

## Att utveckla beslutstöd ur befintliga system

Kan databaser i daglig-rutin-system användas för beslutstöd och hur skall det gå till?

Magisteruppsats 20 p (IA7400), framlagd vid  
Göteborgs Universitet, Institutionen för Informatik,  
våren 2000 av  
Birgitta Landberg

Handledare: Agneta Ranerup, Göteborgs Universitet  
Robert Johansson, Ziphius IT

### Sammanfattning

Många datoriserade system samlar in och lagrar stora mängder data. I dessa datamängder kan finnas en dold kunskap. Samtidigt finns i en organisation ett behov av beslutstöd, som man ofta försöker tillfredsställa med hjälp av egna uppföljningar och Excel-ark. Den här uppsatsen har tagit ett underhållssystem för stora tekniska anläggningar i en kommun som utgångspunkt för att prova om databasen i ett daglig-rutin-system rymmer dold information och om användarna själva har tydliga frågor som skulle kunna besvaras om man tog fram den dolda informationen. Ett delsyfte har varit att ta fram en metod att visa diagram på intranätet med hjälp av en komponent och Excel-diagram. Ett annat delsyfte har varit att kartlägga användarnas inställning till ett beslutstöd och vilka faktorer som påverkar deras inställning.

Eftersom en tyngdpunkt har legat på användarnas medverkan har intervjuerna spelat en central roll. Litteraturstudier, kunskapsutvinning ur databasen och programmering var del av metoden.

Resultatet visar att användarna har en klar uppfattning om vad de vill ha svar på. Deras önskemål rör sig i stor utsträckning om frågor på en operationell nivå. De har också en bestämd uppfattning om hur svaren skall ges – ”det skall vara en knapp att trycka på”. I stor utsträckning finns svaren på deras frågor i databasen. Det praktiska resultatet blev en ”knapp” på företagets intranät, en ActiveX-komponent för grafik och en rad diagram för snabba översikter över verksamheten. Beslutstödsmodulen utformades så att den ger information såväl i sammanfattande form som i detaljerad. Detta betyder att ett och samma beslutstöd tillfredsställer informationsbehoven såväl hos personal på ledningsnivå som personal på operationell nivå. Den viktigaste faktorn som påverkade användarnas inställning till ett beslutstöd var en viss tveksamhet inför någon ny, förväntat komplicerad, programvara.

## Förord

Det finns många jag skulle vilja tacka för deras stöd under min studietid i allmänhet och uppsatsarbetet i synnerhet.

Agneta för konstruktiva och konkreta tips och kommentarer. Robert för stöd i det praktiska arbetet med databas och programmering. Håkan för moraliskt stöd. Mina föräldrar och syskon för att de orkat lyssna och orubbligt stått bakom mig, liksom fondförvaltaren i reklamen.

Men framför allt och framför alla vill jag framföra mitt tack till mina barn, Patrik och Jonas (12 och 10 år), som har stått ut med mig hela denna tid. De har kommit med glada tillrop och tröstande ord, och firat glädjestunderna med mig. Vid något tillfälle fick jag ett stillsamt och vänligt förslag: ”mamma, i kväll ska du nog lägga dig tidigt, för du ÄR rätt grinig....”. Utan en sådan förståelse från mina allra närmaste hade arbetet blivit bra tungt.

---

## Innehållsförteckning

|  |   |
|--|---|
| <b>1. INLEDNING</b>  | <b>5</b>                                  |
| 1.1. Bakgrund  | 5   |
| 1.2. Beslutstöd  | 5   |
| 1.3. Kunskapsutvinning   | 6   |
| 1.4. Praktisk relevans   | 7   |
| 1.5. Teoretisk relevans  | 8   |
| 1.6. Utvärdering av användarönskemål                               | <b>Fel! Bokmärket är inte definierat.</b> |
| 1.7. Disposition   | 9   |
| <br>   |   |
| <b>2. SYFTE OCH FRÅGESTÄLLNING</b>                                 | <b>10</b>                                 |
| <br>   |   |
| <b>3. METOD</b>  | <b>11</b>                                 |
| 3.1. Allmänt kring metodval och inriktning                         | 11  |
| 3.2. Litteraturstudier   | 12  |
| 3.3. Etnografi   | 12  |
| 3.4. Intervjuer  | 12  |
| 3.5. Avgränsning   | 13  |
| 3.6. Kunskapsutvinning ur databasen                                | 14  |
| 3.7. Utvärdering och sammanställning av intervjuerna               | 14  |
| 3.8. Rapportförslag  | 14  |
| 3.9. Programmering   | 15  |
| <br>   |   |
| <b>4. TEORI</b>  | <b>18</b>                                 |
| 4.1. Beslut  | 18  |
| 4.2. Kunskap   | 18  |
| 4.3. Knowledge Discovery in Databases (KDD)                        | 19  |
| 4.4. Decision Support System (DSS)                                 | 19  |
| 4.5. Beslutstödet utseende   | 24  |
| 4.6. Programmeringsbegrepp   | 24  |
| <br>   |   |
| <b>5. UNDERHÅLLSSYSTEMET SIRI</b>                                  | <b>26</b>                                 |
| 5.1. Kort kring en delarbetsorders liv                             | 26  |
| 5.2. Historik  | 26  |
| 5.3. Anläggningsstruktur   | 27  |
| <br>   |   |
| <b>6. RESULTAT</b>   | <b>29</b>                                 |
| 6.1. Faktorer som påverkar användarnas inställning till beslutstöd | 29  |
| 6.2. Informanternas förslag på rapporter                           | 30  |
| 6.3. Lösningsförslag   | 39  |
| 6.4. Respons   | 45  |

|   |           |
|---|-----------|
| 6.5. Rapportval   | 45        |
| 6.6. Rapporterna  | 46        |
| <b>7. DISKUSSION</b>  | <b>50</b> |
| 7.1. Beslutsödets innehåll  | 50        |
| 7.2. Beslutsödets utseende  | 51        |
| 7.3. Beslutsödets resultat - information                          | 52        |
| 7.4. Beslutsödets underlag - databasfrågor                        | 53        |
| 7.5. Faktorer som påverkar användarnas inställning till beslutsöd | 53        |
| 7.6. Har användarna tydliga frågor?                               | 54        |
| 7.7. Svårigheter i intervjusituationen                            | 57        |
| 7.8. Komponenter  | 58        |
| 7.9. Varthän...   | 58        |
| 7.10. På annat sätt   | 58        |
| <b>8. SLUTSATS</b>  | <b>59</b> |
| <b>9. FÖRKORTNINGAR</b>   | <b>59</b> |
| <b>10. REFERENSER</b>   | <b>59</b> |

## Bilagor

1. Intervjufrågor

## 1. Inledning

### 1.1. Bakgrund

Det finns oräkneliga datoriserade system i vårt samhälle. Många av dem används i dagliga rutiner och samlar in och lagrar stora mängder data. Ofta förblir dessa stora datamängder bara en oöverskådlig mängd av enskilda data, trots att här kan finnas en dold kunskap. Dessa datamängder kan utgöra en potential för bättre och effektivare rutiner, bättre underbyggda beslut eller säkrare prognoser.

Exempel på sådana datainsamlade daglig-rutin-system är underhållssystem för stora tekniska anläggningar, ekonomisystem och patientjournaler inom sjuk- eller tandvård.

