

**Kan integration av nätverksresurser förenklas
med hjälp av Javabaserade Jini?
(HS-IDA-EA-99-321)**

Markus Nilsson (b96marni@ida.his.se)

*Institutionen för datavetenskap
Högskolan i Skövde, Box 408
S-54128 Skövde, SWEDEN*

Examensarbete på det systemvetenskapliga programmet under
vårterminen 1999.

Handledare: Dan Lundh

Kan integration av nätverksresurser förenklas med hjälp av Javabaserade Jini?

**Examensrapport inlämnad av Markus Nilsson till Högskolan i Skövde, för
Kandidatexamen (B.Sc.) vid Institutionen för Datavetenskap.**

99.06.11

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: _____

Kan integration av nätverksresurser förenklas med hjälp av Javabaserade Jini?

Markus Nilsson (b96marni@ida.his.se)

Sammanfattning

Syftet med denna rapport är att undersöka om modern teknik kan åtgärda de problem som idag finns vid integration av nätverksresurser. En av grundtankarna i detta arbete är att nätverken i framtiden kommer att behöva anpassas till en mer flexibel och heterogen miljö. Anledningen till detta är att det i framtiden kommer utvecklas en rad nya produkter som ska kopplas in i nätverken.

Rapporten är uppdelad i tre delar. I första delen undersöks vilka problem som finns idag vid resurshandling i nätverk. Denna undersökning gjordes med hjälp av intervjuer med ett antal systemadministratörer som har praktisk erfarenhet av sådana typer av problem. Den andra delen undersöker en ny teknik vid namn Jini som utvecklats av Sun Microsystems. Jini undersöktes med utgångspunkt från de problem som togs fram i första delen. Tredje delen är praktisk och syftar till att undersöka hur Jini kan användas i praktiken.

Resultatet från första delen var sex stycken problem som var vanligast i olika kategorier. Problem med drivrutiner, rättigheter och administration för nätverksresurserna var det som var vanligast. Resultatet från andra delen blev att Jini klarade av flera av problemen på ett smidigt sätt men kräver separata administrationsprogram för vissa problem. Tredje delen resulterade i ett antal erfarenheter från en relativt begränsad fallstudie.

Nyckelord: Nätverksresurser, Nätverksproblem, Flexibilitet, Heterogenitet, Jini, Java,

Innehållsförteckning

1	Introduktion	1
2	Bakgrund	2
2.1	Resurshantering i nätverk - historik	2
2.1.1	Traditionella resurser och tjänster	2
2.1.2	Stabilitet.....	2
2.1.3	Homogena miljöer	2
2.2	Resurshantering i nätverk - nutidens trender.....	3
2.2.1	Nya typer av resurser och tjänster	3
2.2.2	Flexibilitet.....	3
2.2.3	Heterogena miljöer och plattformsoberoende	3
2.3	Resurshantering i nätverk - framtidens teknik	4
2.3.1	Lösning enligt Sun Microsystems	4
2.3.2	Java	5
2.3.3	Java Virtual Machine.....	6
2.3.4	Jini - introduktion	8
3	Problemprecisering	10
3.1	Generella avgränsningar.....	10
3.2	Vanliga problem vid hantering av nätverksresurser.....	10
3.3	Problem som kan lösas med hjälp av Jini och hur detta kan ske.....	11
3.4	Jini i praktiken.....	11
3.5	Förväntat resultat.....	11
4	Metod	13
4.1	Metodvalsprocessen	13
4.2	Möjliga metoder och metodval.....	15
4.2.1	Vanliga problem vid hantering av nätverksresurser	15
4.2.2	Problem som kan lösas med hjälp av Jini och hur detta kan ske.....	18
4.2.3	Jini i praktiken	21
5	Genomförande	24
5.1	Vanliga problem vid hantering av nätverksresurser.....	24
5.1.1	Förberedelser	24
5.1.2	Intervjuernas genomförande	26
5.1.3	Värdering av intervjuvaren	26

5.1.4 Erfarenheter från intervjuerna.....	26
5.2 Problem som kan lösas med hjälp av Jini och hur detta kan ske.....	26
5.2.1 Förberedelser	26
5.2.2 Litteraturstudiens genomförande	27
5.2.3 Värdering av materialet	27
5.2.4 Erfarenheter från litteraturstudien.....	27
5.3 Jini i praktiken.....	27
5.3.1 Förberedelser	27
5.3.2 Fallstudiens genomförande.....	28
5.3.3 Värdering av fallstudien	29
5.3.4 Erfarenheter från fallstudien.....	29
6 Analys	30
6.1 Vanliga problem vid hantering av nätverksresurser.....	30
6.1.1 Bearbetning av intervjuvaren	30
6.1.2 Sammanställning av intervjuvaren.....	31
6.2 Problem som kan lösas med hjälp av Jini och hur detta kan ske.....	32
6.2.1 Sammanställning av litteraturstudien	32
6.3 Jini i praktiken.....	37
6.3.1 Sammanställning av fallstudien.....	37
7 Slutsatser	38
7.1 Vanliga problem vid hantering av nätverksresurser.....	38
7.2 Problem som kan lösas med Jini och hur detta kan ske	38
7.3 Jini i praktiken.....	40
7.4 Sammanfattande slutsatser	41
8 Diskussion.....	42
8.1 Allmän diskussion	42
8.2 Erfarenheter	42
8.3 Vad detta arbete tillfört.....	43
8.4 Förslag till fortsatt arbete	43
Referenser.....	44

1 Introduktion

Idag börjar vi se nätverk och datakommunikation som en naturlig del av arbetet med datorer. Få moderna företag eller organisationer idag saknar någon typ av internt nätverk och flertalet kommunicerar också med övriga världen genom t.ex. Internet. Även i hemmen har datoranvändandet ökat dramatiskt de senaste åren och även här är det väldigt vanligt med någon form av uppkoppling mot resten av omvärlden.

Förutom de datorer och programvara som användarna har lokalt, så tillkommer i nätverket en mängd andra resurser som användare vill utnyttja. Exempel på detta är skrivare, bildläsare, modem och en mängd tjänster som olika typer av servrar erbjuder. Syftet med dessa resurser är att de enkelt ska kunna utnyttjas av användarna och nya resurser ska enkelt kunna integreras i nätverksmiljön.

Nätverksresurserna behöver dock inte begränsas till datorrelaterade applikationer. I princip finns idag möjligheten att bygga in "intelligens" och kommunikationsmöjligheter i alla elektroniska produkter som inte är alltför små. Intelligensen kan bestå av en processor som på något sätt kan behandla ett inkommande meddelande och eventuellt också skicka vidare information till andra enheter. Detta öppnar möjligheter till att fler produkter kan kopplas mot ett nätverk och på det sättet ingå i ett allt större kommunikationsnät. Exempel på sådana produkter kan vara videobandspelare, kylskåp, fläktsystem mm.

Anledningen till att olika typer av resurser ska kopplas in till ett nätverk är flera. Den mest självklara fördelen är att resursen kan delas av flera användare och på det sättet hålla nere kostnader för inköp. Istället för att varje användare har t.ex. en egen skrivare så är det effektivare att koppla in en skrivare till nätverket som alla kan använda.

En annan anledning att koppla resurser till nätverket är att administration och felsökning kan ske över nätverket. Om t.ex. kylskåpet eller videobandspelaren går sönder i en användares hem så är det möjligt för en reparatör att göra en felsökning via ett nätverk. Detta förutsätter självklart att en rad kommunikationsproblem löses men det är idag tekniskt möjligt.

Om nätverksprodukterna förses med en processor så leder det till en ökning av möjligheten för direkt kommunikation mellan resurserna. Ett fläktsystem som kommunicerar med sensorer i en byggnad och automatiskt ställer in rätt temperatur är idag ingen ovanlighet. Dessutom kanske detta system ska övervakas från en dator så att felsignaler kan skickas med en utförlig förklaring. Det stora problemet idag är att olika resurser och system talar olika språk vilket försvårar utbyggnad och sammankoppling.

Idag finns ett ökande behov av att enkelt integrera resurser i nätverket och även att förenkla för mobila datoranvändare som behöver växla mellan olika nätverk. Problemet är dock att dagens nätverk inte är tillräckligt flexibla för att detta ska vara möjligt. Dessutom finns en rad olika plattformar vilket försvårar situationen ytterligare. Identifiering av problem som uppstår i dagens nätverksmiljöer är en del av denna rapport.

Denna rapport undersöker också hur ny teknik kan lösa de problem som finns idag. Exempel på en sådan teknik är Jini från Sun Microsystems som undersöks både teoretiskt och praktiskt.

2 Bakgrund

Detta kapitel försöker belysa hur resurshanteringen i nätverk har skett traditionellt, vilka förändringar som håller på att ske och hur dessa förändringar kan hanteras.

2.1 Resurshantering i nätverk - historik

Problemet med flertalet av de nätverk och nätverksresurser som finns idag anser jag är att de är anpassade efter en stabil och homogen miljö. Med stabil menar jag en miljö där nya resurser och datorer sällan tillkommer eller försvinner. Med homogen menar jag en miljö där datorer eller tillbehör bygger på samma plattform och ”talar samma språk”. Under sådana förutsättningar så kommer nätverket säkerligen att fungera tillfredsställande.

2.1.1 Traditionella resurser och tjänster

Datorer och andra resurser har historiskt sett varit anpassade för att vara relativt orörliga delar i nätverksmiljön. Exempel på detta är stationära datorer, skrivare och bildläsare. Antalet typer av resurser som kan kopplas in mot nätverket har också varit relativt begränsat.

Att ansluta resurser av olika slag till ett nätverk har länge inneburit vissa problem. Vid initieringsskedet måste en rad konfigureringar göras för att användaren ska kunna komma åt och utnyttja resursen på ett effektivt sätt. När en resurs sedan ska tas bort eller bytas ut så innebär detta också viss administration (Melin, 1999).

2.1.2 Stabilitet

På grund av den stationära naturen hos de resurser som funnits traditionellt så har behovet av flexibla nätverk varit relativt litet. Istället har stabila miljöer byggts upp som fungerat tillfredsställande under lång tid.

Vid tillfälliga uppkopplingar mot nätverket, t.ex. vid besök av mobila användare, så får ofta nödlösningar eller omfattande konfigureringar göras för att användaren ska kunna få tillgång till resurserna. Dock så är detta idag relativt ovanligt eftersom graden av mobilitet fortfarande är låg.

2.1.3 Homogena miljöer

De första nätverken var till sin natur strikt homogena. En terminal kopplad till en stordator gjorde att det inte fanns något problem med olika operativsystem eller skiftande programmeringsspråk eftersom stordatorn var den enda intelligenta delen i nätverket. Med tiden kom en rad nya operativsystem ut på marknaden. I början i form av olika UNIX-dialekter men med PC-datorernas framgång utvecklades en rad skilda operativsystem varav många var helt eller delvis inkompatibla med varandra.

För att minimera problemen så har många företag och organisationer anpassat sina nätverk och resurserna i nätverket till i huvudsak en plattform. Detta trots att andra plattformar hade passat bättre för vissa ändamål. Resurser som läggs till i nätverket måste vara kompatibla med den plattformen och om inte så är fallet måste något annat väljas. Detta har bidragit till att Windows har fått mer eller mindre monopol eftersom flertalet av hård- och mjukvarutillverkare har denna plattform i åtanke vid framtagandet av nya produkter (Åsblom, 1999).

2.2 Resurshantering i nätverk - nutidens trender

Det ökande användandet av datorer och datorbaserade tjänster gör att användaren kommer att vilja utnyttja tekniken på ett mer mobilt och flexibelt sätt. Att kommunicera och få tillgång till information i princip var som helst kommer att vara en självklarhet i framtidens samhälle.

2.2.1 Nya typer av resurser och tjänster

Idag är trenden att mobila datorer och tillbehör blir allt vanligare (Åsblom, 1998). Handburna datorer, datorer inbyggda i telefoner, infraröda sändare och mottagare på datorer och tillbehör är bara några exempel på att kraven på flexibilitet och mobilitet har ökat. Till exempel ökade antalet handdatorer med 61% under 1998 (Wallström, 1999). Kraven ökar därmed att åtkomst av nätverksresurser ska fungera oavsett om användaren rör sig inom det egna företaget, i hemmet eller på tjänsteresan.

Dagens teknik har också gjort det möjligt för allt fler produkter som kan kopplas in till ett nätverk. Ett område som kommer att få en kraftig tillväxt är nätverken i hemmen och produkter som kan kopplas till hemmanätet (Nilsson, 1998)

2.2.2 Flexibilitet

Den ökande trenden av mobila datorer och tillbehör innebär att kraven på flexibla nät kommer att öka. Tillfälliga "besök" i andra nätverk än det egna kommer att höra till vanligheterna för framtidens datoranvändare. Vid dessa besök vill också användaren enkelt och snabbt få kontakt med de resurser som nätverket erbjuder utan att större åtgärder ska behöva utföras.

Detta kommer att kräva ett nytänkande när det gäller tekniken bakom resurserna och de tjänster som en resurs kan erbjuda. Idag ligger för mycket fokus på att datorn måste anpassas till den aktuella miljön vilket inte kommer att fungera i längden. De nya typerna av mobila datorer, speciellt handdatorer, har redan börjat bli ett problem eftersom det finns ett dåligt stöd för olika typer vid uppkoppling till nätverket. En artikel i IDG News (översatt i Computer Sweden, 1999), beskriver situationen som ett begynnande kaos, framförallt i USA där handdatorer börjar bli väldigt vanliga. Man skriver bland annat; *"Analytikerna råder företagen att ha regler för hur handdatorerna ska anslutas till nätverken. Men det handlar även om att ha stöd för ett begränsat antal gränssnitt för att inte få en för stor arbetsbörda"*

2.2.3 Heterogena miljöer och plattformsoberoende

Datormarknaden idag är heller inte homogen utan en rad operativsystem och språk finns tillgängliga. Trots att Windows-plattformen är väldigt dominerande så finns ändå alternativ som konkurrerar. UNIX-dialekter av olika slag har lyckats stå emot Windows även om de mest används på serversidan och inte så mycket som klientoperativsystem. Linux har också fått ökat stöd och ses väl av vissa som ett allvarligt hot mot Windows, även på klientsidan. Macintosh såg ett tag ut att konkurreras ut helt av Windows men är idag starkt på uppgång.

I samband med de mobila datorernas framgångar så växer också marknaden för operativsystem. Exempel på detta är handdatorerna där ett flertal nya operativsystem, t.ex. Epoc, Windows CE och Palm OS, har utvecklats de senaste åren.

2.3 Resurshantering i nätverk - framtidens teknik

Det växande behovet av flexibla nätverk och mobila datorbaserade produkter har gjort att ett flertal företag har börjat ta fram produkter som ska råda bot på problemen som uppstår. Vissa av dessa produkter är bara i experimentstadiet men några företag har färdiga produkter som teoretiskt kan fungera i dagsläget.

Problemet är den naturliga trögheten som finns i att skapa nya standarder på en marknad och få denna accepterad av ett stort antal företag. I detta fall finns kravet att standarden ska stödjas av hårdvarutillverkare vilket gör det hela mer komplicerat eftersom antalet hårdvarutillverkare är mycket stort.

I dagsläget (mars 1999) finns inte så många tillgängliga produkter som baseras på konceptet att enkelt kunna integrera nätverksresurser. Exempel på sådana produkter är:

- *Jini* från Sun Microsystems
- *Universal Plug and Play* från Microsoft
- *Infernospaces* från Inferno

Eftersom en studie av alla produkter skulle bli alltför omfattande så valde jag att studera varje teknik tillräckligt mycket för att kunna ta reda på hur tekniken ställer sig mot de problem som tagits upp tidigare i denna rapport.

