorts i arbetet upp.

1.1 Bakgrund

Under de senaste åren har en tydlig förändring ägt rum i flera företag och organisationer. Allt eftersom mobila enheter blivit ett vanligare inslag i arbetet har en tydlig förändring mot en mobil uppbyggnad ägt rum. Det är inte bara datorer som har blivit mer mobila utan en helt ny genre av mobila enheter såsom mobiltelefoner och handdatorer har vuxit fram.

Mobila enheter har således öppnat många nya möjligheter i dagens arbetsituation, men som med all teknik finns det även svårigheter. Svårigheten i detta fall uppstår främst vid kopiering och uppdatering av data från en enhet till en annan. För att lösa detta problem har ett antal olika synkroniseringsteknologier tagits fram. För att undvika de hinder som uppstår med olika synkroniseringsprotokoll har en standard, SyncML, tagits fram av ett antal företag. Synkroniseringen har till uppgift att se till så att båda enheterna innehåller den uppdaterade data och att den är identisk på båda enheterna.

1.2 TietoEnator CarlSys AB

TietoEnator CarlSys AB ingår i TietoEnator under verksamhetsområdet TietoEnator Technical Consultants (Tech) och är engagerade i produktutveckling för högteknologiska företag. TE Technical Consultants har ca 500 medarbetare fördelade på sju kontor i Sverige och fyra i Finland. Inom Tech finns ett antal fristående bolag som arbetar självständigt men samtidigt kan utnyttja fördelen med att ingå i en större organisation.

TietoEnator är den ledande leverantören av tjänster för teknisk systemutveckling till ett flertal internationellt framgångsrika företag. TietoEnator CarlSys erbjuder kvalificerade tjänster inom teknisk systemutveckling och arbetar i huvudsak mot tre branschsegment; Försvar, Tele- och Datakommunikation samt Fordon. TietoEnator CarlSys AB åtar sig projekt för att utveckla system eller delsystem, produkter eller komponenter.

1.3 Syfte

Detta examensarbete har som syfte att klargöra olika synkroniseringsprotokolls användningsområden, funktion samt för- och nackdelar. Vidare ska även en applikation som utnyttjar det valda synkroniseringsprotokoll utformas.

1.4 Uppdragsbeskrivning

Syftet ska uppnås genom ett utrednings- och utvecklingsarbete som kan indelas i tre steg:

1. Nulägesbeskrivning av vilka synkroniseringsverktyg som finns. Utvärdering ska ske enligt noggrant utvalda kriterier för att beskriva funktionalitet, för- och nackdelar med de olika synkroniseringsverktygen.
2. Bygga och utforma en systemlösning baserat på utredningsarbetet och rekommendationen som görs i delsteg 1.
3. Utveckla en applikation som på ett övertygande sätt visar fördelarna med den utvalda systemlösningen. Applikationen skall köras på operativsystem från Palm, Linux, Windows CE eller EPOC beroende på resultatet av det första utredningsarbetet. Tjänsten skall via GSM nätet koppla upp sig på Internet och kontakta servern som är implementerad på en PC.

1.5 Avgränsningar

Det finns relativt många synkroniseringsalternativ varför endast ett urval av dessa kunde studeras. De synkroniseringsalternativ som betraktas i rapporten har varit etablerade på marknaden under en längre period och är väl beprövade. Beroende på vilket synkroniseringsprotokoll som används ställs det speciella krav på den mobila enheten samt vilka enheter som är kompatibla med det valda synkroniseringsprotokollet.

En annan begränsning är att vissa av protokollen endast i dagsläget stöder synkronisering av kalender - och kontaktlistor. Detta sätter en begränsning på applikationen som endast kan utvecklas för att genomföra endera av dessa.

2 Synkroniseringsbegreppet

I detta kapitel kommer en förstudie att göras för att presentera ämnet synkronisering för att därefter kunna avgöra vilket synkroniseringsalternativ som bör användas. Vid detta val kommer ett antal relevanta urskiljningskriterier att presenteras och valet kommer att baseras på dessa.

2.1 Synkroniseringsscenarios

Ett vanligt område för synkronisering är uppdateringar av kontakt- och adresslistor. Då man vill att listan ska vara densamma i alla enheter såväl fasta som rörliga använder man sig av synkronisering. Synkroniseringen underlättar på så sätt att identiska listor fås i alla enheter och ingenting behöver införas eller redigeras manuellt. Denna synkronisering fungerar även för epost- och att-göra-listor.

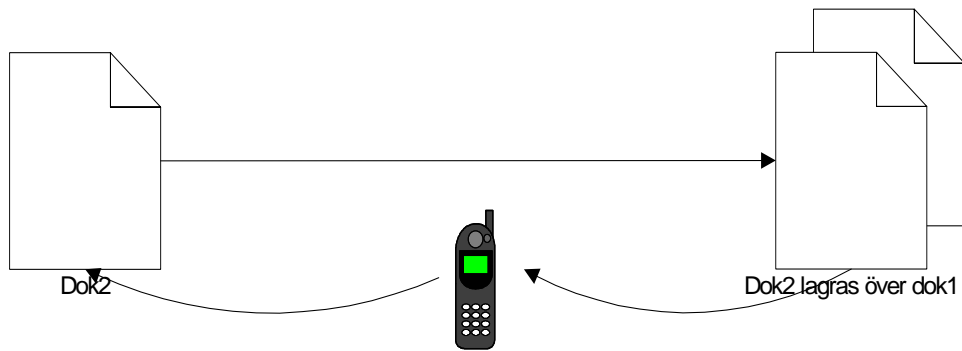
Ett annat synkroniseringsscenario är uppdateringar av databaser. I en databas kan t ex information om antal produkter som är kvar sparas. Om en säljare genomför en försäljning innebär detta att antalet produkter minskas och detta måste således uppdateras i databasen så andra säljare blir medvetna om det verkliga antalet produkter som är kvar. På detta sätt kan varje säljare synkronisera sin bärbara enhet med databasen och få tillgång till uppdaterad data.

2.2 Varför synkronisering

Synkronisering innebär ett datautbyte mellan två enheter som medför att enheterna innehåller ekvivalent information vilket fås genom att överföra varje dataelement i en enhet på det andra dataelementet i den andra enheten. Förmågan att kunna synkronisera data mellan många källor är en förutsättning för att ett nätverk ska fungera önskvärt. Detta då det i nätverket blir av större vikt att ha tillgång till uppdaterad data, vilket i sin tur innebär att synkroniseringens roll blir av en helt annan dimension. Synkroniseringen bidrar till såväl ökad effektivitet som ökad tillit. [4]