Värdet av att lagra sådana enormt stora mängder data är beroende av hur man förmår använda datan. Det är i många fall viktigare att kunna ställa rätt frågor och formulera problemen väl än att optimera algoritmer (Fayyad, Piatetsky-Shapira & Smyth, 1996). Naturligtvis kan man ta fram en uppsjö av jämförelser och rapporter ur en stor databas, och metoden att ta fram dem kan vara briljant i sin teknik – men om resultatet av den briljanta tekniken inte är svaret på en fråga som någon faktiskt ställer sig så är arbetet förgäves.

Programvara som hanterar alla rutiner kring underhållsarbetet i kommunens stora tekniska anläggningar har under hösten 1999 implementerats hos Borås Energi AB. Programmet kallas för SiRi och är utvecklat av Bikan AB, Ljungskile. Jag har varit delaktig som programmerare i projektet. Funderingarna kring uppföljning av arbetet i systemet har naturligtvis funnits med från början av systemeringen, men under arbetets gång har vi börjat fråga oss om man kanske kunde göra mer än bara enkla uppföljningar. Bikan AB har bett mig att studera de här frågorna när det gäller SiRi.

Denna uppsats handlar om att använda sådan dold kunskap för att utveckla beslutstöd, hur interaktionen mellan användare och utvecklare fungerar i en sådan process samt hur en sådan applikation kan skapas och se ut.

### 1.2. Beslutstöd

Innehållet i denna uppsats kretsar kring begreppen beslutstöd och kunskapsutvinning. För att ett beslutstödsverktyg skall fungera krävs bland annat att det är lättåtkomligt, inte ställer alltför komplicerade krav på ytterligare information från användaren samt att det svarar på frågor som är av betydelse för användaren (Finlay, 1994). Det är därför viktigt att ta reda på vad användaren faktiskt har behov av.

För att kunskap ska uppstå ur information krävs att beslutstödet ges med hänsyn till användarens referensramar och begreppsvärld (Andersen, 1994). Det är därför viktigt att ta reda på i vilken miljö och av vilka människor beslutstödet kommer att användas.

Beslutstöd är en företeelse som kan förekomma på många olika nivåer i en organisation (Turban & Aronson, 1998). I litteraturen talar man mest om beslutstöd på ledningsnivå. Grundtanken i denna uppsats var att undersöka förutsättningarna för ett beslutstöd utan att i förväg fastslå på vilken nivå i organisationen beslutstödet skulle ges. Det kan förmodligen tyckas att ett beslutstöd på ledningsnivå får en större inverkan och bättre effekt i en organisa-

tion, än ett beslutstöd som används i produktionen/själva verksamheten. Jag hävdar dock att ett beslutstöd på operationell nivå har en stor betydelse för organisationen i stort och kan innebära en betydande förbättring av ekonomin i verksamheten. Varje beslut i sig är förmodligen ett relativt litet beslut, ekonomiskt sett, på den operationella nivån. Men det tas många sådana beslut under en enda dags arbete.

Beslutstödet på operationell nivå kan vara av ett mindre avancerat slag, frågorna att besluta kring kan vara av ett mindre komplext slag än de frågor en styrelse har att ta ställning till. Vi talar dessutom om en kortare tidsrymd för de operationella besluten.

Där de operationella besluten fattas utifrån någons erfarenhet, eller ”en känsla av” kan det vara väsentligt att på ett enkelt sätt kunna visa för alla andra, svart på vitt, att det man känner på sig kan underbyggas av fakta. Det borde även kunna snabba upp beslutsprocessen i vissa fall, där diskussion annars i onödan kan uppstå. Förhoppningsvis kan en sådan möjlighet till bekräftelse också ge en positiv känsla av ökat självförtroende för användarna, genom en bättre känsla av kontroll och ett handfast underlag som stödjer deras åsikt.

En organisations mål- och resultatområden kan ofta ses hierarkiskt, där resultatet på en lägre nivå är ett medel för att nå önskat resultat på en högre nivå (Abrahamsson & Andersen, 1996). Det borde därför vara lika viktigt för en organisation att värna om besluts kvaliteten på den lägre nivån i organisationshierarkin. Det har inte, vad jag kunnat finna i litteraturen, ägnats mycket intresse åt beslutstöd på denna nivå tidigare.

### 1.3. Kunskapsutvinning

I samband med beslutstöd nämns ofta data mining. Ett antal examensarbeten har gjorts kring data mining och data warehouse och det finns mycket litteratur i ämnet.

Det finns många verktyg för data mining och kunskapsutvinning. Många gånger krävs det att användaren är en van datoranvändare och dessutom har vissa kunskaper i statistik för att dessa verktyg skall vara användbara. Tanken med denna uppsats är i stället att åstadkomma en relativt enkel rapportering, men en rapportering med en fokusering på dolda kunskaper i databasen. Min avsikt är alltså att skapa ett beslutstöd som drar nytta av den stora mängden insamlad data och ger information som förbättrar användarnas möjlighet att fatta väl underbyggda beslut.

Denna uppsats handlar inte om det som data mining definitionsmässigt står för. Data mining extraherar ny information eller kunskap från data, men data mining verktyg gör mer än enkla analyser. Datamining sysslar med komplex, statistisk slutledning och modellbyggande för att finna mönster i stora databaser. Detta är inte föremål för studier i detta skede. Det finns många intressanta tänkbara delprojekt i arbetet med ett beslutstöd, varav de statistiska analyserna kunde vara ett, men eftersom jag är ensam i det här projektet och har begränsad tid har jag bestämt mig för att koncentrera mig på de delar som rör interaktionen mellan programmerare och användare och att i enlighet med Fayyad, Piatetsky-Shapira & Smyths uttalande (1996) försöka mig på att hitta de rätta frågorna snarare än att hitta de mer sofistikerade statistikformlerna.

#### 1.4. Praktisk relevans

Fokuseringen ligger alltså snarare på att upptäcka intressanta möjligheter och synvinklar hos en befintlig databas, att fånga upp användarnas idéer och medvetna eller omedvetna behov samt inspirera till vidare utforskning och sökning efter dold kunskap, än på att i detta skede formulera algoritmer eller använda olika verktyg för att finna sådana mönster.

Fokuseringen ligger också på att basera detta relativt enkla beslutstöd direkt på användarnas önskemål och funderingar, samt att utforma det så att inte ens en ovan tangentbordstryckare behöver känna osäkerhet.

En kunskapsutvinning ur SiRis databas skulle, enligt flera informanter, kunna betyda att man kan ägna sig åt förebyggande underhåll i stället för avhjälpande, vilket är ekonomiskt mer fördelaktigt. Med en tydliggjord kunskap om anläggningarnas löpande underhåll skulle man kunna fatta bättre investeringsbeslut och organisera personal- och arbetsinsatser på ett mer effektivt sätt. Idag kanske man blir tvungen att ha dyrbara stillestånd i anläggningarna på grund av en trasig detalj - kanske borde just den detaljen ingå i någon regelbunden rond för förebyggande underhåll. Kanske köper man idag anläggningsdelar, stora eller små, till ett lågt och bra pris och känner sig belåten med det - men kanske är det ständigt just någon av dessa objekt som orsakar stopp och stora kostnader. Kanske sitter det ett objekt av samma typ men något dyrare i ett annat hörn av anläggningen och puttrar på i det tysta.