- **Jini:** Sun Microsystems (1999a) skriver på sin webbplats att denna teknik har fått relativt stor respons från datormarknaden och stöds av många hårdvarutillverkare. Jini är en färdig produkt som kan börja användas, men antalet hårdvaruprodukter som stödjer tekniken är fortfarande litet. Grundtanken bakom Jini är att enkelt och snabbt kunna integrera datorer och tillbehör i en nätverksmiljö, oberoende av plattform. Kravet för att detta ska fungera är att nätverksresurserna ska kunna förstå Java.
- **Universal Plug and Play:** Enligt Microsoft (1999) har denna teknik liknande grundtankar som Jini men begränsas till Windows-plattformen. Istället för att bygga på ett programmeringsspråk så använder den sig av standardprotokoll för nätverk. Produkten har också fått stöd av flera av de större hårdvarutillverkarna vilket kanske inte är så konstigt med tanke på plattformens dominans.
- **Infernospaces:** Enligt Inferno (1999) ska Infernospaces koppla ihop alla möjliga produkter till ett stort fungerande nätverk. Tekniken ska inte vara beroende av plattform, programmeringsspråk eller hårdvara och verkar därför vara den idealiska lösningen. Nackdelen är att alla produkter måste stödja och kommunicera via Infernospaces vilket inte i dagsläget är en så utspridd teknik. Infernospaces stöds för tillfället heller inte av så många hårdvarutillverkare och finns ej som en färdig produkt.

Eftersom en av mina grundtankar är att framtidens nätverk måste stödja en heterogen miljö så är Microsofts lösning inte intressant. Infernospaces verkar vara en intressant produkt men är för tillfället inte tillräckligt mogen för att kunna utvärderas. Därför har jag valt att studera Jini närmare, eftersom det passar in på mina grundkriterier.

2.3.1 Lösning enligt Sun Microsystems

Sun Microsystems har under många år bedrivit forskning som bygger på att överbygga problemen med mer flexibla och heterogena miljöer där *Jini* är en av de

2 Bakgrund

senaste. Jini är den tredje delen i en vision som Sun har som består av följande komponenter (Lotsson, 1998):

- Ett programspråk som fungerar på alla plattformar.
- En virtuell maskin som kan köra programspråket i alla miljöer, oavsett processor och operativsystem.
- Ett nätverk som gör att dessa virtuella maskiner kan samverka som en enda stor dator.

Första delen har förverkligats med programmeringsspråket Java som numera är etablerat och relativt utbrett. Andra delen löstes till stor del med Java Virtual Machine som möjliggör exekvering av Javakod på flertalet av de kommersiella plattformarna. Tredje delen är alltså Jini vilket ska möjliggöra förenklad integration av nätverksresurser. För att skapa större förståelse för Jini förklaras de tre olika delarna lite mer i detalj nedan. Eftersom Jini kommer att beskrivas noggrannare i senare kapitel ges här endast en introduktion.

2.3.2 Java

Traditionell programmering har i de allra flesta fall handlat om att anpassa programmet till en specifik plattform. Detta gör att anpassning av programmet till en annan plattform blir både tidskrävande och kostsamt. Anledningen till detta är att hänsyn måste tas till både operativsystem och processor eftersom kommandon och systemanrop skiljer sig åt mellan olika system. Grundtanken med Java var att göra ett språk som var oberoende av plattform så att programmet kan skrivas en gång och användas av alla typer av datorbaserade produkter (Eriksson, 1997).

Historik

Enligt Eriksson började historien om Java 1990 då en utvecklingsgrupp om sex personer skapades. Gruppen bestod av de främsta systemutvecklarna på Sun och ibland dem fanns bland annat Bill Joy och James Gosling som idag är några av frontfigurerna på Sun.

Tankarna om ett plattformsoberoende språk uppkom när gruppen undersökte diverse elektroniska apparater och insåg att de allra flesta hade en egen typ av processor vilket gjorde att de inte kunde kommunicera med varandra. Ett av de mest utbredda programmeringsspråken var (och är) C++, ett objektorienterat språk som är både kraftfullt och relativt snabbt. Problemet med C++ är att det inte är plattformsoberoende och det finns en rad programmeringstekniska delar som är oerhört komplicerade vilket gör att det är svårt att lära sig. Vad man då gjorde var att använda delar från C++ så att programmerare skulle känna igen sig, gjorde om eller tog bort de många av delar som var svåra och gjorde språket mer generellt för att passa på olika plattformar.

Tanken med Java var alltså från början att användas i elektroniska apparater såsom TV-dekodrar, brödrostar och tvättmaskiner. Det var dock något annat som gjorde att Java fick sin stora spridning; World Wide Web. Eftersom Internet ska kunna användas av vem som helst så leder detta till att olika plattformar kommer att användas. Detta innebär inget problem så länge som sidorna endast består av text men om applikationer ska skickas till Internet-användarna så är plattformsoberoendet ett krav. Till en början användes Java mest som ett verktyg för att göra sidorna mer attraktiva och interaktiva men tyvärr också långsammare. Idag används dock Java i

större utsträckning som en viktig del av webbaserade, affärskritiska system där kraven är stora på funktionalitet och säkerhet.

Språkets egenskaper

För att ett programmeringsspråk ska få genomslagskraft så krävs en rad positiva egenskaper. Nedan presenteras några av de egenskaper som Sun (1998a) själva anser är skäl som gör att Java är ett programmeringsspråk att satsa på:

- *Java är Internetålderns programmeringsspråk:* Med hjälp av Java så kan man skapa objektorienterade, kompakta program som är anpassade till att skickas över nätverk.
- *Java är plattformsoberoende:* Fördelen med detta är självklar. Att skriva ett program som sedan kan köras på i stort sett alla plattformar möjliggör enkel spridning av koden. "Write Once, Run Everywhere" ("Skriv en gång, kör överallt") har blivit ett slagord för Java.
- *Java sänker kostnaden för programutvecklingen:* Likheten med C++ har gjort att många programmerare snabbt kan börja med utveckling i Java. Att Java har tagit bort de delar av C++ som ansetts allmänt som svåra gör att programutvecklingen går snabbare. Plattformsoberoendet gör också att arbetet med anpassningen till andra plattformar blir litet eller obefintligt.
- *Java kan användas i allt från stordatorer till mobiltelefoner:* Av sin natur så har Java ett väldigt brett användningsområde. Det fungerar både på serversidan i affärskritiska tillämpningar, som webapplikationer eller inbyggt i små elektroniska produkter.
- *Java har fått stor spridning:* Java är ett programmeringsspråk som fått så mycket stor spridning som på väldigt kort tid. Stödet från större företag och organisationer är också stort.

Java har dock inte bara positiva egenskaper utan självklart finns en del negativa sidor. De som oftast nämns är:

- *Java är långsamt:* Tidiga versioner av Java var oerhört långsamma men har med tiden utvecklats. Trots utvecklingen är dock Java fortfarande långsammare än t.ex. C++ vilket av många har uppfattats som starkt negativt. Orsaken ligger i språkets plattformsoberoende (Sridharan, 1997). Eftersom koden ska passa till flera plattformar kan den inte optimeras för en specifik plattform vilket gör programmen långsammare. Javakod kan inte heller köras direkt av operativsystemet utan måste först översättas av en så kallad "virtuell maskin" (se Java Virtual Machine, nedan).
- *Plattformsoberoende är inte helt problemfritt:* Den stora komplexiteten med att göra ett programmeringsspråk plattformsoberoende har gjort att Javaprogram inte alltid fungerar likadant på olika plattformar. "Write Once, Run Everywhere" har ofta ersatts av "Write Once, Test Everywhere" (Skriv en gång, testa överallt") vilket komplicerar utvecklingsarbetet (Danielsson, 1998).

2.3.3 Java Virtual Machine

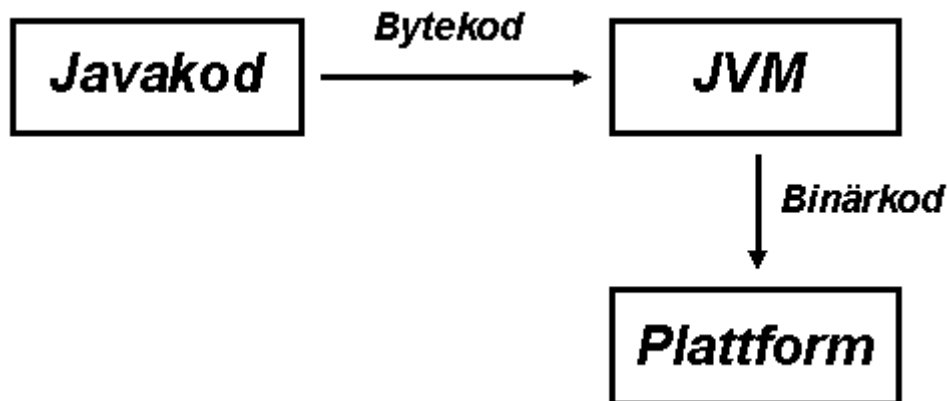
Att Java är plattformsoberoende är en sanning med modifikation. För varje plattform krävs nämligen en komponent som är anpassad till den aktuella miljön (Sridharan, 1997). Denna komponent brukar kallas virtuell maskin och i Javas fall, Java Virtual Machine (fortsättningsvis JVM). För att Javakod ska kunna exekveras på olika

plattformar så måste alltså en specialskrivna JVM finnas och idag finns JVM för de flesta större plattformar. Javaprogram som finns på World Wide Web, sk JavaApplets kan köras tack vare att de flesta större webbläsare har en inbyggd JVM.

Teknisk beskrivning

För att programkod ska kunna tolkas av datorn så måste den kompileras till sk binär kod. Den binära koden är då anpassad till det aktuella operativsystemet och processorn och kan därför inte köras på andra plattformar. En annan variant är att programkoden tolkas vid körning, vilket görs vid sk interpreterande programmeringsspråk (Sridharan, 1997).

När Javakod kompileras så skapas inte binär kod utan istället skapas sk byteskod vilket inte kan köras på någon plattform (Sridharan, 1997). Detta löses med hjälp av den virtuella maskinen som kan tolka byteskoden och sedan i sin tur översätta denna kod till binär kod, anpassad för plattformen. Java är alltså en slags blandning mellan ett kompileringsspråk och interpreterande språk.

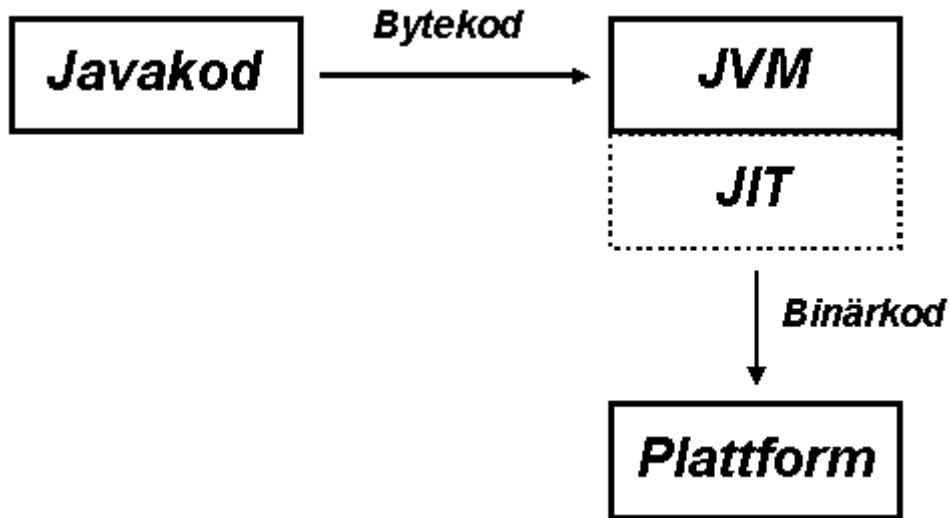


Figur 2-1: JVM översätter byteskoden till plattformsspecifik binärkod (Sridharan, 1997)

Just-In-Time

Nackdelen med att Java inte är ett rent kompilerat språk är att programmen blir långsammare att köra (Sridharan, 1997). Trots att JVM har utvecklats under åren och blivit allt snabbare så har detta inte räckt för att få prestanda som kan mäta sig med t.ex. C och C++. Den lösning som används idag är att lägga till en sk Just-In-Time kompilering (JIT).

När ett Javaprogram körs så sparar JIT-kompileringen mönster av den binärkod som produceras av JVM. När programmet körs igen så känner JIT-kompileringen igen byteskoden som skickas till JVM och kan då direkt skicka binärkod till operativsystemet utan att JVM behöver kompilera. Med denna teknik kan Javaprogrammen köras betydligt snabbare.



Figur 2-2: JIT-kompilatorn kontrollerar byteskoden och skickar binärkod direkt om den finns sparad (Sridharan, 1997)

2.3.4 Jini - introduktion

I detta kapitel ges endast en kort introduktion till Jini som beskriver grundtankar och tekniken i korthet.

Som beskrivits tidigare (se kapitel 2.3.2), så fanns redan i början av Javas utveckling en tanke om att bygga in ”intelligens” hos alla möjliga elektroniska apparater. Problemet är ju dock att en mängd olika processorer och operativsystem finns i datorbaserade produkter vilket försvårar eller omöjliggör kommunikation mellan dessa.

Med hjälp av Java och JVM (se ovan) kan dock dessa problem överbryggas. Eftersom JVM endast är mjukvara som simulerar hårdvara så behöver inga processorer eller operativsystem bytas ut, det enda kravet är att apparaturen kan lagra och köra en JVM på något sätt. Om sedan kommunikationsmöjligheter byggs in så kan Java användas som ett gemensamt språk för att kommunicera.

Detta är självklart inte problemfritt eftersom en specifik JVM måste skrivas för varje plattform. Java är dock en relativt öppen standard som gör att vem som helst kan skapa en egen JVM, bara den följer den gällande Javastandarden.

Grundtanken med Jini är att ”Jini-baserade” produkter, hårdvara eller mjukvara, automatiskt kan göra sig tillgängliga för andra i en nätverksmiljö. Meningen är att detta ska ske snabbt och enkelt utan att administratörer och användare i nätverket ska behöva göra några direkta konfigurationer (Sun Microsystems, 1999b).



Figur 2-3: Jini är ett tillägg till Java som möjliggör integrering av resurser i en nätverksmiljö (Sun, 1999b)

Enkelt uttryckt så är Jini bara en relativt liten utökning av det befintliga klassbiblioteket av Javakod. Resurser som utrustas med denna Jini-kod får dock vissa egenskaper som gör de tillgängliga i ett nätverk på ett helt nytt sätt. Nedan följer en kortfattad beskrivning av hur tekniken bakom Jini är tänkt att fungera (Venners, 1998):

- När en Jini-baserad resurs, t.ex. en skrivare, kopplas in på nätverket så skickas ett anrop ut på nätverket från resursen. Anropet talar om vilken typ av resurs det är och var resursen finns i nätverket.
- På nätverket måste det finnas en eller flera sk ”*Lookup-register*” som enkelt uttryckt är servrar som håller reda på alla Jini-resurser. När ett Lookup-register känner igen ett anrop så skickar den ett meddelande till resursen som talar om hur resursen ska kommunicera med Lookup-registret.
- När resursen har fått kontakt med Lookup-registret så skickar den information om sig själv, dvs vilka egenskaper den har och hur den kan användas. Denna information kallas resursens *interface*.
- När resursens interface finns i Lookup-registret så kommer andra resurser på nätverket, t.ex. en användardator, att automatiskt kunna se att en ny resurs lagts till. I Jini-terminologi kallas resurserna för *tjänster*, dvs varje resurs kan erbjuda tjänster till andra.
- En användare kan se vilka tjänster som finns genom t.ex. ett specialskrivet klientprogram. I detta program kan sedan användaren välja en tjänst och får då upp tjänstens användargränssnitt och kan utnyttja tjänsten på valfritt sätt.
- När en tjänst tas bort från nätverket så kommer tjänsten efter en tid att tas bort från Lookup-registret så att systemet hålls ”rent”.