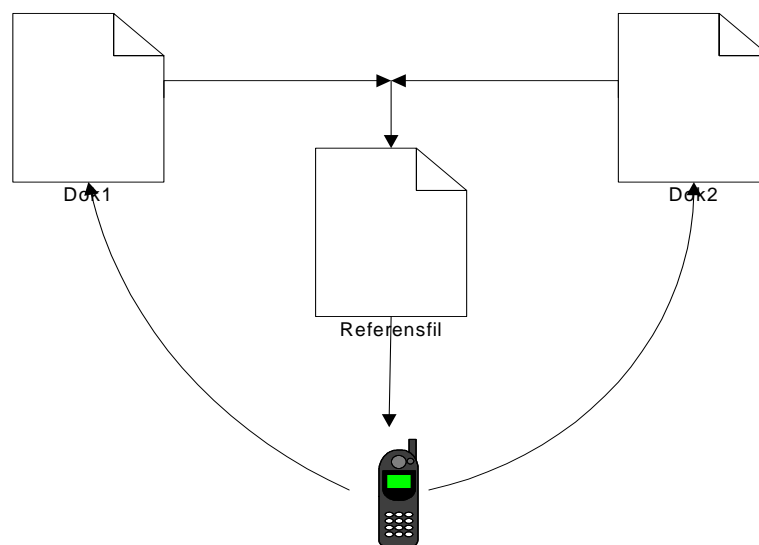
Det finns huvudsakligen tre sorters synkronisering:

- *Synkronisering av data i olika kataloger.* Det här synkroniseringsalternativet har ingen kännedom om innehållet och uppdatering sker genom att kopiera över det äldre innehållet med den nya datan. Den nya datan ersätter alltså den gamla utan att på något sätt jämföra innehållet i dessa två.



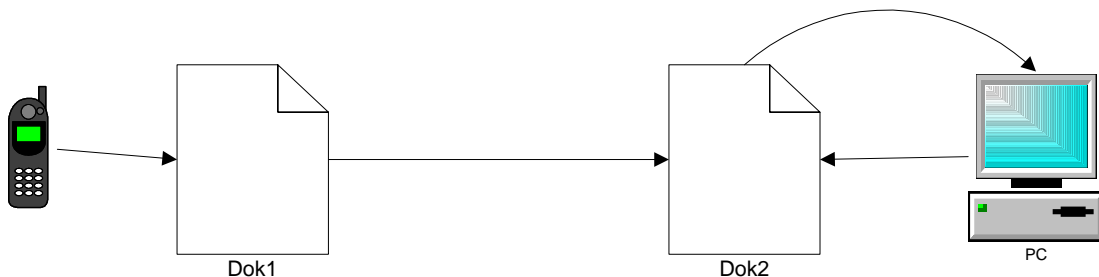
Figur 1. Synkronisering av data i olika kataloger. Den nya datan sparas på den gamla utan att innehållet jämförs innan.

- *Synkronisering av två kopior från en datakälla.* Enheterna kontrollerar en kopia i taget för att sedan producera en referensfil. All data evalueras och synkroniseras. Denna sortens synkronisering är även innehållsmedveten. Till skillnad från föregående metod tas det således hänsyn till datan i ursprungskopian.



Figur 2. Synkronisering av två kopior från en datakälla. Innehållet i de båda kopiorna beaktas och används för att skapa en referensfil som sedan sparas som den nya kopian.

- *Synkronisering av diverse enheter och databaser*, som t ex en mobiltelefon till en PC. Denna är det mest komplexa synkroniseringssättet då synkronisering måste göras över olika enheter och applikationer. För att det ska fungera önskvärt måste denna också vara innehållsmedveten. Detta för att den ska kunna jämföra och bilda en kopia som innehåller all information som finns på de två kopiorna. Det är främst denna metod som kommer att betraktas och behandlas i denna rapport då denna är mer generell och mer användbar än de övriga två.



Figur 3. Synkronisering av diverse enheter och databaser. För att denna synkronisering ska fungera måste den vara innehållsmedveten för att kunna jämföra innehållet på de två kopiorna.

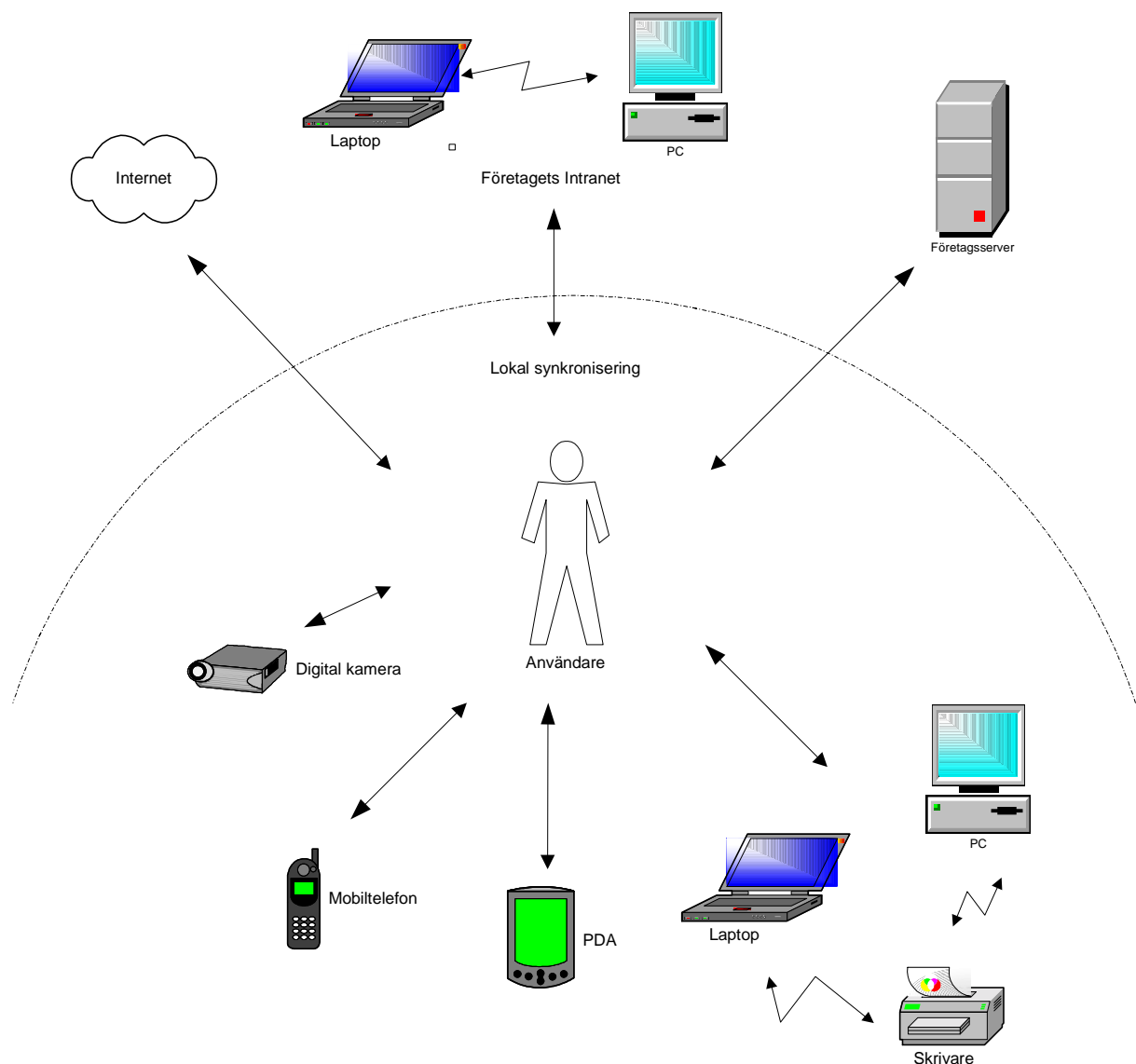
Fördelarna med mobila enheter har på sistone blivit allt mer uppenbara samtidigt som de har blivit ett vanligt inslag i det vardagliga arbetet. Att kunna få tillgång till uppdaterad information oavsett var man befinner sig har blivit allt viktigare i dagens alltmer aggressiva och konkurrerande marknad. Mobila användare är inte kontinuerligt uppkopplade mot ett nätverk utan uppkoppling sker vid de tillfällen användaren behöver få tillgång till data från ett nätverk. Denna sporadiska uppkoppling ställer höga krav på synkronisering mellan användarens mobila enhet och nätverket [2].