Ett annat exempel på vad som skulle kunna göras är arbetsbelastningsrapporter för olika yrkeskategorier. Kanske skulle man inom organisationen ha intresse av att veta hur lång tid det tar från felanmälan till avrapportering, om "rätt" saker felanmäls, vem som i allmänhet gör felanmälningar, om den preliminärt angivna felorsaken i allmänhet visar sig stämma med faktisk felorsak eller om felet skulle ha kunnat förhindras genom ett förebyggande underhållsarbete.

Jag har i mina kontakter med personalen på Borås Energi upptäckt att en hel del beslutstöd av den här typen idag tas fram manuellt av var och en på avdelningarna, med hjälp av egna uppföljningar i Excel. I de fall där uppgifterna faktiskt existerar i SiRi skulle man kunna spara många mantimmar per månad på en automatiserad framtagning av underlaget.

Att man vill ha ett beslutstöd förefaller uppenbart - det är ju en form av beslutstöd man försöker skaffa sig genom egna journaler och egna uppföljningar i Excel-ark, osv. Detta är utmärkt på många sätt - bland annat skaffar sig var och en exakt den kunskap som är viktig just för honom. Men det finns problem. Man har i många fall utvecklat imponerande mallar i Excel, men det är tidsödande att ta fram information och fylla i mallarna. Tiden räcker helt enkelt inte riktigt till för den sortens arbete på den operationella nivån, så trots att man i vissa fall lägger en hel del tid på det så halkar det gärna efter och blir liggande. Detta innebär att kunskapen om ett förlopp eller en anormal kostnad eller liknande ibland tar för lång tid att få och kunskapen blir "gammal". Med en snabbare uppföljning skulle man kunna uppnå en mer effektiv hantering av de frågor som ska lösas. Kanske hinner man köpa ytterligare ett halvdu-sin pumpar av en typ som inte riktigt håller vad den lovar i något avseende - men hade man haft en snabbare uppföljning hade man vetat det och kunnat prova en ny typ.

De förväntade fördelarna med att använda beslutstöd undersöktes 1994 av Udo och Guimaraes (Turban & Aronson, 1998). De fördelar eller vinster man upptäckte var högre beslutskvalitet, förbättrad kommunikation, kostnadsminskning, ökad produktivitet, tidsbesparingar, nöjdare kunder och nöjdare personal.

Enligt Frederick Herzberg (Abrahamsson & Andersen, 1996) kan, kortfattat, trivsel på arbetet beskrivas som tillfredsställelse av behov. De behov Herzberg ser som viktiga i sammanhanget är bl a tillfredsställelse i att lösa problem, erkännande för väl utfört arbete och att arbetet i sig är varierat och intressant. Ett beslutstöd kan, om det blir det redskap det är tänkt som, därför ge gladare och nöjdare personal. Dels beroende på den insparade tiden, dels beroende på en förhoppningsvis ökad känsla av att inte bara ha kontroll över sina beslut utan också kunna visa befogenheten i de beslut man tar. Kanske kan frigjord tid användas för nya inspirerande arbetsuppgifter.

## 1.5. Teoretisk relevans

### Beslutstödet utseende

Hur ett beslutstöd rent praktiskt skall se ut för att komma till störst nytta är ytterligare en fråga man kan ställa sig. Vägvälet handlar ofta om tabeller kontra diagram, detaljerat kontra större svep. Det handlar också om att försöka begränsa de parametrar som skall väljas/matras in av användaren, och att skapa ett lättåtkomligt, lättanvänt gränssnitt. Avsikten i denna uppsats är att låta beslutstödet ingå naturligt, som "en knapp", i befintlig applikation på Borås Energis intranät.

När en enorm mängd data presenteras och antalet intryck materialet avser att ge är få är diagram att föredra framför tabeller, men om rapporten är av ett mindre komplext slag spelar det ingen roll för besluts kvaliteten eller förståelse om diagram eller tabell väljs för presentationen (Dickson, DeSanctis & McBride, 1986).

Viktigare än valet diagram/tabell är många gånger att undvika rapporter som visar en stor mängd resultat eller vill ge ett stort antal intryck i en och samma rapport. Bättre förståelse uppnås om resultatet kan visas upp i flera mindre omfattande tabeller eller diagram (Dickson, DeSanctis & McBride, 1986).

Lösningförslagen har utformats med detta i åtanke. Dock bör det vara upp till användaren (Andersen, 1994) att avgöra vilka yttre egenskaper ett system skall ha. Stor hänsyn har därför tagits till användarnas önskemål om beslutstödet utseende.

### Beslutstöd på olika nivåer i en organisation

Beslutstöd på ledningsnivå är helt uppenbart det man hittills har ägnat störst intresse, medan beslutstöd på operationell nivå nämns i litteraturen mest i förbifarten. Turban & Aronson (1998) nämner att beslut kan fattas på olika nivåer. Finlay (1994) anger organisationens nivåer som en av många möjliga metoder att kategorisera beslutstöd. Ett stort antal artiklar skrivna om beslutstöd talar endast om ledningsnivå – strategiska och taktiska beslut. Den här uppsatsen har inte haft som syfte att göra en fullständig litteraturgenomgång, men jag har ändå lagt en hel del möda på att försöka hitta något om just den operationella nivån och beslutstöd. Om mitt magra resultat beror på att området ägnats ett mycket ljummet intresse – eller inget alls – är denna uppsats kanske en liten extra pusselbit inom området.

Oavsett nivå inom organisationen bör man som systemutvecklare inse att ett datasystem påverkar människors arbetssituation (Andersen, 1994). Den norske arbetslivsforskaren Einar Thorsrud formulerade en rad grundläggande psykologiska arbetskrav. Ett av dessa är "behov



---

av att kunna fatta beslut, åtminstone inom ett avgränsat område som den enskilde kan kalla sitt eget" (Thorsrud, enl Andersen, 1994). Ett beslutstöd kan alltså ses som ett verktyg för att ge anställda, oavsett position inom organisationen, en bättre arbetssituation eftersom det underlättar möjligheten för beslutsfattande. Thorsrud talar möjligen i första hand om en sorts tillåtelse att fatta beslut, vilket naturligtvis ligger utanför beslutstödet inverkan, men även så vill jag påstå att ett bra beslutstöd kan innebära nya möjligheter att delegera beslutsfattande och därmed ge just denna tillfredsställelse åt fler.

### 1.6. Användarmedverkan

Tidiga beslutstöd blev inte mycket använda. Detta berodde dels på att beslutsfattarna hade orealistiska förväntningar på systemen, dels på att systemutvecklarna hade för lite kunskap om beslutsfattarnas arbetssituation. Systemen blev för komplexa och krävde för mycket data input, vilket innebar att de blev tidskrävande att använda. För dålig kunskap om beslutsfattarnas behov av information respektive förståelse ledde dessutom till att systemen kanske inte ens svarade på rätt frågor. (Finlay, 1994).

Det kan därför inte nog betonas att det är nödvändigt att i grunden göra allt för att försöka utvärdera vilket beslutstöd som verkligen utgör ett stöd och inte bara blir ett coolt teknikbygge från programmerarens sida.