3 Problemprecisering

Arbetet kommer att baseras runt de problem som gäller resurshantering i dagens nätverk och vilka lösningar som eventuellt kan finnas med hjälp av tekniken Jini. Detta kommer ske utifrån följande problemställning:

Kan integration av nätverksresurser förenklas med hjälp av Javabaserade Jini?

För att konkretisera och möjliggöra en uppdelning av problemställningen så har denna problemställning i sin tur brutits ned i följande delar:

- *Vilka är de vanligaste problemen vid hantering av nätverksresurser?*
- *Vilka vanliga problem kan lösas med hjälp av Jini och hur kan detta ske?*
- *Hur fungerar Jini i praktiken?*

3.1 Generella avgränsningar

Eftersom nätverksområdet är ett oerhört komplext och stort område så leder detta till att problemområdet blir väldigt omfattande. Därför har jag valt att göra en rad avgränsningar för att få en mer precis och konkret problemställning. Dessa avgränsningar gäller generellt för alla de ovanstående problemställningarna.

Jini är i första hand avsett för lokalt bruk, dvs ett lokalt företagsnät (LAN) eller ett mindre nät i hemmet. Det är också här som det i första hand är aktuellt med nätverksresurser som kopplas in på nätet. Detta är också det område jag har avgränsat mig till, vilket gör att problem i större nätverk såsom WAN eller Internet därför inte kommer att beröras.

Problem som uppstår vid olika nätverkstyper såsom Ethernet, Token Ring mm, kommer inte att beaktas eftersom detta är en fysisk begränsning hos den dator eller det tillbehör som ska kopplas till ett nätverk.

När det gäller plattformar/operativsystem så har jag valt att endast studera UNIX, Windows och Macintosh. Anledningen till detta är att dessa plattformar är de absolut vanligaste när det gäller användning på företag och i hemmen.

Specifika avgränsningar för de olika frågeställningarna tas upp under respektive kapitel.

3.2 Vanliga problem vid hantering av nätverksresurser

Idag läggs stora resurser i form av tid och pengar på att underhålla och bygga ut de existerande nätverken. En stor del av detta innefattar arbete med att byta ut, lägga till eller ta bort tjänster och resurser.

Målet med denna del är att identifiera några av de problem som finns idag och vilka problem som kan tänkas uppkomma om miljön blir mer flexibel och heterogen. Eftersom tanken bakom Jini är att på något sätt förenkla handhavandet eller lösa problem vid resurshantering så kan ett antagande vara att dagens situation innehåller ett antal problem.

Ett antagande som kan göras är att nätverksadministratörer är en grupp som stöter på problem av den typ som beskrivits ovan. Även användare som arbetar mycket mobilt antas stöta på problem av liknande karaktär, framförallt problem som berör flexibilitet. Att rikta in sig på dessa grupper leder dock till en viss avgränsning.

En annan avgränsning kommer att ske genom att bara vissa av problemen beskrivs noggrannare och att dessa problem bara beskrivs utifrån ett förutbestämt antal kriterier. (Se kapitel 4)

3.3 Problem som kan lösas med hjälp av Jini och hur detta kan ske

Jini kommer här beskrivas utifrån de problem som tagits fram i del ett. För varje problem kommer en beskrivning göras av hur Jini är tänkt att lösa problemet men utan att visa tekniska detaljer som t.ex. kod. Med utgångspunkt från dessa beskrivningar kommer sedan en bedömning göras för att avgöra om Jini kan lösa problemet eller inte.

Vid beskrivningen av Jini kommer en naturlig avgränsning ske till vissa delar. Arbetet kommer t.ex. inte att behandla de delar av Jini som berör distribution av processorkraft och hur resurserna ska administreras. Inriktningen kommer istället att vara att undersöka de delar som gäller initiering, åtkomst och borttagning av nätverksresurser och hur tekniken bakom detta fungerar. I första hand kommer detta att gälla resurser i form av hårdvara eftersom detta är den mest självklara och tydligaste delen av sk nätverksresurser. Dock kommer begreppet resurser att utvidgas till att gälla även mjukvara och andra tjänster.

3.4 Jini i praktiken

Eftersom Jini är en relativt ny teknik så finns i dagsläget få exempel på fungerande system. Målet med denna del är därför att undersöka Jini med utgångspunkt från praktiskt arbete. Ett exempel på en sådan undersökning kan vara att utföra någon typ av experiment. Experimentet kan vara att koppla ihop ett mindre antal datorer med minst två olika operativsystem i ett nätverk. På nätverket kan sedan en enhet kopplas in, t.ex. en skrivare, som är Jini-kompatibel och tanken är då att datorerna ska kunna använda skrivaren direkt utan att installera några drivrutiner eller göra några konfigurationer.

Syftet med denna del är att undersöka vilka förutsättningar som behövs och vilka problem som kan uppstå i praktiken. Om någon undersökning relaterad till praktiskt arbete inte går att genomföra så är det ändå intressant att se vilka problem som uppstod. Detta för att kunna göra en bedömning av hur lätt tekniken är att sätta sig in i. Anledningen till att denna del utförs sist är att denna del bygger på kunskaper från första och andra delen samt att det som sagts i del ett och två eventuellt kan illustreras med praktiska exempel. Eftersom tiden är begränsad avses dock inte att verifiera att Jini löser de problem som identifierats och undersökts i del ett och två.

Den naturliga avgränsningen blir här att undersökningen kommer att avgränsas till att gälla en mindre undersökning och bara visa vissa delar av Jini. Anledningen är begränsningar som finns i tid (max 40 timmar) och tillgänglig utrustning i dagsläget. Om inte hårdvara finns tillgänglig så kan detta ersättas av en undersökning av mjukvarutjänster.

3.5 Förväntat resultat

Det övergripande målet med arbetet är att på ett tydligt sätt beskriva hur eller om den nya tekniken Jini kan lösa de vanligaste och mest allvarliga problemen när det gäller resurshantering i nätverk. Tanken med detta är att försöka avgöra om Jini är en teknik som kan förenkla och förbättra den situation som råder idag. Eftersom rapporten

3 Problemprecisering

bygger på tre frågeställningar, där arbetet med varje del skiljer sig mycket åt, så kommer resultatet från varje del att presenteras på olika sätt.

I den första delen, där vanliga problem ska åskådliggöras, är syftet att på ett tydligt sätt presentera vilka problem som de intervjuade systemadministratörerna ansåg var vanligast. Det förväntade resultatet är då att det finns ett antal problem som återkommer vid intervjuerna och på det sättet kan urskiljas från de övriga. En eventuell kategorisering kan behöva göras om många olika typer av problem förekommer.

Nästa del, där Jini ska beskrivas och ställas mot de framtagna problemen, kommer resultatet visa på vilket sätt som Jini kan hantera olika problemsituationer. Fokus kommer att ligga på att presentera hur Jini ska lösa diverse problem och inte så mycket på tekniken bakom. Vid jämförelsen mellan de vanligaste problemen och hur/om Jini ska lösa dessa så kan eventuellt en tabell användas för att tydliggöra resultatet.

Resultatet från den sista delen, där Jini ska användas i praktiken, kommer att försöka att visa hur Jini kan användas för att lösa problemen praktiskt. Resultatet presenteras då i form av en implementation med källkod och/eller beskrivning av de erfarenheter som framkom. Vid en implementation så är målet att utföra några av de delar som togs fram i andra delen och visa hur det i praktiken kan gå till. En utförlig kommentar kommer också att skrivas för varje jämförelse för att förtydliga hur de olika experimentdelarna fungerade.

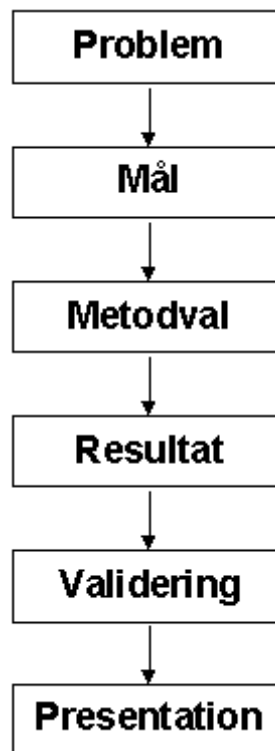
4 Metod

För att kunna lösa ett problem och nå ett bestämt mål så är det ofta nödvändigt och enklare om en strukturerad metod används. Problemet är dock att det i de flesta fall går att lösa ett problem på ett antal olika sätt vilket också leder till att valet av metod blir komplicerat. I detta fall är syftet med metoden att vara ett ramverk för genomförandet men ska också beskriva det förberedande arbetet innan genomförandet.

Vid forskningsinriktade undersökningar kan metoderna huvudsakligen delas in i två huvudkategorier: kvalitativa och kvantitativa metoder (Patel & Davidson, 1991). Enligt Patel & Davidson är skillnaden mellan dessa att kvalitativa metoder baseras på verbala analyser medan kvantitativa baseras på statistiska analyser. Vilken av dessa metodkategorier som passar bäst är ibland svårt att avgöra, en statistisk analys baseras ofta på ett stort urval men analysen av svaren kan bli mindre djupgående och vice versa.

4.1 Metodvalsprocessen

För att förenkla och förtydliga valet av metod så använder jag mig av en enkel mall som redovisas nedan. Mallen är ingen vedertagen modell men bygger delvis på ett översiktsschema för statistisk undersökningsmetodik utformad av Eriksson (1978).



Figur 4-1: Arbetsprocessen för metodval från givet problem till presentation av lösning

För att förtydliga ovanstående figur så förklaras varje steg närmare nedan:

Problem

Att identifiera och konkretisera problem är grunden till all forskning. Svårigheten ligger dock i att begränsa och formulera problemet på ett sådant sätt att beskrivningen blir entydig för olika grupper av intressenter. I kapitel 3 beskrevs problemställningen för denna rapport:

Kan integration av nätverksresurser förenklas med hjälp av Javabaserade Jini?

Jag valde sedan att dela upp denna problemställning i tre olika delar:

- *Identifiering av vanliga problem vid hantering av nätverksresurser*
- *Vilka av dessa problem kan lösas med hjälp av Jini och hur kan detta ske?*
- *Hur fungerar Jini i praktiken?*

Anledningen till denna uppdelning var att förtydliga hur problemet ska undersökas och förhoppningsvis lösas. Att börja med att ta reda på de problem som finns idag med hantering av nätverksresurser, verkade vara ett självklart val. Eftersom produkten (Jini) har som grundfilosofi att lösa ett antal problem så kan slutsatsen dras att det idag existerar problem som eventuellt kan elimineras. Ett alternativ kunde varit att börja titta på Jini direkt och sedan utifrån detta analysera hur bra tekniken passar in i de miljöer som finns idag.

Fallstudien, som ska bygga på hur Jini kan appliceras i praktiken, kan vara svår att realisera men ska förhoppningsvis spegla hur tekniken fungerar egentligen. Denna del ska försöka svara på hur svårt det är att realisera en Jini-lösning, även om den blir starkt begränsad. Fallstudien skulle kunna ses som en validering av tidigare delar men eftersom experimentet blir så pass begränsat så kommer bara en liten del valideras. Jag väljer därför att ha fallstudien som en separat del.

Mål

Utifrån den problemställning som är grunden för rapporten så är nästa steg att formulera mål för det fortsatta arbetet. För att nå målen så krävs i de flesta fall att en av begränsning sker, arbetet tenderar annars att bli för omfattande och ej så djupgående. Målet kan sedan konkretiseras ytterligare i det förväntade resultatet (se kapitel 0).

Metodval

Med utgångspunkt från problemställningen och det formulerade målet så är sedan nästa steg att utvärdera och välja en eller flera metoder. Typen av problemställning och önskat resultat innebär en naturlig begränsning i metodvalet eftersom vissa metoder av praktiska skäl inte är applicerbara på vissa problem.

Att utvärdera metoderna innebär att ställa fördelar och nackdelar mot varandra och på det sättet komma fram till den metod som är lämpligast. Vid t.ex. intervjuer så måste urval av intervjupersoner, typen av frågor och sammanställningen motiveras. På samma sätt måste litteraturens relevans och tillförlitlighet värderas vid en litteraturstudie.

Resultat

Beroende på metodval så erhålls någon typ av resultat. Detta resultat måste värderas och granskas för att hitta eventuella felkällor eller saknad information. Förslag på hur saknad eller felaktig information ska ersättas eller behandlas ska även tas upp här.

Validering

För att säkerställa resultat är det viktigt att kunna validera den information som erhållits. Detta kan göras på många sätt, t.ex. att komplettera med fler uppgifter, utföra någon typ av experiment eller göra om vissa metoddelar. I detta steg ska alltså förslag tas fram hur resultatet kan valideras och vilka förbättringar som kan uppnås.

Presentation

När resultatet är validerat ska informationen sammanställas och presenteras på ett tydligt och överskådligt sätt. Här är det viktigt att tänka igenom vilka delar av resultatet som ska lyftas fram och hur detta ska ske.

4.2 Möjliga metoder och metodval

Eftersom detta arbete bygger på tre olika delproblem så har jag valt att presentera de möjliga metodvalen för respektive del. Varje metodval baseras på den mall som presenterades ovan (kapitel 4.1).

4.2.1 Vanliga problem vid hantering av nätverksresurser

Denna del tar upp de möjliga metoder och metodval som kan göras vid framtagandet av vanliga problem vid hantering av nätverksresurser.

Problem

Problemet för denna del är att identifiera vilka problem som uppstår vid initiering, åtkomst och borttagning av resurser i ett nätverk. Definitionen på problem kan skilja sig beroende på vem som tillfrågas man jag har valt att definiera problem på följande sätt:

”Händelse som kräver att extra resurser i form av tid läggs ned vid initiering, åtkomst och borttagning av nätverksresurser.”

Ett problem behöver alltså inte vara en specifik problemsituation utan kan även vara den extra tid som måste läggas ned för att få nätverket att fungera.

Mål

Målet för denna del är således att presentera ett antal problem som är relevanta och relativt frekventa i sammanhanget.

Metodval

Som möjliga metoder för denna del har jag valt att begränsa mig till följande:

- Intervjuer
- Enkätundersökning
- Litteraturstudier

Anledningen till detta är att jag anser att informationen måste komma från personer som arbetat långa tider med nätverkshantering och har stött på ett antal problem. Att

på något sätt simulera problemsituationer i ett experiment anser jag inte vara praktiskt genomförbart.

- **Intervjuer:** Vid intervjuer kan detta ske med systemadministratörer och eventuellt med några användare som arbetar mobilt med datorutrustning. Denna undersökning tenderar då att bli mer kvalitativ än kvantitativ eftersom det är svårt att hinna med ett stort antal intervjuer. Däremot så kan relativt lång tid ägnas åt varje intervju vilket möjliggör en grundligare och mer noggrann undersökning.