Användaren mottar och sparar data lokalt på den mobila enheten. Lokalt sparad data kan sedan redigeras och sparas på nytt. Data sänds sedan tillbaka till nätverket genom att användaren återigen kopplar upp sig. Användaren kan då också få reda på ändringar som skett under tiden mellan uppkopplingarna. Emellanåt kan konflikter uppstå i nätverket och processen som hanterar dessa konflikter är just datasynkronisering.

Datasynkronisering har således till uppgift att lösa dessa konflikter och se till att data är identisk på nätverket och de mobila enheterna. Ett datasynkroniseringsprotokoll preciserar

arbetsflödet för en kommunikation under en datasynkroniseringssession när den mobila enheten är uppkopplad mot nätverket. [1]

Ett tänkbart scenario där synkronisering är nödvändigt illustreras i figur 4. I figuren återspeglas flera olika enheter där synkronisering kan användas för att underlätta flödet av information mellan olika enheter. T ex kan användaren synkronisera information på sin mobiltelefon med sin PDA. Därefter ska användaren också kunna synkronisera PDA:n med sin PC hemma eller den på företagets intranet utan att några synkroniseringskonflikter ska uppstå.



Figur 4. Tänkbart synkroniseringsscenario. Alla dessa enheter ska kunna synkroniseras med varandra utan att någon hänsyn ska behövas ta till vilket synkroniseringsprotokoll som används, vilket endast är möjligt om det finns en standard.

2.3 Dagens synkroniseringsproblem

Problemet med olika synkroniseringsprotokoll är inte enbart begränsningen av användarens tillgång till data då enheterna har olika protokoll men även en hämning av tillväxten av mobila enheter samt mer utvecklade tjänster. Dagens situation med flertalet icke-kompatibla synkroniseringsprotokoll skulle kunna undvikas med ett standardprotokoll. Målet med en standard skulle vara tvåfaldigt. Det första är att nätverksdata skulle kunna synkroniseras med alla mobila enheter. Det andra är att alla mobila enheter skulle kunna synkroniseras med all nätverksdata. Detta innebär att tvåvägssynkronisering är möjlig.

Med detta som mål blir det möjligt att inhämta data från ett nätverk med alla möjliga mobila enheter under förutsättningen att båda sidor stödjer standardprotokollet. Enheterna skulle vara synkroniserbara med en större bredd av data om en standard var tillgänglig. Data synkroniseringsprotokollet skulle göra det möjligt att hämta och redigera samma data med flera olika enheter. Det skulle på så sätt möjliggöra uppdatering av en ändring till alla inblandade enheter.

Vidare är olika datastrukturer och enheter ett annat problem vid synkronisering. Olika hårdvara och enheter ger upphov till olika databasstrukturer. Detta avgör hur mycket av arbetet som läggs på synkroniseringsmjukvaran. Ett exempel som belyser detta är hur olika program registrerar, sparar och håller ordning på ändringar som görs [3]. Detta innebär att det kan finnas protokoll som innehåller olika antal fält och när de mappas fås inte en entydig avbildning, dvs. ett ensamt fält på ena sidan måste delas upp och dess uppdelade innehåll ska placeras i flera olika fält på andra sidan. [4]

2.4 Urskiljningskriterier

För att kunna urskilja vilket synkroniseringsalternativ som bör användas har ett flertal urskiljningskriterier ställts upp. Dessa användas som grund för jämförelse av de olika synkroniseringsalternativ som finns tillgängliga. De synkroniseringsalternativ som presenteras och undersöks enligt uppställda kriterier är: *IntelliSync*, *ActiveSync*, *ScoutSync* och *SyncML*. Vidare kommer även i detta kapitel en rekommendation angående vilket synkroniseringsalternativ som bör användas att ges.

2.4.1 Innehållskännedom

Innehållskännedom är förutsättning för att uppnå hög noggrannhet och tillförlitlighet i synkroniseringsförfarandet. Då synkroniseringsförfarandet har kännedom om innehållet undersöks och evalueras målfälten dvs. de fält där data ska placeras. Detta görs för att möjliggöra korrekt mappning av källfälten då olika format används och att den semantiska meningen av datan är bibehållen när datan förflyttas från källan. För att få synkronisering med innehållskännedom krävs också att följande punkter beaktas [4]:

- **Bibehållandet av data integritet under synkronisering.** Synkroniseringsprotokollet måste följa de regler som bestäms och är inbyggda i synkroniseringsenheten när den manipulerar och uppdaterar data.
- **Kunna hantera element som återkommer regelbundet,** såsom regelbundet schemalagda möten. En dator kan avgöra om en inmatning är av regelbunden karaktär och i det fallet den är det kan den reproduceras och sparas på rätt position. En handdator har begränsad kapacitet och denna regelbundenhet kan inte alltid identifieras. Detta leder till svårigheter vid synkronisering.
- **Användarberoende konflikthantering.** Kunna hantera konflikter som uppkommer då uppdateringar görs från olika enheter genom att ta hänsyn till användarens inställningar och inte förlita sig på godtyckligt och förbestämda parametrar.
- **Specifika fält för synkronisering.** Detta möjliggör för användaren att bestämma vilka fält som ska synkroniseras och samtidigt skydda data i både enheterna mot förlust.