Att involvera många användare i utvecklingsarbetet har en rad fördelar i jämförelse med att ha få användare engagerade i projektet. Några av fördelarna är att man kan få en bredare acceptans av lösningarna (eftersom de är förankrade hos många) och att man får många idéer och en större förståelse för problem och mål inom organisationen - kanske både hos systemutvecklare och användare. Nackdelarna är exempelvis att det tar tid, både för den enskilde och i betraktande av den sammanlagda utvecklingsperioden. Det blir också mer administration, kan finnas risk för konflikter och kan innebära en belastning på löpande verksamhet (Goldkuhl & Röstlinger, 1988).

Goldkuhl & Röstlinger (1988) poängterar starkt att ett aktivt deltagande föder engagemang och intresse. I denna uppsats har ett syfte varit att utvärdera vilka faktorer som påverkar användarnas inställning till beslutstödet. Att få en ökad förståelse för dessa faktorer skulle kunna ge en ökad möjlighet att engagera och involvera användarna. Utan deras engagemang är man tillbaka till de problem tidiga beslutstöd led av - systemutveckling utan förankring i de tilltänkta användarnas situation, behov och referensramar.

### 1.7. Disposition

Detta första avsnitt innehåller en bakgrund och inledning samt praktisk och teoretisk relevans för detta arbete. I kapitel 2 redogörs för vilket syfte projektet har. Metodavsnittet i kapitel 3 beskriver de metoder som använts. Teoriavsnittet i kapitel 4 ger bl a grundläggande information om begreppen Beslut, Kunskap och Beslutstöd samt förklarar mycket kort vissa programmeringsbegrepp. I kapitel 5 finns en mycket kort beskrivning av vissa kärnfunktioner i underhållssystemet SiRi.

Kapitel 6 beskriver dels resultatet av de intervjuer som gjorts, dels utformningen av det beslutstöd som projektet resulterat i, medan kapitel 7 är ett utförligare resonemang kring de resultat vi fick och vad som kan ha påverkat dem. Här finns även funderingar kring vad som kunde gjorts annorlunda och hur man skulle kunna gå vidare i ett nästa skede.

Slutsatsen i kapitel 8 är en kort sammanfattning av vad jag kommit fram till.

I kapitel 9 finns en liten lista över de förkortningar som förekommer i sammanhanget och slutligen finns referenser i kapitel 10. Som en bilaga till dokumentet finns intervjufrågorna.

## 2. Syfte och frågeställning

Frågeställningen i denna uppsats är om en stor daglig-rutin-databas innehåller användbar dold information och om denna dolda information kan användas för att tillfredsställa informationsbehovet hos organisationens personal på olika nivåer. I frågeställningen ingår också frågor kring interaktion mellan systemutvecklare och användare; hur användarna kan engageras i utvecklingen och vilka faktorer som påverkar deras inställning till projektet. Ytterligare en del av frågeställningen är huruvida det är möjligt att använda Excel för att visa diagram på intranät.

Syftet med det här arbetet är därför att studera kunskapsutvinning och beslutstöd på operationell nivå i ett verkligt sammanhang. Jag vill undersöka om det är möjligt att ta fram enkla men användbara och effektiva verktyg för beslutstöd inom en befintlig applikation genom utnyttjande av den dolda informationen i en stor databas och dessutom prova om användarna har tydliga frågor som kan besvaras på detta sätt.

För att uppnå syftet har ett underhållssystem för stora tekniska anläggningar hos en kommun använts för studier. Underhållssystemet heter SiRi.

Uppbrutet i mer detaljerad punktform kan delsyttena sägas vara att:

1. kartlägga de frågor användarna av SiRi vill ha svar på som underlag för de beslut de tar i sitt arbete och vilka åsikter de har kring ett eventuellt beslutstöd
2. undersöka vilka faktorer som påverkar användarnas inställning till ett beslutstöd
3. studera möjligheterna att finna önskade svar i befintlig databas och utforma lösningsförslag
4. programmera en komponent för diagramvisning på intranätet

### 3. Metod

Under denna rubrik beskrivs de metoder arbetet utfördes med i kronologisk ordning. Visserligen är ett sådant här arbete iterativt och metoderna går i varandra, men de stora huvuddragen kan ändå ses i viss ordning.

#### 3.1. Allmänt kring metodval och inriktning

Det har känts naturligt redan från början i detta arbete att välja en hermeneutisk inriktning. Tidigare erfarenheter har visat att man har lagt för liten vikt vid användarnas begreppsvärld och arbetsmiljö och lagt stor vikt vid invecklade algoritmer och beräkningar (Finlay, 1994). Detta har lett till att beslutstöden förmodligen har varit avancerade och kunnat visa upp rättvisande beslutsunderlag – men de har inte blivit använda eftersom de varit för komplicerade och kanske inte ens svarat på de frågor som beslutsfattaren ställt.

Mänskliga aktiviteter måste tolkas och förstås, inte beskrivas och förklaras (Mouwitz, 1997). Denna tolkningslära formulerades tydligast av Wilhelm Dilthey (1833 – 1911). Mouwitz (1997) ger en punktviss sammanfattning av läran. Några av dessa punkter kan direkt appliceras på arbetet i denna uppsats.

- *Det handlar om att förstå – inte om att förklara.*  
Jag behöver i detta arbete förstå vilket arbete mina informanter utför, vilka beslut de har att ta och hur mitt arbete skulle kunna underlätta deras.
- *Förståelsen är aldrig objektiv eller neutral, utan utgår från mig själv.*  
Min förståelse i det här fallet grundar sig på de kunskaper och den erfarenhet jag lyckats inhämta hittills genom studier och arbete.
- *Förståelse är ett växelspel mellan del och helhet.*  
Det räcker inte att förstå hur varje enskild användare arbetar med systemet eller vilka beslut han har att ta och det räcker inte att ha en tydlig bild av hur hela systemet fungerar i sitt sammanhang. Både delarna och helheten behövs.
- *Människan är både subjekt och objekt – naturvetenskap är inte portförbjuden.*  
Även om informanterna kan tänkas ha olika uppfattning om vad de behöver veta och vilken typ av presentation de vill ha så har de vissa drag gemensamt. Själva utformningen av beslutstödskomponenten bygger på sådana gemensamma drag – tydlighet och ordnad struktur på bildskärmen exempelvis.

I stället för att studera organisationen utifrån, som en positivist skulle göra, har ambitionen varit att arbeta med organisationen och dess medarbetare i ett försök att tillsammans utforma något användbart (Dahlbom & Matthiassen, 1993).

En hermeneutiker är enligt Dahlbom & Mathiassen (1993) intresserad av speciell kunskap, här och nu och tror inte på idén om generell kunskap. Jag vill dock hävda att även om detta arbete i stort är genomfört i hermeneutisk anda så finns här även en generell kunskap att hämta, nämligen den att stora databaser döljer större kunskap samt att enkla beslutstöd kan utformas efter användarnas behov och bli användbara.

### 3.2. Litteraturstudier

Litteraturstudier kring ämnena beslutstöd, kunskapsutvinning och presentationsteknik har genomförts. En utredare bör i första hand använda sig av sitt ”närbibliotek”, som består av redan inlästa handböcker och kursböcker, kunskap man har i huvudet och goda personkontakter (Eriksson & Wiedesheim-Paul, 1997). Detta sparar tid. Jag har därför sökt i min egen hylla. Sökning efter lämplig litteratur har också genomförts främst via Internet, i universitetsbibliotekens databaser samt i ACMs digitala artikeldatabas.