I detta fall skulle en standardiserad intervju nog passa bäst med samma frågeställningar för varje intervju. Däremot bör frågorna vara öppna så att intervjupersonen inte är begränsad till några alternativ. En stor fördel med intervjuer är att justeringar lättare kan göras om svaren tenderar hamna utanför avgränsningarna för rapporten. Dessutom kan följdfrågor ställas som förtydligar svaren.

- **Enkäter:** En enkätundersökning kan i detta fall vara svår att utföra eftersom frågorna som ska ställas tenderar att kräva svar som är relativt utförliga. Speciellt svårt blir detta om enkätundersökning ska genomföras utan personkontakt, t.ex. via brev eller elektronisk post. Det är då mycket viktigt att tydligt beskriva förutsättningar och definitioner för att inte missuppfattningar ska ske. Fördelen med en enkätundersökning är att den skulle kunna utföras med många fler deltagare och på det sättet så skulle "det statistiska värdet" öka.
- **Litteraturstudie:** Vid en litteraturstudie så finns möjligheten att få en mer generell och övergripande bild över kända problem. Chansen finns också att någon annan gjort en liknande undersökning och kartlagt de vanligaste problemen. Jag tror dock att det blir svårt att hitta passande litteratur eftersom dessa problem är av det praktiska slaget. Dessutom så har nog många inte samma definition på problem som jag i detta fall (se ovan).

Utifrån ovanstående förutsättningar så har jag valt att använda mig av *intervjuer* för denna del av arbetet. Anledningen till detta är att jag anser att det krävs personlig kontakt med de sakkunniga för att få möjlighet att ställa följdfrågor och på det sättet bryta ned problemen i mindre bitar.

Mitt val av systemadministratörer eller nätverksansvariga som intervjupersoner anser jag vara väl motiverat eftersom dessa säkerligen har stött på problem av den karaktär som jag söker. Dessutom tror jag de har möjligheten att tänka sig in i en situation då nätverksmiljön bli mer flexibel och heterogen. Antalet intervjupersoner bör vara minst fyra stycken och beroende på vad tiden tillåter så är det självklart bättre med fler.

Vid intervjuerna är tanken att presentera ett antal scenarion för intervjupersonerna och utifrån dessa scenarion så ska intervjupersonen beskriva vilka problem som kan tänkas uppkomma. Exempel på scenario kan vara; "*En resurs ska läggas till i nätverket och samtliga användare ska kunna använda denna*". Tanken bakom dessa scenarier är att på ett verbalt sätt försöka simulera verkliga situationer som kan uppkomma.

För varje problem ska även intervjupersonen försöka bedöma hur allvarligt och resurskrävande problemet är. Anledningen till detta är att några av problemen ska studeras närmare i nästa del av arbetet och då kan valet göras utifrån dessa bedömningar. Om ett problem har nämnts vid flera intervjuer så är det också en anledning till att studera problemet närmare.

Resultat

Intervjuerna kommer förhoppningsvis att resultera i ett antal beskrivningar av problem som kan uppkomma. Svårigheten med frågor som resulterar i öppna svar är dock att det kan bli svårt att kategorisera och jämföra de olika provsvaren. Här får en bedömning göras och målet är att problemen kan delas upp i åtminstone ett antal grupper där problemen inom varje grupp är liknande.

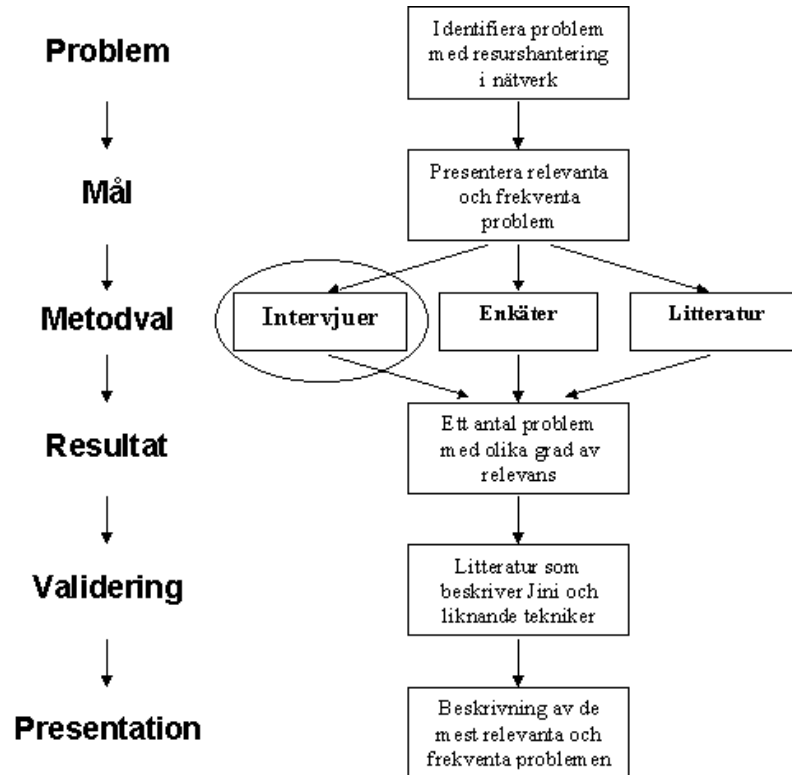
Validering

När resultaten är sammanställda så återstår frågan; är dessa problem de vanligaste och mest allvarliga som uppstår vid hantering av nätverksresurser? För att validera detta så finns ett antal olika alternativ. Ett alternativ är att helt enkelt utföra fler intervjuer och på det sättet kontrollera om liknande svar dyker upp igen. Ett annat sätt är att göra fler intervjuer men med omformulerade frågorna.

Jag har dock tänkt att istället leta i litteraturen för att validera mina resultat. En möjlighet är då att använda nästa del av arbetet, "*Problem som kan lösas med hjälp av Jini och hur detta kan ske*", som en del av valideringen. Eftersom Jini är tänkt att lösa ett antal problem som gäller hantering av nätverksresurser, så borde litteraturen som beskriver denna teknik också ta upp problem som finns idag och som kan tänkas uppstå i framtiden. Dessutom behöver denna validering inte begränsa sig till bara Jini utan litteratur som beskriver liknande tekniker kan även användas (se kapitel 2.3).

Presentation

Efter att ha validerat intervjuerna så återstår att presentera de problem som identifierats. Detta kommer göras i två steg. I första steget presenteras en sammanställning av alla de problem som identifierats under intervjuerna tillsammans med bedömningar av hur allvarliga och resurskrävande dessa är. I andra steget kommer ett antal problem (4-6 stycken) beskrivas noggrannare. Dessa problem är de mest frekventa och allvarliga problemen vilket motiverar att dessa lyfts fram ytterligare.



Figur 4-2: Illustration av metodval för första delen.

4.2.2 Problem som kan lösas med hjälp av Jini och hur detta kan ske

Denna del tar upp de möjliga metoder och metodval som kan göras vid undersökningen av vilka problem som Jini kan lösa.

Problem

Problemet i denna del är att undersöka på vilket sätt tekniken Jini kan användas för att lösa problem relaterade till hantering av nätverksresurser. Detta problem kommer att delas upp i två delar; dels en identifiering vilka problem som Jini är tänkt att lösa, dels hur detta kan ske tekniskt.

Mål

Målsättningen är att identifiera ett antal problem som kan lösas med Jini som är relevanta i sammanhanget, dvs rapportens fokus. Vidare ska tekniken för att lösa problemen beskrivas på ett tydligt och överskådligt sätt. Detta utan att gå in alltför djupt i tekniska detaljer såsom det praktiska genomförandet (kodning, nätverkskonfigureringar mm).

Metodval

Som möjliga metoder för denna del har jag valt att begränsa mig till följande:

- Litteraturstudier
- Intervjuer (med utvecklare av tekniken)

Anledningen till detta är att i dagsläget (april 1999), så finns det ingen organisation som använt tekniken i en "skarp" miljö. Det finns dock många enskilda

användare/programmerare som experimenterat med tekniken men dessa anser jag inte vara relevanta eftersom tekniken då inte används i ett sammanhang som är önskvärt i detta fall. Därför kan inte intervjuer utföras på liknande sätt som i föregående del (kapitel 4.2.1 - Vanliga problem vid hantering av nätverksresurser) och sedan jämföra om Jini löser problemen.

- **Litteraturstudie:** Vid en litteraturstudie kommer informationen att i första hand tas från skaparna av tekniken, dvs Sun Microsystems. Dessa kan åtminstone ge sin version av hur tekniken är *tänkt* att fungera genom produktinformation, tekniska specifikationer mm. Samtidigt så finns den mest utförliga information samlad hos Sun och förhoppningsvis den mest strukturerade. Ett alternativ är att undersöka vad utomstående intressenter har för uppfattning om hur tekniken är tänkt att fungera men risken är att dessa källor är mindre tillförlitliga och bygger på olika uppfattningar och tolkningar.

Vid en litteraturstudie kommer fokus att vara att hitta information relaterad till funktioner som på något sätt är tänkt att lösa ett bakomliggande problem. Det bakomliggande problemet kommer eventuellt inte nämnas i vissa fall och då kan en personlig tolkning krävas för att identifiera problemet. Tekniken för att lösa problemet kommer sedan beskrivas ytterligare för att visa hur detta är tänkt att fungera i praktiken.

- **Intervjuer:** Som beskrivits ovan så kan intervjuer inte ske på liknande sätt som i föregående del. Istället är tanken att intervjuer skulle kunna ske med personer som varit med och tagit fram tekniken. På detta sätt skulle specifika frågor kunna ställas som passar in i sammanhanget för denna rapport. Denna specifika information kan vara svår att hitta i litteraturen och om den finns så är den troligtvis utspridd på många olika ställen.

Den stora nackdelen med intervjuer är just att få tag i intervjupersoner som kan ställa upp. Eftersom tekniken är framtagen i USA så leder detta också till ytterligare ett hinder. Jag har dock fått en kortare frågestund med en chefstekniker på Sun i USA men då var inriktningen av mer allmän karaktär. Att ställa frågor till honom via e-post var inte heller så lätt eftersom han får ca 500 e-postmeddelanden om dagen (!).

Under ovanstående förutsättningar så väljer jag att använda *litteraturstudie* som metod för denna del. Anledningen till detta är den begränsade mängd intervjupersoner som finns tillgängliga och att mängden litteratur anser jag är tillräckligt stor.

Nackdelen med att hämta information från Sun kan vara att denna information tenderar att bli ”färgad” och inte så objektiv. Risken är stor att en alltför positiv bild presenteras vilket kan leda till att en felaktig bild presenteras. Det är därför viktigt att granska och värdera informationen kritiskt. Jag tror dock inte detta inte blir något större problem, det är trots allt upp till läsaren att på något sätt validera den information som erbjuds.

Den mesta litteratur som finns tillgänglig är enbart publicerad på websidor vilket kan försvåra tillförlitligheten för informationen. Det stora problemet med tillförlitlighet för websidor är att den ofta inte, till skillnad från artiklar och böcker, är ordentligt granskad av sakkunniga i ämnet. Jag anser dock att detta är en naturlig del av den tekniska utveckling som sker idag. Viss information kommer helt enkelt aldrig att presenteras i någon annan form än websidor. Dock säger jag inte att information om Jini aldrig kommer publiceras i någon annan form (som kan vara mer tillförlitlig) men i dagsläget så är denna information begränsad.

Resultat

Resultatet av litteraturstudien kommer att bli en identifiering av ett antal problem som kan lösas med Jini. Tekniken för att lösa dessa problem kommer också att beskrivas på en mer teknisk nivå. Liksom i föregående del så kan problem uppstå när det gäller att kategorisera de problem som tagits fram. Tanken är ju att dessa problem ska kunna jämföras med de problem som togs fram med intervjuer i föregående del. En uppdelning i olika grupper kan också här vara aktuellt där varje grupp innehåller liknade problem.

Validering

Vid en validering bör man ställa sig frågan; är Jini tänkt att lösa följande problem och är tekniken för detta rätt beskriven? Ett sätt att försöka validera detta skulle kunna vara att utföra en intervju med någon av utvecklarna av Jini. Detta skulle i alla fall ge en bekräftelse på att litteraturen är rätt tolkad.

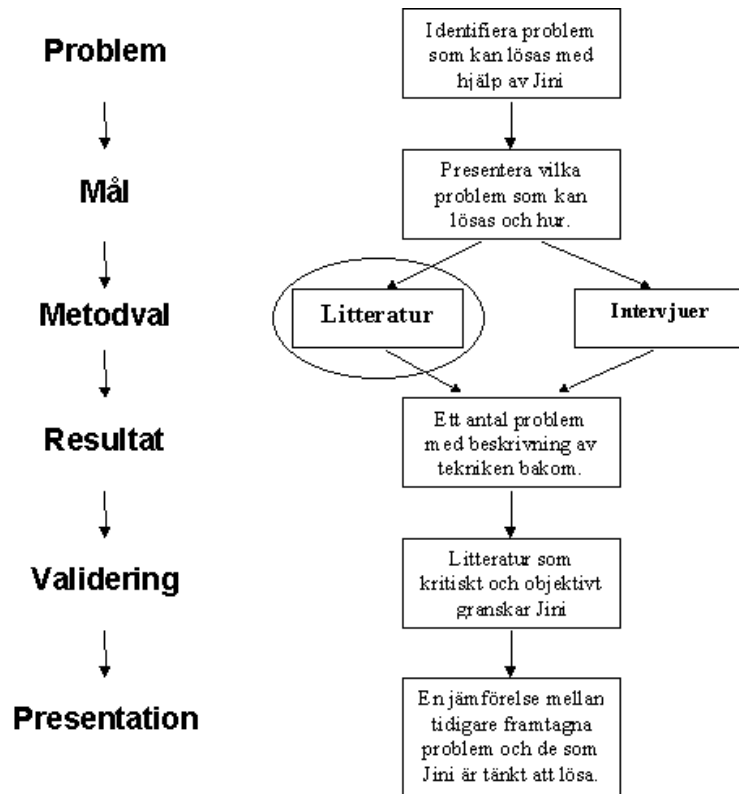
Dock anser jag att det skulle vara bättre att leta upp annan litteratur som på ett kritiskt och objektivt sätt granskar Jini och ger sin version av teknikens problemlösande förmåga.

Presentation

Vid presentation av resultatet så kommer först en jämförelse ske med de problem som togs fram i första delen, dvs intervjuerna med systemadministratörer. Om någon eller några problem överensstämmer så kommer lösningen på problemet att beskrivas mer noggrant. Om Jini är tänkt att lösa problem utöver problemen i första delen så kommer dessa också beskrivas om de är en väsentlig del av Jinis grundtankar.

Problemet med ovanstående jämförelse är att det i många fall kan bli problem med tolkningar. Ett problem kan ju beskrivas och tolkas på många olika sätt vilket försvårar jämförelsen. Risken är att inga problem överensstämmer till fullo men då får ett noggrant övervägande göras. I sådana fall är det viktigt att informera om de skillnader som finns.

4 Metod



Figur 4-3: Illustration av metodval för andra delen

4.2.3 Jini i praktiken

Denna del tar upp de möjliga metoder och metodval som kan göras vid en fallstudie med avsikt att undersöka Jini i praktiken.

Problem

Problemet i denna del är att undersöka hur Jini fungerar i praktiken. Detta genom att göra någon typ av fallstudie. Eftersom denna rapport är avgränsad till att endast studera hur hantering av nätverksresurser kan förenklas så kommer även fallstudien att ha denna inriktning.

Mål

Målet är att genom en fallstudie ge en inblick i hur Jini kan användas praktiskt och även att göra en bedömning av hur enkelt eller svårt detta är.