2.4.2 Simultan synkronisering

Synkroniseringsprotokollet har till uppgift att överföra data från en enhet till en annan och se till att rätt format fås. Synkroniseringsprotokollet måste också tillåta och stödja variationer av standardprotokollet. Synkroniseringsprotokollet måste utveckla en naturlig översättningsförmåga för olika dataformat och kunna översätta data oberoende av enhet för att synkroniseringen ska kunna ske simultant. Översättningen måste även vara möjlig åt båda håll dvs. klient-server samt server-klient. [4]

2.4.3 Direkt synkronisering

En annan fundamental egenskap som krävs är direkt koppling mellan enheter och det förutsätter att synkroniseringen fungerar över både fast och trådlöst nätverk. Den måste stödja de olika transportprotokoll och media som trådlösa nätverk använder sig av t ex;

- WSP (Wireless Session Protocol), en del av WAP protokollet
- OBEX, t ex Bluetooth och IrDA

Transport över nedan nämnda punkter måste också stödjas;

- SMTP, POP3 och IMAP
- TCP/IP nätverk
- Privata trådlösa kommunikationsprotokoll

Ett effektivt synkroniseringsprotokoll måste kunna stödja transport över alla dessa nätverk. Effektiviteten ses i protokollets förmåga att minimera antalet dupliceringar vid transport. Protokollet bör i den utsträckning som är möjlig utnyttja existerande Internet- och Webblösningar som finns och byggas på dessa. [1]

2.4.4 Plattformsberoende

En väsentlig egenskap för synkroniseringsprotokollet är att det måste vara plattformsberoende för att kunna arbeta effektivt och kunna få tillgång till data från diverse enheter. Synkroniseringsprotokollet får inte ha några fastställda förutsättningar vad gäller programmeringsspråk och inte heller förvänta sig att samma språk har använts i såväl klienten som servern. [1]

2.4.5 Effektivitetskrav

För att ett protokoll skall vara effektivt måste de höga krav som ställs på den trådlösa överföringen mötas. Nätverksfördröjning, begränsad bandbredd och höga överföringskostnader är alla delar av den trådlösa nätverksuppbyggnaden och alla dessa delar ställer höga krav på synkroniseringsprotokollet. [12]

2.4.6 Önskvärda egenskaper hos synkroniseringsprotokollet

De punkter som presenteras här är egenskaper som är skulle vara underlättande och önskvärda vid synkronisering.

- Mobila enheter har begränsat minne samt en relativ liten processorkapacitet, vilket måste beaktas vid framtagandet av ett standardprotokoll. Därför bör protokollets egna plats i datautbytet var minimalt för att på så sätt spara kapacitet
- Speciella regler vid dataöverföring som förhindrar felaktig fältkombination
- Möjligheter till envägssynkronisering samt synkronisering med flera enheter. Med envägssynkronisering menas att ena enheten är master och skickar information till den andra enheten som är slave utan att själv göra några uppdateringar
- Möjlighet till utformning och konstruktion av filter som selektivt kan begränsa data synkroniseringen
- Flexibel konfliktlösning som ger användaren flera valmöjligheter för att hantera och lösa eventuella problem som uppkommer vid synkronisering
- Inga licenser ska behövas införskaffas för att använda protokollet
- Protokollet ska vara öppen vilket innebär att tredjepartsutvecklare ska kunna utveckla applikationer utan några restriktioner

2.5 Utvärdering av synkroniseringsalternativ

Här kommer de ovan nämnda synkroniseringsalternativen att presenteras systematiskt. Var och en av dessa kommer att evalueras enligt de kriterier som presenterats tidigare i detta kapitel. Därefter kommer en rekommendation att ges för användning i det fortsatta arbetet. En sammanställning av synkroniseringsalternativen kommer också att presenteras i form av en tabell.

2.5.1 IntelliSync

IntelliSync är ett synkroniseringsprotokoll som utvecklats av Pumatech och som är den ledande leverantören av mjukvara infrastruktur för mobila enheter. Grunden i protokollet utgörs av DSX, Data Synchronization Extensions, som är själva synkroniseringsprocessen. Pumatechs plattform heter Mobile Application Platform (MAP) och är en integrerad lösning som fullt stöder alla mobila enheter och standarder såsom WAP telefoner, Palm handdatorer och stationära datorer. [7]

Styrkan med DSX teknologin är att den penetrerar informationen på båda sidor i detalj för att på så sätt få kännedom om innehållet. Således har IntelliSyncs protokoll innehållskännedom och allt vad det innebär [3].

En annan styrka med detta protokoll är att det är oberoende av plattform, något som är väsentligt för en framgångsrik synkronisering [6]. Flertalet av de uppställda kriterierna är uppfyllda som t ex krav på att kunna hantera återkommande fält, simultan- och direkt synkronisering vilket gör detta till ett väldigt konkurrenskraftigt protokoll. Protokoll är ett s.k. öppet protokoll som innebär att den är öppen för tredjepartsutvecklare, vilket innebär att en användare kan utveckla egna applikationer. För att använda IntelliSync krävs det att licenser köps från Pumatech [14].

Vidare är Intelligents protokoll, ur effektivitetsmässigt perspektiv, väl genomtänkt och utvecklad. Riktlinjen har varit att etablera en ”snabb-sync” för att minimera fördröjningar och uppkopplingstiden mellan en enhet och servern. Denna ”snabb-sync” fås genom att endast utbyta de fält som har ändrats och först då kopplingen bryts görs uppdateringarna databasen på server sidan. [15]

2.5.2 ActiveSync

Ett annat tänkbart synkroniseringsalternativ är ActiveSync som utvecklats av Microsoft. Dess arkitektur är speciellt utvecklad för att klara av synkronisering mellan en enhet som använder Microsoft Windows CE operativsystem och en dator vilket kan ses som en markant inskränkning av plattformsoberoendet [8]. Den tillåter dock datasynkronisering av flera olika datatyper.

En fördel med ActiveSyncs plattform är möjligheten att kombinera tredjepartsutveckling och applikationer med olika hårdvara möjligheter för att få fram den bästa möjliga lösningen för en speciell målgrupp [2]. ActiveSync är en s.k. out-of-box synkronisering vilket innebär att programvaran är installerad på en dator som tillåter synkronisering av data. Detta tillåter att flera mobila enheter kan synkroniseras med en dator. Synkroniseringen kan ske över flera olika media, såväl fasta som trådlösa.

Vidare får användaren själv definiera hur innehållskänedom ska fungera och vilka villkor som ska uppfyllas. Detta möjliggör att flertalet av de önskvärda kriterierna kan uppfyllas om användaren själv utvecklar dem. För att få en hög effektivitet och minska sändningstiden använder ActiveSync filkompression och fildifferensering då stora mängder data ska skickas. Även egna applikationer kan skrivas för att förbättra effektiviteten.

En nackdel med ActiveSync är den tillsynes kompletterande mjukvara som behövs för att dataöverföringar ska fungera önskvärt vid t ex transmissions avbrott. [2]

2.5.3 ScoutSync

ScoutSync är den del av ScoutWare plattformen från Aether Software som sköter synkroniseringen. Den är utformad för att vara plattform- och operativsystemsberoende och fungerar därför i Palm OS samt i Windows CE miljöer. Den främsta anledningen varför detta är ett intressant alternativ är att Palm utnyttjar den i sin server. Synkroniseringen kan ske över flertalet media både fasta och trådlösa. [10]

En fördel med detta synkroniseringsalternativ är APIet (Application Programming Interface) som möjliggör utveckling av mobila applikationer. Med ScoutSync är det möjligt att utveckla applikationer så användare får tillgång till information oavsett var de befinner sig. [11]

Det protokoll som ScoutSync använder möjliggör snabb kommunikation och effektivt utbyte mellan klienten och servern. Protokollet har även två olika synkroniseringsmoder varav en är snabb och en som är långsam. Detta för att optimera och minska fördröjningstiden. Vidare är egenskaperna för ScoutSync väldigt likartade de för IntelliSync. [11]