### 3.3. Etnografi

För att kartlägga vilka behov av beslutstöd som fanns inom organisationen använde jag en etnografisk ansats. Etnografi är en kvalitativ metod, som tar sikte på att förstå hur människor beter sig och tänker i specifika situationer, genom fältstudier. Fältstudier kan sägas bestå av intervjuer, observationer, videofilmning, interaktiva analyser (Bly, 1997 enligt Arvidsson & Johansson, 1999). Inom etnografien finns en metod som kallas ”quick-and-dirty” (Hughes et al, 1994), som är inriktad på en mindre omfattande kartläggning snarare än en fullständig förståelse av den miljö man undersöker. För denna uppsats ansågs en quick-and-dirty-ansats tillräcklig.

Några observationer i etnografisk mening genomfördes inte. De beslut som informanterna tar sker idag i huvudsak baserat på erfarenhet och känsla och ej synligt för något öga. Det är således inte ett synligt eller i tiden utbrett fenomen som går att studera.

Att etnografi har blivit en alltmer populär metod när det gäller systemutveckling beror bl a på att många system misslyckas på grund av att deras design inte tar tillräcklig hänsyn till den sociala miljö de skall användas i (Hughes et al, 1994). Man har med andra ord insett att det krävs en större kunskap om användarna och deras verklighet, referensramar och behov.

### 3.4. Intervjuer

Hela den här uppsatsen bygger på nödvändigheten av att förstå vilken information användarna upplever som viktig i sitt dagliga arbete. Det var därför en självklarhet att gå ut och fråga dem!

Det är framför allt tre avdelningar på Borås Energi som använder SiRi i sitt dagliga arbete – Projekt-, Produktions- och Distributionsavdelningen. Att välja informanter slumpmässigt hade i det här fallet varit utan mening, eftersom det är relativt få som kan tänkas ha behov av den här sortens beslutstöd och tillräcklig insikt i arbetet. Arbetsituationerna är något olika på de olika avdelningarna, varför det kändes viktigt att prata med folk från alla tre avdelningarna.

Min arbetsledare på Bikan AB tog fram en lista på 13 personer på alla tre avdelningarna. Han ville dessutom själv gärna bli intervjuad. Han känner företaget och personalen väl, efter många års samarbete. Fjorton personer att intervjuas var möjligen lite i överkant, men tanken var att det skulle finnas marginal om det skulle falla bort några på grund av svårigheter att bestämma en tid eller liknande. Listan innehöll såväl enhetschefer (ledningsnivå), gruppchefer, beredare, arbetsledare samt projektledare eftersom vi inte ville gå in i arbetet med en på förhand bestämd organisationsnivå för beslutstödet.

Något större bortfall blev det dock inte, utan tolv intervjuer genomfördes. En person var långtidssjukskriven och en hade fulltecknat schema för en lång tid framöver.

---

För att ge informanterna en möjlighet att tänka igenom frågeställningarna och förbereda dem på intervjuerna skickade deras chef ut ett epost-meddelande om att detta arbete skulle utföras. Ett kort brev från mig bifogades. Brevet innehöll endast en kort presentation av mig och uppsatsen, samt vad intervjun skulle komma att handla om. Några dagar senare ringde jag upp dem och vi bestämde en tid för samtalet.

Intervjuerna genomfördes under två dagar.

Intervjuerna hölls som relativt öppna samtal, med vissa hållpunkter snarare än efter ett bestämt frågeformulär. Stödpunkterna/frågorna finns i bilaga 1. För varje samtal hade jag satt av en timme, vilket visade sig vara lagom. Någon enstaka intervju gick på någon halvtimme, någon annan tog sin timme och lite till – de allra flesta tog en trekvart med vingelmån.

Jag ställde inga direkta frågor om hur informanterna ställde sig till ett beslutstöd, utan lät detta ingå i samtalet. På den här punkten förlitade jag mig alltså i viss utsträckning på min egen tolkning av deras sätt att uttrycka sig när det gällde beslutstödet.

Vid intervjuerna förde jag anteckningar. Någon bandspelare använde jag inte, dels för att jag bedömde att en sådan skulle upplevas som hämmande både för informanterna och mig själv, dels för att jag ansåg att det inte var viktigt med de allra minsta röstnyanserna och ordagranna citaten i det här fallet. De frågor och ämnen vi skulle diskutera var av teknisk och relativt opersonlig natur och dessutom mycket konkreta. Några tolkningssvårigheter kunde jag inte förutse.

När man intervjuar finns det en viss risk att man påverkar informanten (Easterby-Smith et al, 1991). Man kan rentav säga som Eva Lundgren (1992): "Naturligtvis påverkar jag situationen med hela min subjektiva personlighet!", och vidare "Forskaren kan inte ställa sig utanför en bestämd verklighet och definiera sig som observatör. Forskaren är automatiskt deltagare i den verklighet som skall utforskas".

Detta behöver dock inte vara en nackdel. De idéer eller förslag som jag kan tänkas tillföra kan ju resultera i nya idéer eller nya synvinklar för informanten. En sådan intervju, som snarare är ett samtal, kan ses som ett balanserat utbyte av erfarenheter och kunskap (Eriksson & Wiederheim-Paul, 1997). Det viktiga var att mina funderingar kom först sedan informanten fått säga sitt.

Intervjuerna skulle ge mig svar på tre huvudfrågor. Dels vilken information jag behövde finna i databasen för att kunna underlätta beslutsprocessen för mina informanter, dels i vilken form de ville se svaren. Dessutom ville jag känna av informanternas inställning till beslutstödet som sådant och förhoppningsvis få reda på vilka faktorer som påverkade deras inställning.

### 3.5. Avgränsning

Redan från början var vi medvetna om att det kunde finnas en stor mångfald av önskemål om rapporter/beräkningar inom SiRi. Om intervjuerna gav ett alltför brett resultat skulle en avgränsning bli nödvändig. Koncentrationen måste läggas på några frågor – projektet skulle annars riskera att bli övermäktigt för en person under en termins relativt begränsade tidsperiod.

### 3.6. Kunskapsutvinning ur databasen

Efter utvärdering av intervjuerna och med större kunskap om vilken information som var eftersökt gällde det att studera databasen för att kunna bestämma om efterfrågad information verkligen fanns i den. Det är viktigt i det läget att ta ställning till hur man bör behandla eventuella inkonsekvenser i datan och om materialet behöver ”städas” eller konverteras till beräkningsbara faktorer.

För detta användes flera olika metoder och verktyg i samarbete med varandra. I första hand sql-serverns två egna hjälpmedel, vybyggaren och SQL Query Analyzer. I vybyggaren kan man på ett enkelt grafiskt sätt bygga enkla sql-satser. När man har en sql-sats som fungerar på grundnivå kan man ta den med sig till analyzern och där bygga vidare med mer invecklade sql-satser. Ett annat sätt är att koppla en Access-databas mot tabeller i sql-servern och på det viset ta hjälp av Access-funktioner. Exempelvis testade jag en del diagramidéer på det viset. Ytterligare en variant är att köra databasfrågor från Excel, få data skrivet på Excel-arket och skapa diagram utifrån dem.