Metodval

Som möjliga metoder för denna del har jag valt att begränsa mig till följande:

- Implementation
- Studie av befintlig implementation
- Intervju med sakkunnig

I denna del är det viktigt att komma så nära det praktiska arbetet som möjligt. Istället för att i litteratur läsa hur man skulle kunna göra så anser jag att man måste utföra det själv eller studera något som redan är gjort. Skillnaden mellan teori och praktik är tyvärr ofta stor.

- **Implementation:** Vid en egen implementation så skulle fallstudien bli starkt begränsad och endast en liten del av problematiken skulle kunna undersökas. Risken är då att fallstudiens värde skulle minska drastiskt. Med tanke på den korta tid som kan läggas på en fallstudie så är det stor risk att detta blir fallet.

Som beskrivits i målet för denna del ovan så är även målsättningen att visa hur komplicerat det är att själv skapa en egen implementation. Det kan därför vara intressant att undersöka hur mycket resurser som behövs läggas ned för att bygga upp en fungerande praktisk miljö. Självklart skulle detta bli en subjektiv bedömning vars värde kan ifrågasättas men det kan ändå ge en indikation till andra på vad som krävs för att utnyttja tekniken.

- **Studie av befintlig implementation:** Trots att antalet implementationer är få i dagsläget och de som finns är mest i experimentsyfte så kan det ändå vara intressant att studera. Fördelen framför en egen implementation är att dyrbar tid kan sparas och implementationen kan få ett högre ”värde” som visar en större del av problematiken.

Nackdelen med denna typ av fallstudie är att det är väldigt svårt att uppskatta hur komplicerat och resurskrävande det har varit att ta fram den befintliga implementationen. Det är stor skillnad att studera något som andra gjort än att börja från början på egen hand.

- **Intervju med sakkunnig:** En sakkunnig i detta fall är en person som på något sätt arbetat praktiskt med Jini och gjort någon typ av implementation. En fördel med en intervju skulle vara att denna person skulle kunna redovisa för de problem som uppstått under det praktiska arbetet och hur mycket resurser som lagts ned. Om intervjupersonens förkunskaper redovisades så skulle en uppskattning kunna göras av vad som krävs för att arbeta praktiskt med tekniken.

Problemet är att det finns en väldigt liten mängd sakkunniga inom det aktuella området och att få kontakt med en sådan person kan bli svårt.

Under ovanstående förutsättningar så väljer jag att använda en blandning av två av dessa metoder; *implementation och studie av befintlig implementation*. Anledningen till detta är att jag anser att en sakkunnig person i nuläget inte är så stor chans att få kontakt med. Jag vill dessutom utföra delar av det praktiska arbetet själv för att kunna göra en noggrannare bedömning av hur problematiskt och resurskrävande det praktiska arbetet är. Dock är tiden alltför begränsad vilket gör att studier av en befintlig implementation är nödvändig för att resultatet ska få större relevans. Trots allt så är det ju ofta genom att studera andra som effektivt lärande uppnås.

Resultat

Resultatet från denna del kommer förhoppningsvis att visa hur Jini praktiskt kan användas för att lösa några av de problem som tagits fram i de två föregående delarna. Dessutom kommer en bedömning att kunna göras av hur lättanvänd och resurskrävande tekniken är att använda.

Validering

Att validera denna del är relativt svårt eftersom Jini i dagsläget inte används ”skarpt” i någon organisation. Det blir därför svårt att ge exempel på att tekniken verkligen fungerar praktiskt. Att själv ta fram en implementation leder till att väldigt begränsad

4 Metod

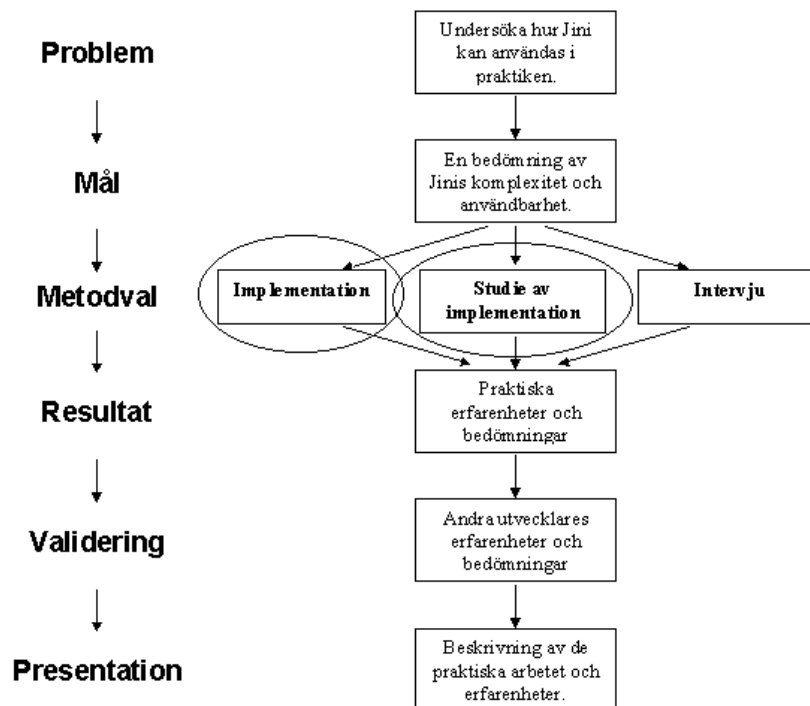
del av tekniken kan användas. Därför kommer större fokus läggas på att undersöka vad som krävs för att kunna använda tekniken i praktiken.

Ett sätt att validera skulle kunna vara att göra en enkätundersökning som skickas ut till den nyhetsgrupp som finns för "Jini-användare". Denna grupp används av personer som arbetar med Jini, både för experimentellt och kommersiellt bruk. Enkäten skulle kunna innehålla frågor om vilka erfarenheter de har haft och vad som är problematiskt vid praktisk utveckling. Med utgångspunkt från de problem som identifierades i första delen skulle medlemmar av nyhetsgruppen kunna få svar på hur eller om dessa kan lösas.

Presentation

Beroende på implementationens omfattning så kan en redovisning av Jinis problemlösande förmåga visas genom att lösningar på tidigare framtagna problem visas praktiskt.

Bedömningen av hur komplicerat Jini är att använda kan redovisas som en erfarenhetsbeskrivning, eventuellt kompletterad med andra utvecklares erfarenheter.



Figur 5-4: Illustration av metodval för tredje delen

5 Genomförande

Detta kapitel redovisar det arbete som utfördes under genomförandet. Strukturen på detta kapitel är, liksom vissa av de andra kapitlen, en uppdelning motsvarande de tre delproblemen. För varje delproblem redovisas förberedelser, genomförandet, värdering av material och erfarenheter.

I detta kapitel redovisas inget av materialet som togs fram under genomförandet utan detta redovisas istället i kapitel 7 och det slutliga resultatet presenteras i kapitel 8.

I första delen, *Vanliga problem vid hantering av nätverksresurser*, kommer arbetet med ett antal intervjuar att redovisas. Syftet med dessa intervjuer var att ta reda på vilka problem som kan uppstå när resurser ska initieras eller tas bort från en nätverksmiljö.

Det andra delproblemet, *Problem som kan lösas med hjälp av Jini och hur detta kan ske*, inriktar sig på att undersöka vilka problem tekniken Jini är tänkt att lösa. En jämförelse med problemen i första delen kommer också ske för att se hur eller om Jini kan lösa detta problem och i sådana fall hur.

Tredje delproblemet, *Jini i praktiken*, är tänkt att undersöka hur Jini kan användas i praktiken och ska även försöka värdera hur lätt eller svår tekniken är att använda.

5.1 Vanliga problem vid hantering av nätverksresurser

I denna del redovisas arbetet som genomfördes vid intervjuerna med systemadministratörer för olika verksamheter.

5.1.1 Förberedelser

Här beskrivs det förberedande arbetet som skedde innan intervjuernas genomförande.

Scenarier

Som beskrivits tidigare i metoddelen så använde jag mig av scenarier vid intervjuerna. Detta för att kunna simulera händelser som kan uppstå i verkligheten och på det sättet få intervjupersonerna att lättare tänka sig in den situationen. Dessutom så var målet att intervjupersonerna skulle få scenarier presenterade för sig som presenterade en mer flexibel och heterogen miljö.

Det första problemet i förberedelserna var alltså att ta fram ett antal scenarier som kunde återspegla det jag hade tänkt att undersöka. För detta tog jag fram ett antal ämnen som scenarierna borde behandla. Syftet med dessa ämnen var att senare enklare kunna dela in problem i olika grupper. Dessa ämnen innehåller även alla de aspekter som jag ville undersöka i mitt arbete.

- Initiering av nätverksresurs.
- Borttagning av nätverksresurs.
- Åtkomst av nätverksresurs.
- Initiering av nätverksresurs i form av mobil dator.
- Mer flexibla resurser och datoranvändare.
- Mer heterogen miljö.

5 Genomförande

Utifrån dessa ämnen skrev jag sedan ett antal scenarier som innefattade de olika delarna. Dessa scenarier presenteras nedan:

- *En ny resurs ska läggas till i nätverket och alla datorer ska ha kunna använda denna.*
- *En resurs ska tas bort permanent.*
- *En användare med bärbar dator, baserad på samma plattform/operativsystem som redan finns i nätverket, ska kunna komma åt ett antal resurser. Användaren har aldrig tidigare besökt nätverket.*
- *En rad nya produkter tas fram som kan kopplas in i ett nätverk. Många användare tar med sig dessa resurser och utnyttjar temporärt i det stationära nätverket. Förutom ägaren till resursen så ska fler användare i det stationära nätverket kunna utnyttja resursen.*
- *I en snar framtid har en stor del av användarna bärbara datorer eller handdatorer. Många av dessa baseras på plattformar/operativsystem annorlunda från de som finns i det stationära nätverket (t.ex. Linux, PalmOS, Windows CE, Epos, Macintosh). När dessa användare besöker nätverket ska de komma åt samtliga resurser som de har rättigheter till och de ska kunna utnyttja tjänster hos varandra vid möten.*

För att förenkla vid intervjuerna så skrev jag ner dessa scenarier tillsammans med instruktioner för intervjun så att alla intervjupersoner fick samma instruktioner innan intervjun startade (se Bilaga 1). Dessutom så skrev jag ett formulär som förenklade vid upptagning av intervjusvar (se Bilaga 2).

Ett problem som finns med dessa scenarier är att de kan leda till att vissa problem inte identifieras eftersom de inte uppstår i samband med de scenarier som presenteras. Jag anser dock att scenarierna innefattar de områden som jag vill undersöka och anser därför att de är berättigade.

Organisationer

Nästa steg i förberedelserna var att välja ut och kontakta ett antal organisationer. Vid valet av dessa fanns följande kriterier:

- Minst 30 användare inkopplade till nätverket.
- Minst två olika plattformar/operativsystem aktiva i nätverket.
- Någon tidigare erfarenhet av mobila användare som besöker nätverket.

Anledningen till dessa kriterier är för det första att det borde vara större risk att problem uppstår i nätverket om antalet användare är fler. Därför sattes en gräns vid 30 användare. Det andra kriteriet är viktigt eftersom en heterogen miljö borde leda till en mer problematisk miljö eftersom olika plattformar har olika standarder och olika arkitektur i grunden. De systemadministratörer som ansvarar för ett sådant nätverk borde också ha större möjlighet att tänka sig in i situationen att nätverket blir mer heterogent i framtiden. Det tredje kriteriet leder till att systemadministratören har fått större insikt i vilka krav som ställs när besökare ska ges tillträde till nätverket. Detta kan vara vilka specifika åtgärder som krävs för att upprätthålla säkerheten i nätverket och möjliggöra åtkomst av resurser för besökaren. Utifrån detta kan även systemadministratören tänka sig in i situationen att nätverket blir än mer flexibelt.

5.1.2 Intervjuernas genomförande

När det var tid för intervjuernas genomförande tog jag kontakt med fyra företag enligt nedanstående ordning (jag har valt att hålla alla namn anonyma eftersom några av verksamheterna helst ville detta):

- Datainstitution på högskola
- IT-enhet på högskola
- Driftenhet på IT-konsultföretag
- Driftenhet på nätverksinriktat företag

Den första intervjun, med datainstitutionen, fungerade som en pilotstudie i det avseendet att denna intervju var mer eller mindre en utvärdering av de scenarier jag tagit fram. Dock så fungerade denna intervju bra och jag ansåg att svaren från intervjun var relevanta i sammanhanget och några större missförstånd uppstod inte. Därför kunde jag använda denna intervju och fortsätta med de andra intervjuerna utan att ändra på frågeställningarna.

Varje intervju tog mellan 1 och 1½ timme och innan varje intervju informerade jag om syftet med intervjun och förklarade vissa av mina definitioner såsom *resurs*, *problem* osv.

5.1.3 Värdering av intervju svaren

Jag anser att intervjuerna var givande och gav bra information om vilka problem som kan uppstå vid resurshantering i nätverk. Dessutom fick jag många svar som var lika vilket kan tolkas som att det finns vissa problem som är vanligt förekommande. Tack vare att jag i förväg beskrev mina avgränsningar och definitioner tydligt så var det få frågor som tolkades fel. Intervjuernas utformning och genomförande värderas mer i kapitel 8.

5.1.4 Erfarenheter från intervjuerna

Den stora fördelen med intervjuer till skillnad från enkäter är att det är mycket enklare att få intervjupersonen att precisera och utveckla sitt svar. Speciellt viktigt är det vid öppna svar som här användes. Att rätta till eventuella missförstånd och feltolkningar är också lätt vid intervjuer vilket leder till ett mindre bortfall bland svaren.

5.2 Problem som kan lösas med hjälp av Jini och hur detta kan ske

Denna del redovisar arbetet med litteraturstudien där Jinis problemlösande förmåga skulle undersökas.

5.2.1 Förberedelser

Med utgångspunkt från de problem som togs fram i del ett så förberedde jag mig genom att leta igenom en stor mängd litteratur. Eftersom inga böcker eller forskningsrapporter fanns i ämnet så använde jag mig av litteratur från websidor. I första hand från Suns sidor (Sun Microsystems 1999a) men även artiklar av olika slag (Computerworld, 1999, Computer Sweden, 1999, Nätvärlden, 1999) samt programmeringsinriktade webplatser (Gamelan 1999, Javaworld 1999)

5.2.2 Litteraturstudiens genomförande

Med utgångspunkt från de problem som tagits fram i första delen började jag söka efter information som behandlade Jini problemlösande förmåga. Eftersom tekniken är relativt ny så var det svårt att hitta utförlig information om alla delar, vissa delar är ännu inte lösta och det kommer med jämna mellanrum program och tilläggs kod som gör Jini mer heltäckande. I de artiklar som beskriver Jini så är tekniken beskriven relativt överskådligt och den framtida visionen med Jini är det som beskrivs mest.

Där inte artiklar och websidor var utförliga nog fick informationen hittas på annat håll. Exempel på detta är tekniska specifikationer och sk "white papers" (Sun Microsystems, 1999c). Problemet med dessa är att de oftast beskriver tekniken på kodnivå vilket gör det svårt att se hur saker ska lösas praktiskt.

Problemen undersöktes ett efter ett och för varje problem dokumenterades om eller hur Jini är tänkt att lösa problemet.

5.2.3 Värdering av materialet

Svårigheten med litteraturstudien var att tolka informationen på rätt sätt. När ett problem undersöktes så kunde det hända att litteraturen beskrev att Jini kunde lösa något liknande men inte riktigt samma problem. Detta gjorde att en rad överväganden fick göras. Detta diskuteras dock mer i kapitlet 8 – Diskussion. Jag tycker ändå att jag lyckades ta fram information som svarar på om Jini kan lösa problemen eller inte. Vid tveksamma fall har jag diskuterat detta i kapitel 7 – Analys och försökt visa i vilka situationer som problemet kan lösas eller inte.