2.5.4 SyncML

För att få en lösning på problemet med flera synkroniseringsprotokoll har ett allmänt synkroniseringsprotokoll framtagits. Initiativtagare till detta protokoll är Ericsson, IBM, Lotus, Motorola, Nokia, Palm Inc., Psion och Starfish Software. Målet med detta synkroniseringsprotokoll är att få ett protokoll med en sådan uppbyggnad och struktur så att den är kompatibel med flera olika enheter och datatyper. Protokollet ska följaktligen vara oberoende av plattform, tillverkare och applikation. [5]

SyncML protokollet ska verka som en öppen standard för synkronisering och erbjuda synkronisering av personlig data som t ex kontakter och kalendrar i mobila enheter med flertalet applikationer och tjänster såsom personal information managers (PIM), Internet kalendrar och adressböcker. Men även synkronisering av bilder, filer och andra applikationer ska vara möjligt. [13]

Eftersom det är ett öppet protokoll är det upp till användaren att definiera många av applikationerna vilket gör att de uppsatta kriterierna kan uppfyllas. SyncML kan synkronisera data oberoende av programspråk eller synkroniseringsapplikation som används av de olika enheterna. SyncML kan synkronisera data över alla ovan nämnda nätverk och protokoll. SyncML har tagits fram med dagens synkroniseringsbehov och krav i beaktande. Dessutom är SyncML licensfritt dvs. till skillnad från övriga synkroniseringsprotokoll är det kostnadsfritt att använda. [1]

Även de kriterier som berör effektiviteten och de höga kraven på trådlös kommunikation måste övervinnas. Genom att använda WAP Binary XML (WBXML) kan SyncML hålla de transmitterade datapaketerna små. SyncML kan även upprätthålla en koppling mellan enheter vid kortare avbrott. [12]

En av de mest uppenbara nackdelarna med detta protokoll är antalet produkter som stödjer den samt dess relativt begränsade utbredning och anonymitet. Dessa nackdelar grundar sig främst i att SyncML standarden är tämligen ny. Detta innebär att antal produkter som kommer att stödja detta protokoll kommer att öka i antal med tiden samt att det krävs även en viss tid innan protokollet etableras. En annan restriktion är att SyncML endast i dagsläget kan användas till synkronisering av kalender och kontaktlistor. Detta är enbart en tillfällig restriktion och SyncML är uppbyggd för att kunna synkronisera bilder och ljud i en snar framtid.

2.6 Motivering av val

I detta avsnitt kommer motivation rörande val av synkroniseringsalternativ att ges.

Nedan följer även en tabell som redovisar en sammanställning av de olika alternativen.

Kriterie	IntelliSync	ActiveSync	ScoutSync	SyncML
Innehållskänedom	Ja	Ja	Ja	Ja
Simultan synkronisering	Ja	Ja	Ja	Ja
Direkt synkronisering	Ja	Ja	Ja	Ja
Plattformsberoende	Ja	Nej	Ja	Ja
Effektivitetskrav	Ja	Ja	Ja	Ja
Önskvärda egenskaper				
· Protokollstorlek	Ja	Ja	Ja	Ja
· Envägssynkronisering	Ja	Ja	Ja	Ja
· Filter	Ja	Ja	Ja	Ja
· Flexibel konflikthantering	Ja	Ja	Ja	Ja
· Licensfri	Nej	Nej	Nej	Ja
· Öppen standard	Ja	Ja	Ja	Ja

Tabell 1: Sammanställning av de olika synkroniseringsalternativen

2.6.1 Val av synkroniseringsprotokoll

Flertalet av synkroniseringsalternativen har likvärdiga egenskaper men vissa grundläggande skillnader kvarstår dock. Med detta i åtanke måste ett val göras, ett val som grundar sig på de egenskaper som krävs av protokollet. En vital egenskap är plattformsberoendet varför ActiveSync inte kan ses som ett godtagbart alternativ. En annan orsak varför ActiveSync inte bör väljas är behovet av kompletterande mjukvara som behövs för att det ska fungera önskvärt vid t ex transmissionsavbrott.

Både ScoutSync och IntelliSync är välbeprövade och etablerade på marknaden medan SyncML är nytt vilket gör att en jämförelse mellan dessa kan bli svår att genomföra. I fallet ScoutSync och IntelliSync har den senare en fördel då den stödjer SyncML protokollet vilket är en fördel vid synkronisering mellan enheter med olika synkroniseringsprotokoll. Det som talar till ScoutSyncs fördel är att det är det protokoll som Palm använder sig av vilket indikerar att det är ett väl fungerande och utvecklat protokoll. Då Palm ingår i det konsortium som varit initiativtagare för framtagandet av SyncML är det en stor nackdel att ScoutSync inte stödjer SyncML. IntelliSync och ScoutSync besitter liknande egenskaper men fördelen IntelliSync har gentemot ScoutSync är att den har mer utvecklad konflikthantering samt ett bättre utvecklat filter. Med detta som grund verkar IntelliSync som det naturliga valet.