De olika sätten liknar varandra ganska mycket. Själva grävandet i databasen och utvidgandet av min förståelse för vad som finns i databasen kändes naturligt och mest givande att göra utifrån Sql-serverns egna verktyg. Här fick jag relationerna och sambanden med på köpet och befann mig närmare datamängden.

Att bygga databasfrågor i Excel eller Access är lite obekvämare, men ger ett mer direkt användbart resultat i form av data eller diagram.

### 3.7. Sammanställning av intervjuerna

Utgångspunkten för denna uppsats var bland annat att utröna vilka frågor användarna kunde tänkas vilja få besvarade i ett beslutstöd och att ta reda på huruvida dessa svar faktiskt fanns gömda i databasen.

Med hjälp av intervjuerna sammanställde jag en lista över de önskemål och frågor som informanterna visat intresse för under samtalen, samt vilka faktorer som påverkade deras inställning till beslutstödet. Med hjälp av kunskapsutvinningen fick jag en uppfattning om huruvida frågorna var möjliga att besvara.

Med hjälp av dessa två faktorer var det sedan dags att ta itu med att konstruera något användbart.

### 3.8. Rapportförslag

Nästa steg i arbetet innefattade att ta fram en rad förslag på rapporter/beslutsunderlag och visa dem för berörd personal i Borås. Förslagen togs fram utifrån resultatet från intervjuerna i kombination med resultatet från kunskapsutvinningen. Förslagen gjordes helt manuellt i Excel, utan någon anknytning till databasen eller programmering. Helt enkelt färgglada sidor med förslag på selekteringar, knapptryckningar och diagram- respektive tabellutseende.

Här kom ett nytt inslag av användarmedverkan, således. Rapportförslagen gjordes något mer omfattande än vad som sedan skulle byggas, för att användarna skulle kunna välja ut de viktigaste varianterna. I något fall nämndes också rapporter som egentligen inte kan genomfö-

---

ras idag, eftersom någon information saknas i databasen. Vi bedömde dock att man skulle kunna, och kanske borde, lägga till något saknat fält på varje nivå och därmed uppnå den önskade funktionen.

Rapportförslagen visades upp vid ett arbetsmöte, men tyvärr var de närvarande inte riktigt beredda att ge respons vid detta tillfälle. Bättre hade varit att i förväg skicka ut förslagen. Reaktionen blev därför väldigt liten, eller ingen alls, och vi bestämde oss för att lägga till ett extra moment här. I stället för att omedelbart ta diskussionen om vilka förslag som var intressanta att arbeta vidare med bestämdes att arbetsgruppen skulle få förslagen via e-post, och att diskussionen skulle tas vid nästa arbetsmöte.

Så skedde inte. Vad som i stället hände var att två arbetsgruppsdeltagare på var sin avdelning, P och D, lät förslagen gå ut på remiss. Detta gav ingen överväldigande respons. På D-avdelningen fick man in ett svar, på P-avdelningen två svar. Dessa svar bestod i princip av förkryssade alternativ bland rapportförslagen.

På D-avdelningen tog arbetsgruppsdeltagaren resultatet till sin enhetschef. De gick igenom punkterna tillsammans och kom fram till att det enda remiss-svar de fått in var ett förståndigt och bra val.

På P-avdelningen satte sig arbetsgruppsdeltagaren med de två som hade kommit med någon respons och tillsammans kom man fram till vilka rapportförslag man ville prioritera.

Listorna från P och D sammanställdes sedan av projektledaren för SiRi. Ett kortfattat epost-meddelande till mig gav till känna vilka fem förslag som var intressanta.

Den här delen av utvecklingen skedde således utanför min kontroll och den enda respons jag fick till mig var en lista på valda rapportförslag. Eftersom jag var intresserad av funderingarna kring valen ringde jag själv upp projektledaren och de två arbetsgruppsdeltagarna.

### 3.9. Programmering

Till slut gällde det att ta fram en effektiv och lättanvänd programmodul för beslutstöd, en modul som var en naturlig del av det befintliga systemet. Modulen skulle lätt aktiveras från befintligt system. Användaren skulle endast behöva göra enkla val och begränsat antal ytterligare inmatningar för att få en presentation av den beslutstödande informationen.

Att visa diagram på ett intranät har sina intressanta problem. Det finns på marknaden åtminstone en komponent för detta ändamål. Med komponent menas här en fristående, inkapslad programkodsenshet. Denna komponent installeras på datorn och är inte synlig för användaren. Komponenten kan av programmeraren anropas i vilken applikation som helst med rätt parametrar. Komponenten utför då något moment och levererar eventuellt också något resultat, t ex en utvald mängd data från databasen. Den komponent för diagram på Internet som finns att köpa är utmärkt för att visa några få värden i ett diagram. Problemet är att i takt med att inmatade data ökar i antal blir diagrammet allt svårare att läsa av, eftersom skalningen inte förändras.

Efter ett tips från min handledare Robert Johansson bestämde jag mig för att försöka använda Excels diagramfunktioner till det här. Idén var att mata ut data till Excel och låta Excel skapa

---

ett diagram. Diagrammet kunde sedan exporteras ut på hårddisken i form av en bild. Denna bild kunde sedan visas upp på intranätet.

För att åstadkomma något sådant föreföll det enklast att programmera en komponent. Denna komponent kunde sedan anropas från intranätet med de data man ville få presenterade. Komponenten skulle sedan sköta skapandet av diagram och bild.

Eftersom Visual Basic (VB) är det programmeringsspråk jag är mest insatt i föll det sig naturligt att använda det. HTML är inte riktigt min hemmaplan (men det kan kanske bli). Till att börja med kodade jag min komponent i en VB-grupp, så att jag direkt kunde prova komponenten mot en VB-applikation. Metoden med VB-grupp går ut på att man inte kompilerar och installerar sina försöksversioner av komponenten. I stället inför man en enkel VB-applikation i samma programmeringsenhet, och låter den anropa komponentkoden direkt. Efter några dagars testande hade jag åstadkommit en relativt enkel, första version, som fungerade.

Att få denna komponent att fungera från intranätet visade sig vara betydligt mer tidsödande och besvärligt. Särskilt när jag började ge mer detaljerade anvisningar om hur diagrammen skulle se ut. Ett problem visade sig så småningom vara att Excel -97 kom ut på marknaden några månader tidigare än VB 6.0. I Excel används det som kallas VBA (Visual Basic for Applications). De månader extra man hade för VB 6.0 innebar att man lade till en del funktioner och metoder och därmed är de två inte fullt kompatibla (Lomax, 1998). Ett annat problem var att jag hade använt mig av en genväg i min ursprungliga komponent – jag drog igång Excels diagram-guide och skapade med hjälp av den mitt önskade diagram. Det visade sig att den inte ville låta sig dras igång från intranätet. Varför vet jag ännu inte, eftersom jag i stället koncentrerade mig på att skapa diagrammen från grunden.

I programmering används något som kallas enumeration. Det är ett sätt att förenkla programmeringen. I stället för att t ex programmera att en etta i ett visst sammanhang betyder typregister, en tvåa betyder hjälpregister osv ger man typerna namn. Det finns inbyggda enumerationer i såväl VB som Excel, men man kan också skapa egna. I VB kan det se ut så här:

```
Public Enum Report_Mode
    icTypregister = 1
    icHjalpregister = 2
End Enum
```

Vitsen är att man sedan inte behöver komma ihåg vilken siffra som motsvarar vilket register, utan kan använda beteckningen. Man behöver egentligen inte ens komma ihåg namnet på varje beteckning utan det räcker att man kommer ihåg enumerationens namn. Skriver man i koden in Report\_Mode med en punkt efter hjälper programmeringsmiljön i VB till och visar upp en lista över de beteckningar som finns att välja på under Report\_Mode.