5.2.4 Erfarenheter från litteraturstudien

Att utföra en litteraturstudie som behandlar en så relativt ny teknik har sina för- och nackdelar. En fördel är att informationsmängden är relativt begränsad och därför kan studien koncentreras och fördjupas på ett enklare sätt. Detta kan dock också vara en nackdel eftersom litteraturen tenderar att bli ytlig eftersom ingen är riktigt insatt i ämnet. Istället får man förlita sig till tillverkaren av tekniken, dvs Sun, och tvingas lita på den information som de erbjuder.

Jag anser dock att den information jag studerat är trovärdig och pålitlig, trots allt har ett fyrtiotal av världens större teknikföretag granskat tekniken och ska i framtiden använda den i sina produkter.

5.3 Jini i praktiken

Denna del redovisar arbetet som genomfördes under fallstudien.

5.3.1 Förberedelser

Tanken med denna del var att en implementation skulle utföras som visade hur Jini kan användas i praktiken. En passande implementation skulle då vara att koppla ihop ett antal datorer med olika operativsystem i ett nätverk och sedan koppla in en nätverksresurs som direkt kunde användas av alla datorer. Problemet med detta var att hitta hårdvara som stöder Jini eftersom tekniken är så ny. Ett företag i Sverige tillverkar några Jini-baserade produkter men de kunde (förståeligt nog) inte låna ut någon sådan till mig.

Hårdvaran behövde dock inte vara något direkt problem. En Jini-baserad produkt är ju inget annat än hårdvara med en liten bit mjukvara i sig och tanken med Jini är en tjänst ska vara både hård- och mjukvara. Därför inriktades praktikfallet mot att undersöka hur en Jini-baserad mjukvarutjänst kan implementeras.

Som beskrevs i metoddelen tidigare (se kapitel 4) så var målet med denna undersökning inte bara att göra en implementation utan också att undersöka hur lätt eller svårt Jini är att använda. På grund av tidsbrist blev det dessutom svårt att hinna utveckla en seriös implementation.

Som förberedelse inför fallstudien så studerades kodexempel och det klassbibliotek (koden som är grunden för tekniken) som medföljer Jini (Sun Microsystems, 1999d)

5.3.2 Fallstudiens genomförande

Min undersökning började med att jag studerade Jinis klassbibliotek. Detta är relativt litet vid en jämförelse med Javas klassbibliotek men egentligen är ju Jini bara ett mindre tillägg som för tillfället inte ingår som standard. Jag upptäckte dock snart att det var svårt att börja med detta. För det första så anser jag att det behövs en relativt stor kunskap i Java för att kunna sätta sig in direkt i koden för Jini. För det andra så blir det väldigt abstrakt att bara studera fristående delar utan att förstå hur dessa kan användas tillsammans.

Jag insåg därför att jag behövde titta på färdigskrivna kodexempel för att få en större förståelse. Några sådana var inte dock inte så lätt att hitta. Tillsammans med Jini-paketet så följer det med några exempelprogram men inte källkoden så att denna kan studeras. På Suns websidor finns några demonstrationsprogram och sedan finns en del privatpersoner som lagt upp exempelkod på sådant de skrivit själva.

Jag studerade också en demonstrationsvideo från Sun (Sun Microsystems, 1998b) där de visar hur tekniken kan användas praktiskt. Denna video visar hur olika hårdvaruresurser såsom skrivare, webbkamera och handdator enkelt kan kopplas ihop och direkt kan användas utan några konfigurationer. Genom kontakter på Sun lyckades jag också få tag i källkoden till denna demonstration. Dock så kunde inte så mycket användas eftersom demonstrationen till största del bygger på hårdvara men vissa delar var användbara.

Med hjälp av olika kodexempel lyckades jag sedan bygga en mindre fungerande "Jini-miljö" som bygger enbart på mjukvara. Dock visade denna bara en mycket liten del av Jinis egenskaper. Denna "Jini-miljö" omfattade en Jinibaserad mjukvarutjänst som startas på en server och direkt syns denna tjänst i en enkel "Jinibrowser" på en klientmaskin. Att starta tjänsten via nätverket är dock en helt annan sak som kräver mer arbete. Dock så visade experimentet principen med att tjänster anmäler sig själva och direkt kan nås från en klientmaskin. Samma princip borde då också gälla för hårdvara men detta återstår att se.

Eftersom detta byggde på ett exempel som någon annan skrivit så var det relativt enkelt att implementera eftersom den var så pass liten. Att modifiera koden kräver dock förstås större kunskaper i både Jini och Java. Totalt så lade jag ner drygt en veckas tid på olika typer av experiment vilket jag anser inte är så mycket när det handlar om programmering.

5.3.3 Värdering av fallstudien

Som beskrevs ovan så tenderade fallstudien att bli mer en undersökning av hur lätt eller svår tekniken är att använda men jag anser att även detta kan vara viktigt att påvisa. Detta för att ge en förståelse för andra som vill utnyttja tekniken vad som krävs. Jag anser också att det är nyttigt att prova på en teknik som man undersöker i praktiken eftersom litteraturen ofta tenderar att förenkla och försköna beskrivningen.

Om mer tid hade kunnat avläggas så kunde fallstudien blivit mer ingående och därmed mer pålitlig. Dock anser jag att det är först när ett färdigimplementerat system, som används i verkligheten, är klart som en korrekt bedömning av teknikens potential kan göras.

5.3.4 Erfarenheter från fallstudien

Att utföra en fallstudie i form av en implementation är något som jag anser ofta tar mycket längre tid än vad som är tänkt. Därför är det viktigt att en tydlig avgränsning och specificering av fallstudien sker innan så att tiden kan uppskattas mer noggrant.

Mina erfarenheter från fallstudien diskuteras mer utförligt i kapitel 8.

6 Analys

Detta kapitel innehåller analys av de olika delarnas insamlade material med utgångspunkt från problemställningen.

6.1 Vanliga problem vid hantering av nätverksresurser

Denna del redovisar och analyserar det material som togs fram under intervjuerna.

6.1.1 Bearbetning av intervjuvaren

Här följer en sammanställning av de problem som togs fram under intervjuerna för de olika scenarierna. Siffran inom parentes anger hur många systemadministratörer som svarade likadant i de fall då det var mer än en systemadministratör som påpekade problem vid detta scenario.

Scenario 1

En ny resurs ska läggas till i nätverket och alla datorer ska ha kunna använda denna.

- Drivrutiner måste läggas in på varje maskin. (3)
- Bärbara användare ej på plats och dessa får "jagas".
- Användarprofiler i Win95 krånglar till det och gör att resursen visas för vissa men inte för andra.
- Vid olika nätoperativsystem är det nästan omöjligt.
- Automatisering skulle behövas. (2)
- Rättigheter över olika plattformar. (2)
- Åtkomst över olika plattformar.

Scenario 2

En resurs ska tas bort permanent.

- Drivrutiner ligger kvar i onödan (3).
- Länkar till resursen som ligger kvar skapar problem. (2).
- Svårt att identifiera vad som påverkas (2).
- Svårt att administrera centralt.
- Länkar och ikoner som ligger kvar stör användaren.

Scenario 3

En användare med bärbar dator, baserad på samma plattform/operativsystem som redan finns i nätverket, ska kunna komma åt ett antal resurser. Användaren har aldrig tidigare besökt nätverket.

- Användare måste läggas till och rättigheter ska tilldelas (3).
- Drivrutiner krävs för de olika resurserna (3).
- Olika nätverksprotokoll (2).
- Svårt att få tag i drivrutiner, ofta krävs speciella drivrutiner till bärbara.

Scenario 4

En rad nya produkter tas fram som kan kopplas in i ett nätverk. Många användare tar med sig dessa resurser och utnyttjar temporärt i det stationära nätverket. Förutom ägaren till resursen så ska fler användare i det stationära nätverket kunna utnyttja resursen

- Ny resurs måste registreras (4).
- Uppdateringsproblem vid många borttagningar och tillägg (2).
- Intressekonflikt kan uppkomma mellan datorer (servrar i huvudsak).
- Säkerhetsproblem, okänd hård- och mjukvara tas in i systemet (2).
- Olika nätverksoperativsystem försvårar situationen.
- Standardisering av protokoll, gränssnitt och drivrutiner behövs.
- Sätta rättigheter för ny resurs (3).

Scenario 5

I en snar framtid har en stor del av användarna bärbara datorer eller handdatorer. Många av dessa baseras på plattformar/operativsystem annorlunda från de som finns i det stationära nätverket (t.ex. Linux, PalmOS, Windows CE, Epoc, Macintosh). När dessa användare besöker nätverket ska de komma åt samtliga resurser som de har rättigheter till och de ska kunna utnyttja tjänster hos varandra vid möten.

- Loginscripts för olika plattformar bör skrivas för att förenkla för användarna.
- Drivrutiner för skilda plattformar kan krångla till det (2).
- Standardisering krävs för protokoll och drivrutiner. (3)
- Stöd för fler plattformar i nätverken krävs (2).
- Omöjligt med dagens nätverk, nya arkitekturer krävs!
- Automatisering krävs för tillägg av användare.
- Säkerheten äventyras när nya system tas in i nätverken.

6.1.2 Sammanställning av intervjuvaren

Efter att intervjuerna var genomförda började arbetet med att bearbeta och sammanställa intervjuvaren. Det var relativt komplicerat eftersom det fanns många svar som liknade varandra men inte var exakt samma. Det är ju ett av problemen med öppna svar vid intervjuer. För att förenkla sammanställningen så delade jag in svaren i olika kategorier. De kategorier jag använde var de som jag använde för att ta fram scenarier tidigare (se kapitel 5.1.1):

- Initiering av nätverksresurs.
- Borttagning av nätverksresurs.
- Åtkomst av nätverksresurs.
- Initiering av nätverksresurs i form av mobil dator.
- Mer flexibla resurser och datoranvändare.
- Mer heterogen miljö.

Med utgångspunkt från dessa kategorier så delade jag in de olika problemen under en kategori och de vanligaste problemen togs fram för respektive kategori. Dessa presenteras i Tabell 7-1 nedan. Eftersom nästa del är beroende av resultatet av denna del så presenterar jag resultatet redan här.

Kategori	Vanligaste problem
Initiering av nätverksresurs	Drivrutiner behöver läggas till och konfigureringar behöver göras på varje klientmaskin.
Borttagning av nätverksresurs	Drivrutiner och länkar till resurser ligger kvar på klientmaskinerna.
Åtkomst av nätverksresurs	Resursen måste registreras och rättigheter måste sättas.
Initiering av nätverksresurs i form av mobil dator	Tilldelning av rättigheter för användare måste utföras.
Mer flexibla resurser och datoranvändare	Mycket administrations- och uppdateringsproblem.
Mer heterogen miljö	Krav på standardisering av protokoll och standarder ökar.

Tabell 7-1: De vanligaste problemen för olika kategorier.

6.2 Problem som kan lösas med hjälp av Jini och hur detta kan ske

Denna del redovisar och analyserar det material som togs fram under litteraturstudien.

6.2.1 Sammanställning av litteraturstudien

Detta kapitel redovisar hur varje problem som togs fram i föregående del undersöktes med utgångspunkt från Jini. För teknisk introduktion till Jini se kapitel 2.3.4. För varje problem anges problemställning och det sätt jag anser Jini löser problemet att skrivas med kursiv stil.

Problem 1

Vid initiering av en nätverksresurs, behöver drivrutiner läggas till och konfigureringar behöver göras på varje klientmaskin.

Detta problem är ett av det vanligaste som tas upp vid beskrivningar av Jini och jag anser att det är ett av huvudsyftena med Jini. Vid intervjuerna så framställdes inte detta problem som speciellt allvarligt men däremot väldigt tidskrävande. En del såg det knappt som ett problem utan något som ingår i de vanliga arbetssysslorna. Men med min definition på problem så är detta en händelse som kräver resurser utöver det vanliga och kan därför tolkas som ett problem. Speciellt blir detta problematiskt vid riktigt stora nätverk och vid mer flexibla nätverksmiljöer.

Enligt Sun (Sun Microsystems, 1999a) så kan detta problem lösas med Jini på följande sätt:

Tanken med Jini är att tjänsten som ska användas, själv ska bidra med sin egen drivrutin och sitt eventuella gränssnitt mot användaren. Drivrutinen och det eventuella gränssnittet skickas som Javakod till användaren via nätverket när användaren behöver använda resursen och därmed så behöver inga drivrutiner läggas på klientmaskinen. På detta sätt behöver inga konfigurationer göras på varje klientmaskin.

Enligt denna beskrivning kan Jini sägas lösa detta problem utifrån den information som tagits fram. Vid initiering av en nätverksresurs behövs alltså inte drivrutiner läggas in eller konfigurationer göras för varje klientmaskin om Jini används.

Problem 2

Vid borttagning av nätverksresurs, ligger drivrutiner och länkar till resurser kvar på klientmaskinerna.

Vid intervjuerna så ansågs detta inte vara ett så allvarligt problem eftersom det var allmänt accepterat att datorerna med tiden innehåller ”skräp” från hård- och mjukvara som tagits bort. Dock ansåg alla att detta skulle bli allvarligt om nätverksmiljön blev mer flexibel och resurser skulle läggas till och tas bort med jämna mellanrum.

Enligt Sun (Sun Microsystems, 1999a) så kan detta problem lösas med Jini på följande sätt:

Lösningen på detta problem är direkt kopplat till lösningen på föregående problem. När en resurs används så kan koden för att använda denna resurs laddas ner till klienten. När klienten har använt resursen klart för tillfället så avslutas den Javaprocessen och koden för resursen tas bort från klientens virtuella Javamaskin. På detta sätt behöver ingen information om resursen lagras på klienten permanent utan istället dynamiskt.

Om resursen ska tas bort från nätverket så försvinner alla länkar till denna på följande sätt: varje resurs som kopplas in till nätverket anmäler sig till det centrala ”lookup-registret” som håller reda på alla ”Jiniresurser”. För att inte resursen ska försvinna från registret så måste den med jämna mellanrum tala om att den finns kvar och förlänga sin tid i registret. Detta kallas i tekniska termer för ”Lease” och fungerar ungefär som en parkeringsmätare. Om resursen tas bort från nätverket så finns det inget som kan förlänga tiden och därmed så tas tjänsten bort från registret.

På klientsidan så kommer resursen därför inte längre att synas i det klientprogram som visar vilka Jiniresurser som finns i nätet och detta sker i princip omedelbart. På detta sätt så belastas inte klienten med några kvarliggande drivrutiner och länkar till resurser som inte finns.

Enligt denna beskrivning kan Jini sägas lösa detta problem utifrån den information som tagits fram. Vid borttagning av en nätverksresurs ligger alltså inga drivrutiner eller länkar till resursen kvar på klientmaskinen.

Problem 3

Vid åtkomst av nätverksresurs, måste resursen registreras och rättigheter för resursen måste sättas.

Detta problem var egentligen inget stort problem eftersom de nätverksmiljöerna som de intervjuade systemadministratörerna arbetade i var relativt stabila. Däremot så

insåg alla att detta skulle bli ett resurskrävande arbete om miljön blev mer flexibel och om resurser lades till av användare istället för systemadministratören.