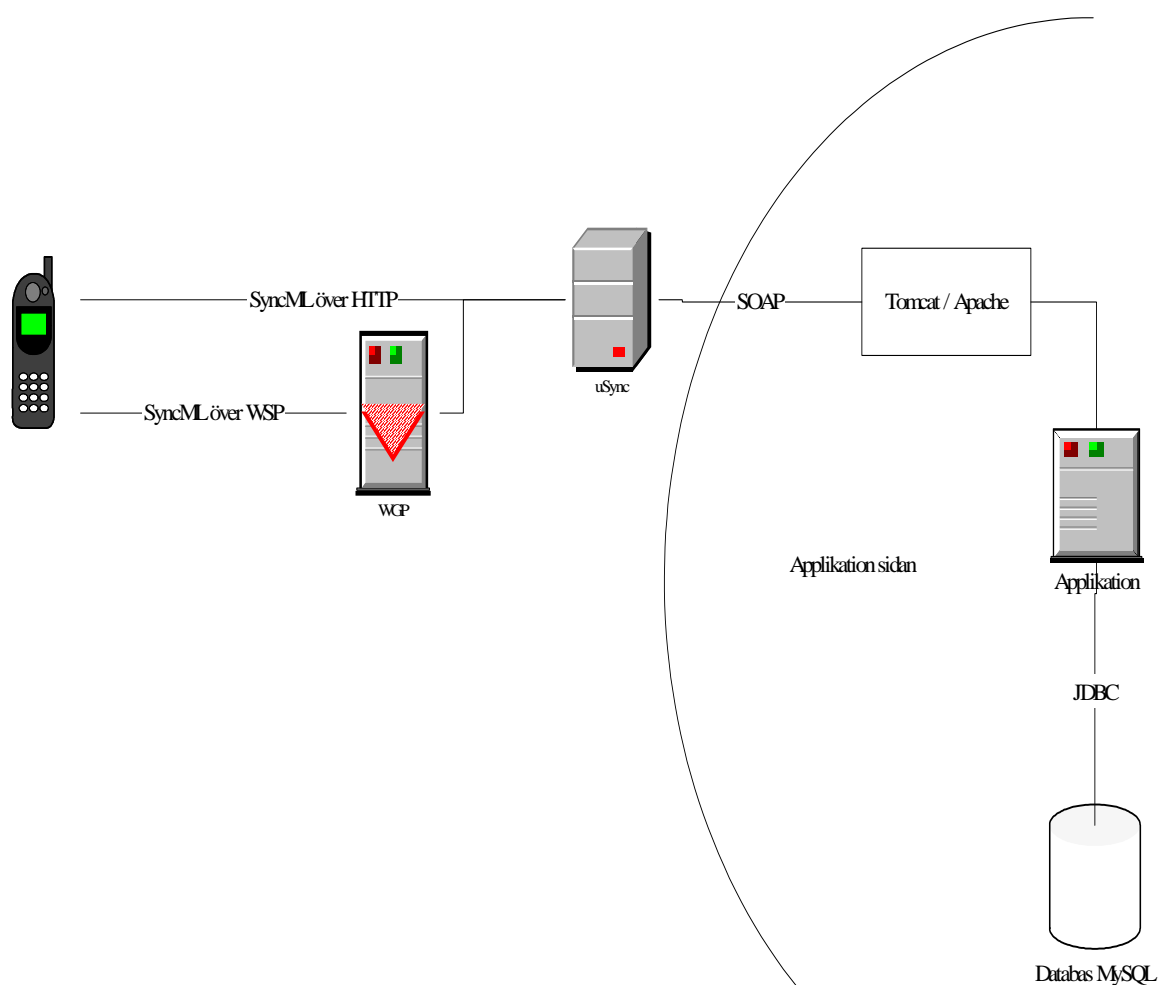
Vidare har SyncML utvecklats för att ha alla de egenskaper som förväntas av ett standardprotokoll. Ur utvecklingsperspektiv har SyncML de kvaliteter som behövs för att uppnå likvärdiga egenskaper som IntelliSync. SyncML är en öppen standard vilket ger alla som vill utveckla applikationer möjligheten utan några restriktioner. Dessutom behövs det inga licenser som i fallet med IntelliSync. Ett annat skäl är den framgång som SyncML har uppnått med den nya versionen. Fler och fler produkter stödjer SyncML och även antalet servrar som stödjer SyncML har haft en enorm ökning. Då även SyncML är den nya standarden blir valet mellan dessa protokoll SyncML.

3 Systembeskrivning

I det här kapitlet ges en beskrivning av systemlösningen. Applikationen som är en del av denna lösning kommer att presenteras och även de krav som ställs på den. Gränssnittet mellan applikationen och SyncML servern behandlas för att ge en bättre förståelse för hela systemet.

3.1 Systemlösning

Systemet används för att synkronisera data med hjälp av SyncML protokollet från flera användare via en uSync 1.0 server. Systemet består av ett godtyckligt antal användare med varsin enhet. Nedan illustreras det på vilket sätt de olika delarna förhåller sig till varandra. En mer ingående förklaring ges även senare.

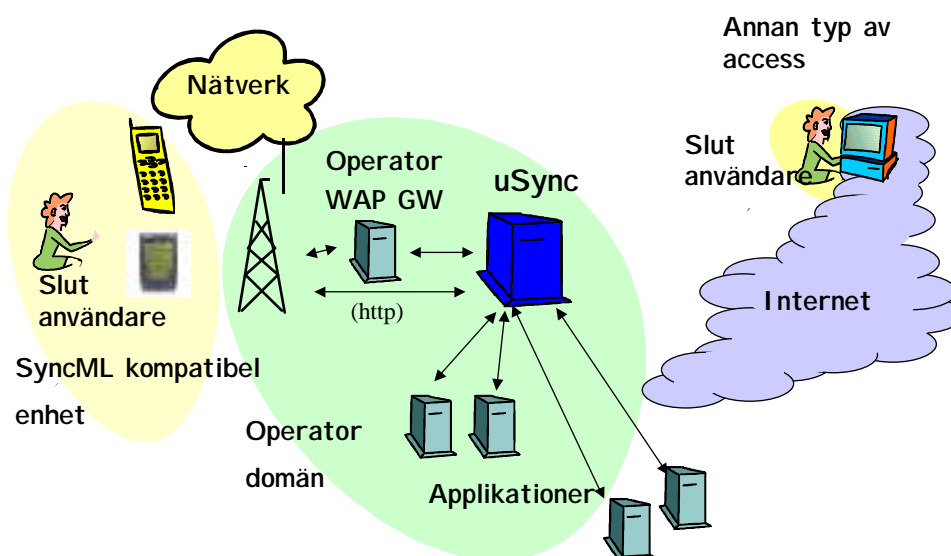


Figur 5. Systemlösning som beskriver hur de olika delarna förhåller sig till varandra.

SyncML synkroniseringsserver uSync 1.0 ska följaktligen kunna synkronisera data mellan en mobil enhet och en nätverksbaserad server genom att användandet av SyncML ver1.0 specifikationen. Den specifikationen tillåter synkronisering av all data som kan sparas i XML.

3.2 uSync servern

uSync¹, som fullt stödjer SyncML, kommer att fungera som en synkroniseringsmotor som sköter kommunikationen med alla applikationer istället för att ha flera synkroniseringsmotorer som kommunicerar med en applikation, se figur 6. Antalet applikationer kan variera och applikationerna kan också modifieras däremot förblir uSync oförändrad. All data, som sparas centralt, är tillgänglig från alla enheter och uSync har ett väldefinierat gränssnitt mot applikationerna. Dessutom fungerar den som en switch för att koppla ihop rätt enhet med anropad applikation. uSync stödjer synkronisering av kalender, adressbok, anteckningar samt att-göra-listor. [17]



Figur 6. uSync server med omkringliggande delsystem

Enheten ansluts till uSync servern genom en Wap Gateway Proxy, WGP, eller via http. Även mer traditionell synkronisering via kabel, IR och Bluetooth är möjligt. Då sker en lokal synkronisering av enheten och den lokala databasen på PC och sedan en synkronisering med uSync. Synkronisering sker således med hjälp av två databaser. [18]

¹ uSync servern har utvecklats av Ericsson

uSync servern har två gränssnitt mot applikationen. Ena används vid installation och konfigurerings av applikationen i uSync servern. Det andra är ett SOAP (Simple Object Access Protocol) baserat gränssnitt där alla uppkopplingar är initierade av uSync servern. Det är fortfarande applikationens uppgift att hålla reda på de ändringar som görs. Det är dessutom nödvändigt att ett SOAP gränssnitt implementeras i applikationen för att kommunikationen med uSync ska fungera tillfredställande. Användandet av ett SOAP gränssnitt medför att det mer komplexa SyncML gränssnittet inte behöver användas vilket underlättar vid framtagandet av applikationer. [18]

En annan fördel med uSync är att den har sammanhangs- och sessionskontroller för att skapa en så effektiv kommunikation som möjligt mellan klienten och servern. Servern är också utvecklad för att stödja åtskilliga användare. Varje meddelande tilldelas ett klient skapat ID och ett server skapat ID. Dessa skiljer sig från varandra och den resulterande relationstabellen finns i uSync servern. Den innehåller sålunda tillståndet för alla meddelanden. [18]

3.3 SOAP

SOAP är ett XML baserat protokoll som används för att utbyta information i decentraliserade system. Protokollet består av tre olika delar: ett kuvert som beskriver vad meddelandet innehåller samt vart det ska, en del kodningsmetoder för att uttrycka instanser av applikations definierade datatyper, och en konvention för hur procedur anrop och svar på dessa anrop ska hanteras. En styrka med SOAP är dess förmåga att kunna kombineras med andra protokoll i synnerhet med SyncML och http. Den data som skickas till uSync servern från enheten översätts till SOAP meddelanden och skickas sedan vidare. Anledningen till denna omvandling är att SOAP är mycket enklare att arbeta med samt att det är relativt utbrett.