Att det inte går att skicka sådana beteckningar från intranätet var jag medveten om. Det brukar dock vara möjligt att skicka den rätta siffran och ta emot den som en enumerations-variabel i komponenten.

Inte heller detta fungerade i det här fallet. Detta visade sig bland annat bero på att det som i Excel heter Charttype är indelat i tre olika undergrupper. Dessa undergrupper kallas gallerier. Eftersom detta med gallerier inte är dokumenterat i Excel-dokumentationen (Roman, 1999) var det svårt att hitta någon exakt beskrivning av funktionen. Lösningen på detta blev till slut en villkors-sats som oberoende av gallerierna sätter Charttype till rätt typ utifrån den siffra



som kommer in. Om t ex siffran 23 skickas in står det i villkorssatsen att ett stapeldiagram av en viss typ skall skapas.

Att konfigurera alla Excels 72 diagram för anrop utifrån får bli en senare fråga. Koncentrationen fick läggas på de diagram som var viktigast för denna uppsats, dvs diverse olika stapeldiagram.

Programmeringen av sidorna till intranätet var en utmaning, eftersom HTML var relativt nytt för mig.

För att kunna göra sidorna dynamiska användes ASP-programmering, där ASP står för Active Server Page. Förenklat kan man säga att HTML målar upp saker på användarens bildskärm medan ASP-sidan har en aktiv kontakt med servern i intranätet. Denna kontakt är givetvis nödvändig så fort man skall hämta data från en databas eller använda komponenter som ligger på servern.

För kalenderfunktioner kodade jag en liten kalender med tre boxar för år, månad och dag. Här hittade jag ett amerikanskt exempel (från Isaacs, 1997) som i någon mån liknade det jag behövde, och skrev vidare utifrån det. En lustig detalj i sammanhanget tycker jag är att jag satt en hel dag och undrade varför Sql-servern skickade mig kryptiska felmeddelanden när jag startade upp mina rapporter. Det föreföll först att ha ett samband med datatyper och frågan jag kämpade med var om jag trots alla mina kontroller exempelvis skickade ett anrop med en textsträng som parameter fast parametern borde ha varit av någon datum-datatyp. Lösningen var att jag hade satt uppstartperioden till 1990-01-01--9999-12-31 för att låta användaren börja med en stor mängd information och sedan avgränsa sig. Men Sql-servern accepterade bara datum t o m 2079-06-06!

Det visade sig ganska snabbt bli mycket kod på mina ASP-sidor. Dessutom, som tidigare nämnts, är komponenter en bättre lösning bland annat av prestandahänsyn och för att samma kod och samma anrop då kan användas av flera olika applikationer. Utöver grafikkomponenten fick det därför också bli en komponent som vid anrop söker upp information ur databasen, summerar och/eller sorterar och levererar en hanterbar resultatmängd.

## 4. Teori

I det här avsnittet vill jag klargöra innebörden av en rad begrepp som vi använder i dagligt tal men som i teorin kring beslutstöd och kunskapsutvinning i vissa fall har en något annorlunda eller mer distinkt betydelse.

Litteraturstudierna har utgjort en relativt liten del av detta arbete. Om man finge använda begreppet ”quick and dirty” även i detta sammanhang hade det passat ganska bra in. Visserligen har jag lagt en del möda på att hitta artiklar och litteratur i aktuella ämnesområden, men problemet har varit, som redan nämnts, att stöd på operationell nivå har behandlats ganska styvmoderligt i den litteratur jag fann. Det jag fick var en bakgrund och en överblick över ämnesområdet som sådant och en allmän kunskap om begreppen beslutstöd och kunskapsutvinning. Det talas inte heller mycket om beslutstöd av ett mindre avancerat slag, som enklare rapporter eller verktyg inom befintligt system, utan man vill helst diskutera data warehouse, data mining – stora komplexa verktyg som kräver sin kvinna för att hantera.

### 4.1. Beslut

Olika typer av beslut fattas på olika nivåer inom en organisation. En vanlig nivåindelning är Anthonys (1965, enligt Finlay, 1994) – den tredelade där man talar om operationell, taktisk och strategisk nivå.

**Operationell nivå** handlar i första hand om dagliga problem av strukturerad natur. Besluten baseras främst på historik, på vad som redan har skett.

**Taktisk nivå** kännetecknas av att rutinmässiga beslut påverkas av ej rutinmässig input (Murray, 1974, enligt Finlay 1994). Exempel på en taktisk fråga är hur man bäst använder de resurser man redan har.

**Strategisk nivå** rör långsiktiga beslut kring organisationens relationer med omvärlden. Besluten handlar om framtiden i första hand. Exempel på en strategisk fråga är hur man skaffar sig mer resurser på ett bra sätt.

Ett annat sätt att dela in beslut är att se på hur strukturerat problemet som ska lösas är (Simon, 1977 enligt Turban & Aronson, 1998).

### 4.2. Kunskap

Kunskap betyder dels förmågan att utföra något, dels vetskap och kännedom om något. Att ha kännedom om något kan i vissa fall börja med att man har en aning – att empiriskt kunna visa att denna aning är riktig är ett bra stöd i en beslutssituation.

En kunskapsmodell innefattar data, information, intelligence och den process genom vilka dessa formas. (Finlay, 1994).

**Data** kan sägas vara en mängd osorterade uppgifter där inget specifikt mönster kan urskiljas.

**Information** kan sägas vara data som i någon transformerad, sorterad form emottas av en person för att vara honom till nytta i hans arbete (Finlay, 1994). Information är det vi förmedlar och tar emot (Andersen, 1994).

**Kunskap (intelligence)** är resultatet av samordning och sammanjämkning av en mängd information som bär i sig eller föranleder vissa slutsatser (Murray 1979 i Finlay, 1994). Råmaterialet för kunskap är information. Kunskap är kopplad till en bestämd människa och beroende av denna människas referensram, begreppsvärld (Andersen, 1994).

### 4.3. Knowledge Discovery in Databases (KDD)

Det förekommer en lång rad benämningar på det som i korthet betyder att hitta användbar information i en mängd data. Exempel är, utöver KDD, kunskapsutvinning, data mining, informationsupptäckt, dataarkeologi osv.

Fayyad et al sätter inte likhetstecken mellan data mining och KDD. I deras definition av KDD är själva data mining processen endast ett steg av sju. KDD är hela processen från undersökandet och förståelsen av problemområdet, via val av data som skall bearbetas och städning/förberedelse av datan respektive reducering eller transformering av data. Därefter själva utförandet av datagrävandet. Processen avslutas med tolkning och användande av kunskapen. (Fayyad et al, 1996).

Som redan har nämnts fokuserar detta arbete inte på data mining. Själva processen är ändå att likna vid en KDD. Arbetet har, precis som i den beskrivna KDD-processen, startat med studier kring problemområdet, en kartläggning av vilka frågor vi hade att besvara. Därefter kom en analys av databasen för att se om behövliga data fanns och om de var användbara direkt eller behövde "städas" - dvs det som i KDD kallas förberedelse av datan. I KDD kommer som nästa steg data mining - här har vi i stället formulerandet av de sql-frågor som skulle ge svaret på de frågor vi hade att besvara. Slutligen sker en presentation av kunskapen och användarna har att tolka och använda denna kunskap.