I dagens nätverk så måste en ny resurs på något sätt registreras av systemadministratören för att användarna i nätverket ska kunna komma åt resursen. En variant är dock att den nya resursen kopplas lokalt till en dator i nätverket och att sedan användaren "delar ut" resursen till andra. Denna lösning är dock så flexibel eftersom resursen inte går att använda om datorn stängs av eller tas bort från nätverket. Istället är tanken att en resurs ska kopplas direkt till nätverket utan att vara beroende av någon annan resurs.

När resursen är registrerad så ska rättigheter sättas för att begränsa eller möjliggöra användning av den nya resursen. Detta sker enligt intervjupersonerna helt manuellt och vissa tyckte att någon typ av automatisering vore önskvärd.

Enligt Sun (Sun Microsystems, 1999a) så kan detta problem lösas med Jini på följande sätt:

När en resurs kopplas in i ett Jinibaserat nätverk så kommer resursen själv att anmäla sin närvaro hos "lookup-registret". Detta kallas med Jini-termer för "discovery" och sker genom ett anrop över hela nätverket. När registret får en sådan anmälan så skickar det tillbaka information om var registret befinner sig så att resursen nästa gång kan kommunicera direkt med registret. Resursen skickar sedan information om sig själv i form av Javakod och denna kod sparas i registret och kallas för resursen eller tjänstens "interface". I Jini-termer så har resursen utfört en sk "join", dvs förenat sig med nätverket.

På datorerna i ett Jini-nätverk kan ett specialskrivet program användas som kan visa resurserna som finns i registret på olika sätt. Om ett sådant program används så kommer resursen direkt att dyka upp med sitt namn tillsammans med en eventuell ikon. Användaren kan då använda denna resurs på olika sätt beroende på vilken typ av resurs det är. Om det t.ex. är en webbkamera så kan användaren dubbelklicka på ikonen och sedan visas ett gränssnitt skrivet i Java som visar kamerabilden. Om det är en skrivare kan användaren skriva ut genom att välja denna resurs.

På detta sätt kan Jini förenkla och i princip ta bort arbetet med att registrera resurser som systemadministratören eller en användare lägger till. Resursen anmäler sig själv och visas på nätverket.

Dock kvarstår problemet med att sätta rättigheter för resursen. Det är ju inte säkert att alla användare ska kunna komma åt den tillagda resursen utan rättigheterna kanske ska begränsas till vissa arbetsgrupper eller användare. Eftersom Jini omfattar alla typer av resurser, både mjukvara och hårdvara så är det svårt att ge en entydig lösning på problemet. Hela idén bakom Jini är just att resurser ska läggas till och direkt kunna användas i nätverket utan några ansträngningar. Självklart kan speciallösningar byggas för respektive resurs, t.ex. att resursen kräver lösenord eller liknande vid användning men detta är inte så flexibelt.

Jag har heller inte hittat något som tyder på att det finns någon slags inbyggd administration i "grundpaketet" för Jini. Det jag hittat är att det går att ange vid programmeringen av en resurs att resursen är "administrerbar" och med hjälp av detta kan det tillhörande administrationsverktyget erhållas. Detta verktyg är dock meningen att vara någon typ av Javaprogram som skrivs separat.

Utifrån detta så anser jag att Jini i sitt grundutförande inte stöder administration för att sätta rättigheter för resurser. Dock kan självklart t.ex. ett Javaprogram skrivas som sköter detta men det blir då ett separat program.

Exempel på ett administrationsprogram finns dock redan, *Java Dynamic Management Kit* (JDMK). Det är framtaget av Sun, är skrivet i Java och har i sin senaste version stöd för Jini (Sun Microsystems 1999c). Detta verktyg är väldigt nytt på marknaden (maj 1999) och därför har jag ej hunnit studera detta närmare.

Tanken med JDMK är att administrationen av dynamiska och flexibla nätverk ska förenklas avsevärt. För att underlätta för systemadministratörer så ska mycket av arbetet skötas automatiskt av sk ”smarta agenter”. Dessa är små Javaprogram som kan kommunicera med de nya resurser som läggs till och göra inställningar såsom att ställa in rättigheter. Varje resurs kan ha ett eget administrationsprogram som agenten kommunicerar med. Om resursen ska ställas in manuellt kan administratören kontakta den inbyggda administrationsmodulen på en mängd olika sätt, t.ex. genom en webläsare eller ett Javaprogram. Inställningar kan sedan göras för resursen innan resten av nätverket får tillgång till resursen. För att automatisera förfarandet kan en mängd olika administrationsprofiler skapas. Varje profil kan vara anpassad för en viss typ av resurs och gör då inställningar automatiskt.

Hela denna idén med automatisk och dynamisk administration låter väldigt bra anser jag men kan jag inte avgöra hur detta fungerar i praktiken. Dock verkar det vara så att ett separat administrationsprogram är ett krav för att kunna använda Jini i större nätverk

Problem 4

Vid initiering av nätverksresurs i form av mobil dator, måste tilldelning av rättigheter för användare utföras.

När en mobil användare besöker ett nätverk så leder detta idag till en rad problem. Förutom de problem som uppstår med drivrutiner och konfigurationar för att kunna använda nätverkets resurser (se Problem 1, ovan), så måste användaren registreras och tilldelas rättigheter för att kunna använda nätverket. Detta blir problematiskt om antalet besökare och frekvensen för detta ökar kraftigt.

Enligt Sun (Sun Microsystems, 1999a) så kan detta problem lösas med Jini på följande sätt:

Att få tillgång till resurser för en besökare är egentligen inga problem om Jini används. Det enda som behövs är det speciella klientprogrammet som kan visa vilka Jini-tjänster som finns i nätverket. Detta program brukar kallas ”Jini-browser” och kan utformas på en mängd olika sätt. När besökaren öppnar sin Jini-browser kommer denna att utföra ett anrop på hela nätverket och leta efter Jini-servrar som innehåller information om nätverkets resurser. Dessa presenteras sedan för användaren.

Problemet är just att begränsa tillgången för obehöriga. I litteraturen är information om detta dock väldigt begränsad. Istället talas det nästan enbart om hur lättillgängligt alla resurser ska vara för alla användare. Som jag beskrev i Problem 3, ovan, så anser jag att detta är meningen att skötas med ett separat administrationsprogram som JDMK.

Med ett sådant program så skulle detta problem kunna lösas relativt lätt. När en besökare kopplar upp sig mot ”lookup-registret” så kan detta upptäckas och genom ett

enkelt inloggningsförfarande kan besökaren få rättigheter som "gäst". Om det finns en speciell profil för gäster så kan endast tillåtna resurser skickas som svar tillbaka till besökaren och visas då i dennes Jini-browser. Via smarta agenter kan detta skötas helt automatiskt och utan att systemadministratören behöver lägga ner någon tid.

Liksom i föregående problem så anser jag inte att Jini ger fullt stöd för att lösa detta problem utan behöver ett separat administrationsprogram. Dessutom kan detta problem lösas även idag med sk loginscripts som är program som exekveras vid inloggning. Genom att skapa speciella loginscripts för olika ändamål kan användningen av resurserna styras. Dock finns ofta inget direkt behov i de flesta nätverk idag eftersom nätverken fortfarande är relativt stabila.

Problem 5

Vid mer flexibla resurser och datoranvändare, leder detta till mycket administrations- och uppdateringsproblem

Om kravet på flexibilitet i nätverken höjs så ansåg alla intervjuade systemadministratörer att detta skulle leda till problem av olika slag. Att lägga till resurser och sedan ta bort dessa med jämna mellanrum leder till att mycket resurser måste läggas ned och datorerna blir svåra att hålla uppdaterade.

För detta problem så hänvisar jag till föregående problem, ovan. När det gäller tillägg och borttagning av resurser samt enkel åtkomst så anser jag att Jini löser detta på ett bra sätt. Dock behövs ett administrativt redskap som tillägg.

Problem 6

Vid mer heterogen miljö, ökar kraven på standardisering av protokoll och standarder.

Ett problem som ständigt funnits med datorer är att olika plattformar inte är kompatibla med varandra, dvs de har olika standarder och mjukvara måste anpassas för olika plattformar. En av mina teser i denna rapport är dock att nätverken i framtiden måste erbjuda stöd för en mer heterogen miljö (se kapitel 2.2 - Resurshantering i nätverk - nutidens trender). Vid intervjuerna så ansåg alla systemadministratörer att detta skulle leda till stora problem i framtiden.

Enligt Sun (Sun Microsystems, 1999a) så kan detta problem lösas med Jini på följande sätt:

En av huvudtankarna med Java är att koden ska vara plattformsoberoende och ska enkelt kunna flyttas mellan olika datorbaserade resurser. Eftersom Jini är helt baserad på Java så gäller detta självklart för Jini också. Alla resurser som ska integreras i ett Jini-nätverk ska alltså kunna förstå Java för att kommunikationen ska fungera.

När det gäller protokoll för nätverk så baseras Jini på protokoll som idag är standard, såsom TCP/IP och HTTP. En Jini-server är inget annat än en HTTP-server men fördelen är att alla som kan "tala" HTTP kan hämta information från servern. Den fysiska kopplingen till nätverket ska enligt Sun inte heller spela någon roll. Det kan t.ex. vara traditionell koppartråd, radiovågor, infrarött ljus mm.

Tanken med Javas plattformsoberoende är mycket tilltalande anser jag. Dock dröjer det nog länge innan alla nätverksresurser kommunicerar via Java. Jag anser dock att det finns god chans att Java ska bli en standard eftersom det idag inte finns så många alternativ som möjliggör plattformsoberoende. Att Jini baseras på nätverksstandarder

som idag är i princip helt dominerande gör att Jini har en bra möjlighet att slå igenom.

Det är svårt att avgöra om Jini löser problemet med heterogena nätverksmiljöer men under förutsättning att Java blir den standard som Sun tror så anser jag att Jini löser detta problem.

6.3 Jini i praktiken

Denna del redovisar och analyserar arbetet som genomfördes under fallstudien.

6.3.1 Sammanställning av fallstudien

På grund av den korta tid som kunde ägnas åt fallstudien så gick det inte att få en klar uppfattning om hur Jini fungerar i praktiken. Att bara studera en liten del av tekniken ger också en alldeles för dålig av Jini i sin helhet. De mindre exempel som jag utförde fungerade som de skulle men det går ändå inte att göra en kvalificerad bedömning.

Fallstudiens inriktning blev istället en fråga om att undersöka hur tekniken är att använda praktiskt och om det var svårt eller lätt att lära sig. Dock så gäller detta självklart grundläggande delar av Jini men det brukar ju allmänt sett vara svårast i själva inledningsskedet.

Fallstudien resulterade inte i något material som kan sammanställas utan mer av erfarenheter. Slutsatser från fallstudien tas upp under kapitel 7.

7 Slutsatser

I detta kapitel redovisas de resultat som tagits fram i genomförandedelen. Dessa relateras även till den problemställning som ligger till grund för denna rapport. Även detta kapitel är uppdelat i de delproblem som problemställningen är uppdelad i och för varje del redovisas det aktuella resultatet.

7.1 Vanliga problem vid hantering av nätverksresurser

Som beskrevs i kapitel 6.1.2, så delades intervju svaren in i kategorier för att enklare kunna jämföras och för att enklare kunna ta fram det vanligaste problemen. Utifrån denna kategorisering så togs följande problem fram som de vanligaste:

1. Vid initiering av en nätverksresurs, behöver drivrutiner läggas till och konfigurationer behöver göras på varje klientmaskin.
2. Vid borttagning av nätverksresurs, ligger drivrutiner och länkar till resurser kvar på klientmaskinerna.
3. Vid åtkomst av nätverksresurs, måste resursen registreras och rättigheter för resursen måste sättas.
4. Vid initiering av nätverksresurs i form av mobil dator, måste tilldelning av rättigheter för användare utföras.
5. Vid mer flexibla resurser och datoranvändare, leder detta till mycket administrations- och uppdateringsproblem
6. Vid mer heterogen miljö, ökar kraven på standardisering av protokoll och standarder.

I min problemprecisering så var målet med denna del att identifiera några av de problem som finns idag vid hantering av nätverksresurser och vilka problem som kan tänkas uppkomma om miljön blir mer flexibel och heterogen. Jag anser därför att jag uppfyllt detta mål.

De slutsatser som kan dras från denna del är att det finns en hel del problem när det gäller resurshantering i nätverk. Detta trots att avgränsningarna för detta arbete var relativt stora. I min problemställning hade jag även en avgränsning att inte titta på administration av resurserna men detta blev väldigt svårt att bortse ifrån eftersom detta var ett av de största problemen.

En av de viktigaste slutsatserna anser jag vara att problemen kommer öka om nätverksmiljön blir mer flexibel och heterogen vilket är min grundtes för detta arbete. Det kommer alltså ställas högre krav på nätverken och de tekniker som nätverksarkitekturen bygger på.

7.2 Problem som kan lösas med Jini och hur detta kan ske

Med utgångspunkt från de problem som togs fram i första delproblemet så var nästa steg att undersöka hur Jini eventuellt kan lösa dessa problem och beskriva hur detta kan ske. Från början var tanken att först beskriva Jini mer översiktligt och ta upp dess problemlösningsförmåga mer allmänt. Eftersom antalet problem blev relativt stort så ändrades detta och undersökningen fokuserades bara runt dessa problem istället.

För varje problem så undersöktes Jini genom litteraturstudier och nedan presenteras resultaten för varje problem.

Problem 1 - Vid initiering av en nätverksresurs, behöver drivrutiner läggas till och konfigureringar behöver göras på varje klientmaskin.

Enligt den litteratur som jag undersökt så kan Jini lösa detta problem på ett enkelt och smidigt sätt. Detta sker genom att drivrutinen skickas över till användaren när den behövs. Vid initiering av en nätverksresurs behövs alltså inte drivrutiner läggas in eller konfigureringar göras för varje klientmaskin om Jini används.

Självklart så krävs vissa förutsättningar för att detta ska ske. Java är grunden för Jini och därför måste användaren av resursen förstå Java. Resursen själv måste också vara "Jini-kompatibel" vilket innebär att den måste förses med Java- och Jini-kod samt ha minne och en Java Virtual Machine. Av naturliga skäl så finns inte många sådana produkter på marknaden men i slutet av 1999 bör en rad produkter finnas.

Under ovanstående förutsättningar anser jag dock att Jini löser detta problem.

Problem 2 - Vid borttagning av nätverksresurs, ligger drivrutiner och länkar till resurser kvar på klientmaskinerna.

På samma sätt som drivrutiner skickas över till användaren när de behövs, så tas också drivrutiner bort när de inte används. På detta sätt kan systemen för användare av resurser hållas "rena" och inget behov finns att t.ex. rensa klientmaskiner med jämna mellanrum. Även länkar till resurser tas bort automatiskt om en resurs tas bort från nätverket. Detta sköts med den mekanism som finns inbyggd i Jini där varje resurs får ansöka om mer tid vid jämna mellanrum. Om ingen ansökan kommer in tas resursen bort från registret som håller reda på resurserna.

Under liknande förutsättningar som i problem 1, så anser jag att Jini löser detta problem.

Problem 3 - Vid åtkomst av nätverksresurs, måste resursen registreras och rättigheter för resursen måste sättas.

När det gäller registrering av resurser för att användare ska komma åt dem så är enligt litteraturen detta en av grundtankarna med Jini. Den mekanism som automatiskt registrerar resursen ger det dynamiska nätverk som till stor del saknas idag. På detta sätt kan Jini förenkla och i princip ta bort arbetet med att registrera resurser som systemadministratören eller en användare lägger till. Resursen anmäler sig själv och visas på nätverket.