3.4 Tomcat 4.0

Tomcat 4.0 är en servlet container som känner av vilka servlets² som finns tillgängliga och kopplar in önskad servlet. Det innebär att tomcat känner av vilka servlets som finns tillgängliga och registrerar dem. Tomcat 4.0 fungerar som en server add-on vilket betyder att den tar emot kommandon och dirigerar vidare dem till rätt servlet.

En webbapplikation är utformad som en hierarki av kataloger och filer i ett standardformat. Denna hierarki finns i två former. Det ena är en opackad form där alla filer och kataloger existerar separat i filsystemet. Den andra är en packad form som kallas Web ARchieve eller WAR fil. Den första är att föredra i utvecklingsstadiet medan den andra är att föredra då applikationen ska installeras.

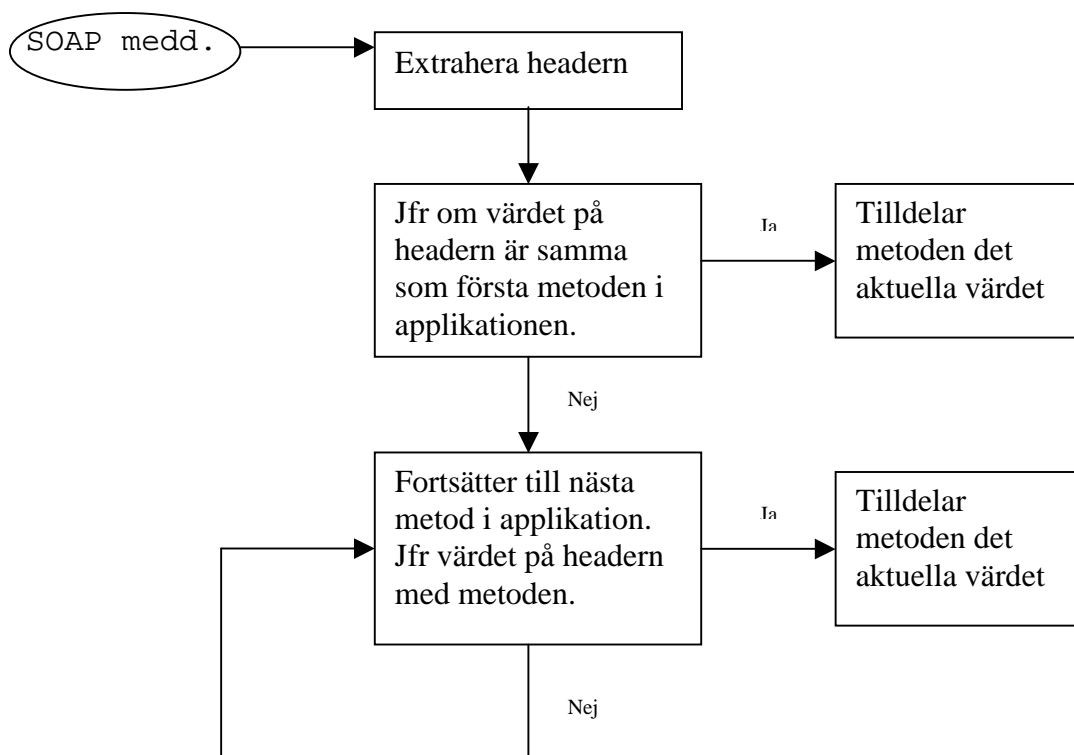
² Applikation som anropas av servern

3.5 Applikationen

Applikationen som ska utvecklas har till huvuduppgift att ta emot och tillgodose uSync 1.0 servern med data. Data som tas emot ska senare skickas vidare till en databas där den lagras. Applikationen använder dessa data för att skicka eller hämta önskad information från databasen. Varje synkronisering har en identifierare och denna identifierare definierar vilken typ av data synkroniseringen kommer att innehålla.

Applikationen måste registreras hos uSync servern och när denna registrering görs definierar applikationen vilken datasort den kan hantera. Detta innebär att en gemensam applikation för bearbetning av både kalender- och adressdata inte kan göras, utan applikationen måste delas in i två fristående delar. Applikationerna skulle påminna mycket om varandra varför endast en applikation utvecklas för kalendern. Applikationen kommer att skrivas i java för att vara plattformsoberoende.

Då uSync servern skickar data i SOAP format måste denna först översättas till text för att applikationen ska kunna ta emot dem och använda sig av dem.



Figur 7. Tilldelning av headervärdet till metod

3.6 JDBC

JDBC är ett Java API som är utformat i Java. JDBC har ett standard API som gör det möjligt att skriva applikationer genom att enbart använda Java. JDBC gör det möjligt att skicka SQL kommandon till mer eller mindre vilken relationsdatabas som helst. Detta innebär att endast ett program behöver skrivas oavsett till vilken relationsdatabastyp som anropas. JDBC har tre huvuduppgifter nämligen;

- etablera en koppling med databasen
- skicka SQL kommandon
- behandla resultaten

JDBC är ett lågnivågränssnitt vilket innebär att det används för att anropa SQL kommandon direkt. Det är utformat och designat på ett sådant sätt att det ska fungera som en grund på vilken ett hög nivågränssnitt ska byggas på. Ett hög nivågränssnitt karakteriseras av att det använder ett mer användarvänligt API. [19]

Den form av JDBC API: et som används här kallas två-skikts modell för databas access. Det innebär att applikationen kommunicerar direkt med databasen. För att det ska vara möjligt behövs en JDBC driver som kan kommunicera direkt med databasen. Användarens SQL kommandon skickas till databasen som i retur skickar tillbaka resultatet av kommandona.

3.7 MySQL

MySQL är en relationsdatabas som innebär att data lagras i separata tabeller istället för att all data sparas i en stor fil. Detta gör att databasen blir snabbare och mer flexibel. Tabellerna är länkade genom definierade relationer vilket möjliggör att data från flera tabeller kan kombineras vid behov. Det standardspråk som används för att arbeta i MySQL kallas Structured Query Language, SQL. I fallet med uSync servern kommer en databas med flexibilitet och snabbhet att vara av stor vikt varför MySQL är ett enstående alternativ. [20]

Tabellens utformning beskrivs i tabell 2 nedanför. För varje fält som ingår i tabellen tilldelas en typbeskrivning och en default värde samt ett antal andra parametrar som är av mindre betydelse.