När det gäller att sätta rättigheter på resurser så är detta inte lika utförligt beskrivet i litteraturen. Att sätta rättigheter på en resurs kan lösas på olika sätt men detta är något som ingår i Javas säkerhetsmodell och kan därför användas av Jini. Problemet är dock att administrera detta på ett smidigt sätt, speciellt om nätverksmiljön blir mer flexibel. I detta fall anser jag att det krävs ett separat program som kan sköta detta och ett exempel på ett sådant program är JDMK (Java Dynamic Management Kit).

Detta problem anser jag kan lösas med vissa modifikationer. Registrering av resurser kan lösas helt av Jini medan administration av att sätta rättigheter för resurser kräver ett fristående program.

Problem 4 - Vid initiering av nätverksresurs i form av mobil dator, måste tilldelning av rättigheter för användare utföras.

Detta problem är likartat föregående problem. Med tanke på att tekniken är så ung så är det kanske inte så konstigt att alla detaljer inte är uttänkta. Istället finns en rad visioner som beskriver hur det skulle kunna gå till i framtiden. I den litteratur jag undersökt så har jag dock inte hittat något konkret exempel på hur användare som besöker nätverk ska kunna begränsas i sin åtkomst till resurserna. Dock så finns många exempel på hur enkelt en besökare kan komma till alla resurser snabbt och enkelt.

Ett sätt att lösa detta är att genom ett inloggningsförfarande som sedan begränsar åtkomsten till de resurser som finns registrerade i nätverket. Att administrera detta på ett smidigt sätt kräver dock ett separat administrationsprogram.

Liksom i föregående problem så anser jag inte att Jini ger fullt stöd för att lösa detta problem utan behöver ett separat administrationsprogram eller någon form av modifikation.

Problem 5 - Vid mer flexibla resurser och datoranvändare, leder detta till mycket administrations- och uppdateringsproblem

Om resurser och användare frekvent ska läggas till och tas bort så skulle detta med dagens teknik leda till stora problem. I sitt grundutförande så löser Jini problemet med att lägga till och ta bort resurser från nätverket (se problem 1 och 2). När det gäller de administrativa problemen så hänvisas till problem 3 och 4 ovan.

När det gäller tillägg och borttagning av resurser samt enkel åtkomst så anser jag att Jini löser detta på ett bra sätt. Dock behövs ett administrativt redskap som tillägg.

Problem 6 - Vid mer heterogen miljö, ökar kraven på standardisering av protokoll och standarder.

Att Jini bygger på Java, vars grundtankar är plattformsoberoende och därmed standardisering, så talar detta för att Jini kan vara en bra teknik vid mer heterogena miljöer. Att Jini också bygger på standarder inom nätverksteknik talar också för att den har potential att lyckas.

Jag anser därför att Jini till stora delar löser de problem som uppstår vid heterogena miljöer. Detta under förutsättning att tekniken får stor spridning och att en allmän standardisering sker när det gäller nätverk och nätverksresurser.

7.3 Jini i praktiken

På grund av tidsbrist så blev denna del inte så omfattande som var önskvärt. De erfarenheter som togs fram anser jag dock vara viktiga och lärorika. De erfarenheter som togs fram sammanfattas nedan:

- Den viktigaste erfarenheten är att Jini bara är en liten utökning av Javaspråket och ska därför behandlas på samma sätt. För att kunna använda Jini så krävs grundkunskaper i Java och relativt bred sådan kunskap. Grundtänkandet i Java

finns på samma sätt i Jini och det går inte att separera dessa delar. Om Jini används fullt ut så krävs kunskap om bland annat nätverksöverföring, säkerhet och grafiska användargränssnitt vilket till stora delar finns i Javas grundpaket. Att lära sig Jini handlar mer om att lära sig förstå hur de nya koddelarna fungerar och hur dessa delar kan kombineras med Java. Detta gör att Jini är svårt eller omöjligt att lära sig utan riktiga kunskaper i Java vilket kan vara lätt att missförstå i början.

- En annan viktig erfarenhet är att titta på exempel eller implementationer som andra gjort om det finns tillgängligt. Detta anser jag är det absolut bästa sättet att lära sig en ny teknik. Att studera klasspaketet med Jini samt de tekniska specifikationerna som finns är ett svårt sätt att lära sig på enligt mig.
- Något som talar mot föregående erfarenhet är dock att det är svårt att få tag i bra exempelkod för Jini i dagsläget. Sun själva har i dagsläget väldigt lite kodexempel på sin webbplats men jag antar att detta kommer öka framöver.
- En stor hjälp för den som vill börja med Jini är att det finns sk ”mailing-listor” där Jini-utvecklare byter erfarenheter med varandra via e-post. På detta sätt kan en fråga skickas ut och om man har tur så kan svar erhållas inom en timme eller åtminstone under dagen.

7.4 Sammanfattande slutsatser

I dagens nätverksmiljöer finns en rad problem men många av dessa uppfattas kanske inte som problem förrän en bättre lösning presenterats. Många av de resurskrävande åtgärder som idag krävs kanske kan tas bort när ny teknik som t.ex. Jini börjar användas istället.

Jini kan lösa några av de åtgärder som idag kräver mycket resurser på ett snabbt och smidigt sätt. Jag anser också att en teknik som Jini eller liknande behövs i framtiden då kraven på nätverkens flexibilitet och heterogenitet kommer öka. Jini i sitt grundutförande är dock inte gjort för att administrera resurserna när det gäller t.ex. att sätta och tilldela rättigheter, utan kräver då ett separat administrationsprogram.

Att besvara min huvudsakliga problemställning, om det är vision eller verklighet att Jini förenklar integration av nätverksresurser, är inte helt lätt. I dagsläget är det mycket visioner som presenteras och det är svårt att bedöma när eller om dessa realiseras. Jini är dock något som är verklighet i den mening att produkten är utgiven och kan börja användas. Den naturliga trögheten som finns vid förändringar kommer dock att försena utvecklingen av produkter som använder Jini. Det är först när olika typer av Jini-baserad hårdvara kommer ut på marknaden som frågan kan besvaras och detta beräknas bli i slutet av 1999 eller början av år 2000.

8 Diskussion

I detta kapitel diskuterar jag mina resultat och mina erfarenheter av arbetet. Jag försöker även bedöma vad detta arbete kan tillföra forskningen och på vilket sätt arbetet skulle kunna drivas vidare.

8.1 Allmän diskussion

Från början var min tanke att enbart undersöka Jini och hur Jini förhåller sig i jämförelse med dagens tekniker. Detta hade då blivit en mer ingående och mer teknisk inriktad undersökning. Dock så tyckte jag att det också vore intressant att titta på bakgrunden till att en teknik som Jini har tagits fram. Jag valde då att undersöka vilka problem som finns idag och vilka problem som kan tänkas att uppkomma i framtidens nätverksmiljöer. Även detta kunde varit nog för en separat undersökning som då hade blivit betydligt mer ingående. Frågan är dock hur resultaten hade skilt sig i jämförelse med den undersökning som genomfördes i detta arbete.

Jag valde i ett tidigt skede att bryta ned min problemställning i mindre delar och detta anser jag har lett till att arbetet blivit mer strukturerat. Det gav även en naturlig ”röd tråd” genom arbetet eftersom efterföljande del bygger på den föregående. Dock blev sista delen, fallstudien, inte så utförlig som det var tänkt.

Det är svårt att göra en bedömning av de resultat som tagits fram. Validering av resultaten kunde gjorts på många sätt vilket jag också tog upp under metoddelen. Tiden räckte dock inte till för att göra detta vilket gör att mina resultat kanske inte har den trovärdighet som är önskvärd. Det hade också varit önskvärt att undersöka Jini med utgångspunkt från flera typer av källor. I detta fallet togs den mesta litteraturen från tillverkaren själv, vilket gör att den behöver valideras med andra källor. Detta var dock problemet med en så pass ung teknik, litteraturen var starkt begränsad.

Ett alternativ kunde varit att inte utföra den praktiska delen av arbetet. Den veckan kunde lagts på de två första delarna istället och på så sätt förbättrat och fördjupat detta. Trots det så är det inget jag ångrar eftersom jag anser det är viktigt att kombinera teoretiska delar med praktiskt arbete.

8.2 Erfarenheter

Arbetet har gett några bra erfarenheter som är användbara i framtiden. Den viktigaste erfarenheten är att den planerade tiden så gott som alltid skiljer sig från den verkliga. Detta gör att man tvingas göra avväganden och begränsa vissa delar av arbetet trots att resultatet inte blir lika bra som det var tänkt.

Problemet med att undersöka ny teknik är att den tillgängliga litteraturen inte alltid är tillräckligt utförlig och bygger mycket på visioner. Detta gör att det är svårt att bedöma tillförlitligheten. En risk är också att den blir väldigt partisk om informationen nästan bara finns tillgänglig från de som tagit fram tekniken.

En annan erfarenhet är att vid intervjuer så måste intervjupersonerna kontaktas i tid. Speciellt verkar detta gälla i databranschen där alla verkar vara mer eller mindre stressade hela tiden.

8.3 Vad detta arbete tillfört

Min förhoppning är att detta arbete ska ge en inblick i vilka problem som finns idag när det gäller resurshantering i nätverk. Eftersom min tes är att nätverksmiljön kommer bli mer flexibel och heterogen så hoppas jag också att detta leder till att fler tekniker tas fram för att åtgärda de stundande problemen som jag tror kommer.

För en organisation kan det vara viktigt att åskådliggöra vilka problem som finns med den nätverksstruktur som används idag. Speciellt intressant blir det om organisationen har planer på att försöka göra nätverket mer flexibelt och heterogent. Då kan detta arbete påvisa de problem som finns idag och vilka som kan uppstå vid förändringar.

Jag har också belyst en av de tekniker som nu är under utveckling. Detta kan vara intressant för organisationer som bestämt sig för att undersöka alternativ till de tekniker som finns idag. Detta bör också vara mycket intressant för systemadministratörer och nätverkstekniker som ställs inför liknande problem kanske dagligen. För dessa arbetsgrupper kan arbetet visa hur de dagliga problemen skulle kunna minskas eller tas bort om ny teknik användes.

Den sista delen av arbetet, Jini i praktiken, kan vara intressant för personer som tänkt att börja använda Jini praktiskt. Åtminstone kan en indikation ges på vad som krävs för att börja använda tekniken.

8.4 Förslag till fortsatt arbete

Jag skulle tycka det var intressant om den praktiska delen i detta arbete utvecklades och fördjupades. Detta vore dock mest intressant om ungefär ett år då tekniken är mer utvecklad och spridd. Ett förslag till fortsatt arbete kan då vara att jämföra hur mycket resurser som kan sparas vid en jämförelse med traditionell teknik som med största sannolikhet finns kvar då i hög utsträckning.

Beroende på utvecklingen av andra konkurrerande tekniker så vore det också intressant att jämföra Jini med Microsofts Universal Plug'n Play, Lucent's Infinospaces eller någon annan liknande teknik. Exempel på aspekter vid jämförelser kan vara flexibilitet, stöd för heterogena miljöer, hur lätt tekniken är att implementera mm.

Referenser

- Danielsson, L., 1998. Javasofts kickstart ska tysta kritiken, *Computer Sweden*. Nr 10.
- Eriksson, H-E., 1997. *Programutveckling med Java*. Lund: Studentlitteratur.
- Eriksson, S., 1978. *Statistisk undersökningsmetodik*. Lund: Studentlitteratur.
- IDG News, 1999. Handdatorer skapar kaos i USA, översatt i *Computer Sweden*. Nr 29.
- Lotsson, A., 1998. Jini gör nätet till en stor dator, *Computer Sweden*. Nr 85.
- Lucent Technologies Inc., 1998. *THE FACE OF THE NETWORKING LANDSCAPE HAS JUST CHANGED*. Tillgänglig på: <http://www.lucent-inferno.com/> (1999-03-18).
- Melin, J., 1998. Sysadmin - 4 program för koll på nätet, *Nätverk & Kommunikation*. Nr 2.
- Microsoft Corporation, 1999. *Computing Everywhere, Connecting Everything...* Tillgänglig på: <http://www.microsoft.com/homenet> (1999-03-18).
- Nilsson, B., 1998. Nätverk hemma hett på Comdex, *Computer Sweden*. Nr 99.
- Patel, R., Davidson, B., 1994. *Forskningsmetodikens grunder*. Lund: Studentlitteratur.
- Sridharan, P., 1997. *Advanced Java networking*. New Jersey: Prentice-Hall Inc.
- Sun Microsystems AB., 1998a. *Språket som vänder upp och ner på invanda begrepp i datavärlden*. Tillgänglig på: <http://www.sun.se/java/10skal.html> (1999-03-10).
- Sun Microsystems AB., 1998b. *Jini Technology: Strategy, Demonstration and Tech. Overview*. Videokassett. Oktober.
- Sun Microsystems Inc., 1999a. *JiniTM Connection Technology*. Tillgänglig på: <http://www.java.sun.com/products/jini> (1999-02-29).
- Sun Microsystems Inc., 1999b. *JiniTM Connection Technology Executive Overview*. Tillgänglig på: <http://www.sun.com/jini/overview/> (1999-03-05).
- Venners, B., 1998. Jini technology, out of the box, *JavaWorld*. December
- Wallström, M., 1999. Kraftig ökning för handdatorer, *Computer Sweden*. Nr 16.
- Åsblom, J., 1998. Gemensam plattform för handdatorer, *Computer Sweden*. Nr 13.
- Åsblom, J., 1999. Pc säljs enbart med Windows, *Computer Sweden*. Nr 6.

Intervjuer - Systemadministratörer

Syfte: Identifiera problem som uppstår vid initiering, åtkomst och borttagning av resurser i ett nätverk.

Definitioner: *Problem* behöver inte vara konkreta problemsituationer utan kan även vara arbete som krävs utöver det ordinarie. *Resurser* är i första hand hårdvara, t.ex. skrivare, stationär eller bärbar dator, scanner mm.

Avgränsning: Problemen ska ej beröra filformat mellan olika system. Fysiska begränsningar när det gäller uppkoppling till nätet är heller ej intressant.

Instruktioner: Preciserat problemet så noggrant som möjligt och försök dela upp i mindre delar.

Scenario 1

En ny resurs ska läggas till i nätverket och alla datorer ska ha kunna använda denna.

Scenario 2

En resurs ska tas bort permanent.

Scenario 3

En användare med bärbar dator, baserad på samma plattform/operativsystem som redan finns i nätverket, ska kunna komma åt ett antal resurser. Användaren har aldrig tidigare besökt nätverket.

Scenario 4

En rad nya produkter tas fram som kan kopplas in i ett nätverk. Många användare tar med sig dessa resurser och utnyttjar temporärt i det stationära nätverket. Förutom ägaren till resursen så ska fler användare i det stationära nätverket kunna utnyttja resursen.

Scenario 5

I en snar framtid har en stor del av användarna bärbara datorer eller handdatorer. Många av dessa baseras på plattformar/operativsystem annorlunda från de som finns i det stationära nätverket (t.ex. Linux, PalmOS, Windows CE, Epoc, Macintosh). När dessa användare besöker nätverket ska de komma åt samtliga resurser som de har rättigheter till och de ska kunna utnyttja tjänster hos varandra vid möten.

Namn:

Företag/organisation:

Scenario:

Problembeskrivning:

.....
.....
.....

Hur allvarligt är detta problem (sätt kryss)?

Mindre allvarligt Mycket allvarligt

Hur resurskrävande är detta problem (sätt kryss)?

Mindre resurskrävande Mycket resurskrävande

Bakomliggande orsak till problemet:

.....
.....
.....