Field	Type	Null	Key	Default	Extra
Entry	VARCHAR(50)	YES		NULL	
Anchor	TIMESTAMP()	YES		NULL	
EID	INT(14)	YES		NULL	
EIDList	VARCHAR(14)	YES		NULL	
EIDListAll	VARCHAR(14)	YES		NULL	
CharacterSet	VARCHAR(20)	YES		NULL	
RequestedCharacterSet	VARCHAR(10)	YES		NULL	
ReturnedCharacterSet	VARCHAR(10)	YES		NULL	
MaxTotalSize	INT(4)	YES		NULL	
OverHeadPerEntry	INT(6)	YES		NULL	
CheckDuplicate	INT(4)	YES		NULL	
ContentType	VARCHAR(10)	YES		NULL	
Archieve	INT(10)	YES		NULL	
Return	INT(10)	YES		NULL	

Tabell 2. Tabellbeskrivning av de ingående fälten

Ett exempel på tabellen ses nedanför, tabell 3. Entry är fältet där själv informationen som ska synkroniseras är skriven. Anchor är en tidsstämpel för att se när fältet senast blev redigerat. EID är ett unikt identitetsnummer för en Entry som används för att kunna komma åt ett speciellt fält. För en beskrivning av resterande fält, se [16].

Ett exempel på tabellen kan se ut som följande:

Entry	Anchor	EID	EID List	EIDL istAll	Character Set	Request edChar acterSet	Returned Character Set	MaxT otalSi ze	OverHead PerEntry	CheckD uplicate	Content Type	Archieve	Return
Ring Johan	2002011 5130000	2136											
Fika	2002011 6143000	2658											

Tabell 3. Exempel på tänkbar tabell

4 Resultat av applikationsutveckling

Applikation som utvecklades skulle kunna synkronisera kalenderdata från en mobil enhet med data från en dator. Detta skulle göras via en uSync server. Applikationen fungerade på önskvärd sätt, den testades lokalt på datorn, men vid test av applikationen tillsammans med resterande systemet uppstod det ett fel någonstans i systemlösningen. Eftersom det inte gick att testa systemet överhuvudtaget är det svårt att säga var felet ligger. Systemet kunde inte testas då alla delar är sammankopplade och när en del inte fungerar blir det omöjligt att testa. Med största sannolikhet ligger felet mellan uSync servern och tomcat, dvs. då servern försöker kontakta datorn, men det är även möjligt att felet ligger mellan tomcat och applikationen.

Målet att utveckla en fungerande applikation var därmed uppnått. Dessvärre kunde inte hela systemlösningen testas vilket innebär att inga slutsatser om hur applikationen fungerar med det resterande systemet kunde göras.

Resultatet blev följaktligen en fungerande applikation som kan användas men för att få den att fungera i symbios med uSync servern måste mer tid ägnas till kopplingen mellan servern och tomcat.

5 Slutsats och framtidsutsikter

Första delen av detta examensarbete bestod i att undersöka marknadsledande synkroniseringsprotokoll och objektivet välja ut det som är ledande enligt upplagda kriterier. Efter en grundlig utredning kom jag till den uppfattning att SyncML är det protokoll som är både ledande och mest lämpat för fortsatt arbete.

Huvudmomentet i detta arbete bestod i att utveckla en applikation som med hjälp av SyncML skulle genomföra en synkronisering av kalendern i en mobil enhet. Applikationen utvecklades planenligt och fungerade enligt uppsatta krav och kriterier. Däremot fungerade inte kopplingen mellan applikationen och uSync servern vilket ledde till att det inte går dra några slutsatser om hur applikationen fungerar tillsammans med resterande system.

Utvecklingsarbetet av applikationen har fungerat som planerat medan svårigheterna har varit uppsättningen av utvecklingsmiljön samt att få de olika delsystemen att fungera ihop.

Framtidsutsikterna för SyncML ser ljus ut och det kommer att fylla ut det tomrum som en avsaknad av en standard medför. Speciellt om SyncML utvecklas för att stödja synkronisering av bilder och ljud kommer detta protokoll att göra ett genombrott. Med utveckling åt det hållet kombinerat med de möjligheter som medföljer GPRS och 3G kommer applikationer som stödjer SyncML att gå en ljus framtid till mötes.

6 Referenser

- [1] Building an Industry-Wide Mobile Data Synchronization Protocol, 2001-06-16, <http://www.syncml.org/download/whitepaper.pdf>
- [2] Managing Mobile Devices in the Enterprise, 2001-06-18, <http://www.microsoft.com/MOBILE/enterprise/papers/managingmobile.asp>
- [3] DSX Technology – Data Synchronization Extensions, 2001-06-21, <http://www.pumatech.com/dsxwp.html>
- [4] Designing Effective Synchronizaion Solutions, 2001-06-21, <http://www.pumatech.com/syncwp.html>
- [5] Nokia showcases the world's first SyncML implementation with the Nokia 9210 Communicator, 2001-06-28, http://press.nokia.com/PR/200012/801382_5.html
- [6] Casio selects Pumatech's Intellisync software as synchronizarion standard for new handheld device, 2001-07-03, <http://www.pumatech.com/062601a.html>
- [7] Pumatech to publicly unveil Enterprise Intellisync software at PC Expo, 2001-07-03, <http://www.pumatech.com/062501.html>
- [8] RESCO – ActiveSync, 2001-07-04, http://www.resconet.com/resco/en/solution_actsync.asp
- [9] Compaq, 2001-07-04, <http://www.compaq.com/products/handhelds/activesync.html>
- [10] ScoutWare, 2001-06-13, <http://www.newones.com/scout.htm>
- [11] ScoutSync – connecting the extended enterprise, 2001-07-06 http://www.aethersystems.com/software/software_template.asp?PAGE=ssvc_sft_scoutsync_main
- [12] How it Works, 2001-07-06, http://www.nokia.com/syncml/how_to.html
- [13] Specifikation för synkronisering färdig, 2001-06-12, <http://www.computersweden.idg.se/includes/print.asp?id=001209-CS1>

- [14] All synchronization software was not created equal, 2001-07-09,
<http://www.pumatech.com/5reasons.html>
- [15] Invasion of data snatchers, 2001-07-09, <http://www.pumatech.com/enterprise/wp-1.html>
- [16] SyncAPI – Interface Description uSync 1.0, Alnås S. och Grealish P., 2001-08-24,
Ericsson Internal Information Nr 155 19-CRH 109 168 Uen
- [17] uSync 1.0 Feature Description, Ericsson, 2001-09-25
- [18] uSync 1.0 FAQ, Ericsson, 2001-09-25
- [19] JDBC Guide: Getting Started, 2001-10-03,
<http://java.sun.com/products/jdk/1.1/docs/guide/jdbc/getstart/introTOC.doc.html>
- [20] MySQL, 2001-10-03, <http://www.mysql.com/doc/>
- [21] Simple Object Access Protocol 1.1, 2001-11-30,
<http://xml.apache.org/soap/docs/index.html>

På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>

© [Arash Afsarinejad]