### 4.4. Decision Support System (DSS)

I detta avsnitt görs ett försök att bena ut olika sätt att definiera DSS. En kort bakgrund ges till att börja med, därefter en mycket kort beskrivning av några definitionsansatser. Slutligen görs ett ställningstagande för det här fallet, och beskrivs vilka funderingar som lett till detta ställningstagande.

#### Bakgrund

Hjälpmiddel för beslutstöd har utvecklats sedan 50-talet då kalkylatorer, tidiga datorprogram, statistiska modeller och enkla management science modeller användes för att organisera, summera och beräkna. På 60-talet kom databashanteringssystemen och med dem Management Information Systems (MIS). Kontorsautomation utvecklades främst under 70-talet. DSS dök upp på 80-talet och har fortsatt att utvecklas under 90-talet då också expertsystem och executive information systems blivit aktuella. Nästa generation av expert system, grupp-DSS och artificiella neurala nät är nästa steg i utvecklingen.

Trots att man alltså har talat om DSS i närmare tjugo år finns det ingen generellt accepterad definition av DSS. Begreppet kan betraktas som ett paraply-begrepp som innefattar varje datoriserat system som används för beslutstöd i en organisation (Turban & Aronson, 1998). En bred definition av ett DSS kan alltså sägas vara "ett datorbaserat system som underlättar beslutsprocessen" (Finlay, 1994).

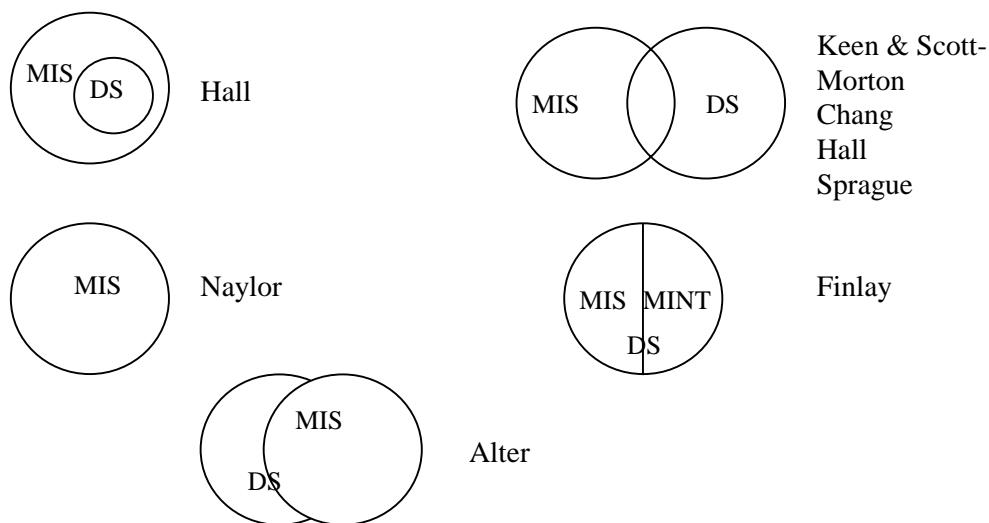
### Definitionsansatser

Enligt Finlay (1994) finns det ingen teori om DSS i strikt akademisk mening, eftersom utveckling av DSS inte kan sägas följa en rad fastslagna och testade lagar. Dock har många aktiva inom området gjort försök att definiera vad ett DSS är och göra lämpliga klassificeringar.

Keen och Scott Morton (1978, enligt Finlay 1994) utgick från impact, pay-off och relevans för ledningen i organisationen och urskilde då de tre områdena MIS, Management Science och DSS. De skiljer alltså på ett MIS och ett DSS.

Naylor (1982, enligt Finlay, 1994) däremot hävdar att det inte finns något hos DSS som inte redan existerar i Management Science och MIS och anser därmed inte att det över huvud taget finns behov av begreppet DSS. Se figur 1.

Turban & Aronson (1998) ser sex huvudsakliga typer av datoriserade beslutstödssystem. Det är Transactions Processing Systems (TPS), Management Information Systems (MIS), Decision Support Systems (DSS), Expert Systems (ES), Executive Information Systems (EIS) och Neural Computing. TPS resulterar enligt Turban & Aronson i summerande rapporter på operationell nivå, medan MIS producerar återkommande rapporter och efterfrågade rapporter i realtid över strukturella flöden och avvikelser från de normala flödena. DSS producerar information för att stödja specifika beslut. Här skiljer man återigen på DSS och MIS, och tillför till samma grupp även TPS.



**Figur 1.** Olika synsätt på relationen mellan MIS och DSS enligt olika informatiker (Källa: Finlay 1994)

---

Finlay (1994) föreslår att DSS kan indelas i olika typer utifrån vad de producerar. Ett DSS som producerar information kan kallas Management Information System (MIS), medan ett system som producerar intelligence kallas Management Intelligence System (MINTS) (Finlay, 1994). I det här synsättet ingår tydligen MIS som en typ av två i begreppet DSS. Linjen mellan MIS och MINTS är dock inte en skarp skiljelinje, det kan i vissa fall vara svårt att avgöra om produkten från ett system endast är information eller om det är mer kunskapsbringande.

En man vid namn Little gav på 70-talet (Turban & Aronson, 1998) ett antal kriterier för om ett system skulle få kallas beslutstöd. Enligt dessa är systemet ett beslutstödssystem om det:

1. är enkelt att förstå.
2. är robust.
3. är lätt att kontrollera.
4. är användarvänligt
5. innehåller viktig och relevant information
6. är lätt att interagera med

Förutom att punkt 6 och 4 tycks vara ungefär samma sak kan sägas om dessa kriterier att de inte egentligen gäller beslutstödssystem utan alla informationssystem. Endast punkt 5 är av något mindre allmän natur.

De flesta tycks dock vara överens om att ett beslutstödssystem, oavsett vilken indelning eller namngivning man föredrar, kännetecknas av att ett sådant system ger nödvändig information för slutliga beslut fattade av människor.

Gemensamt för de flesta definitionerna är att de ignorerar den huvudsakliga betydelsen av DSS (nämligen att stödja och förbättra beslutsfattande) och i stället försöker närma sig en definition genom att beskriva beslutssituationerna (exempelvis strukturerade/ostrukturerade problem) eller genom att studera befintliga system. Ett tungt argument emot diskussionen kring strukturerade respektive ostrukturerade problem är naturligtvis att ett problem kan uppfattas som strukturerat av en person med en viss kunskap och som ostrukturerat av en person med en annan kunskap och bakgrund (Finlay, 1994). En definition som bygger på idag befintliga system löses snabbt upp i tomma intet, eftersom nya system utvecklas hela tiden.

### DSS grundstenar

Ett DSS kan sägas bestå av en datamodell, en logikmodell och ett användargränssnitt (Finlay, 1994). Datamodellen är faktiska data (ex. sålda cyklar: 12, 10, 15 / priser: 750, 1250, 4000). Logikmodellen utgör en representation av hur data skall behandlas (ex. sålda cyklar x pris = försäljning).

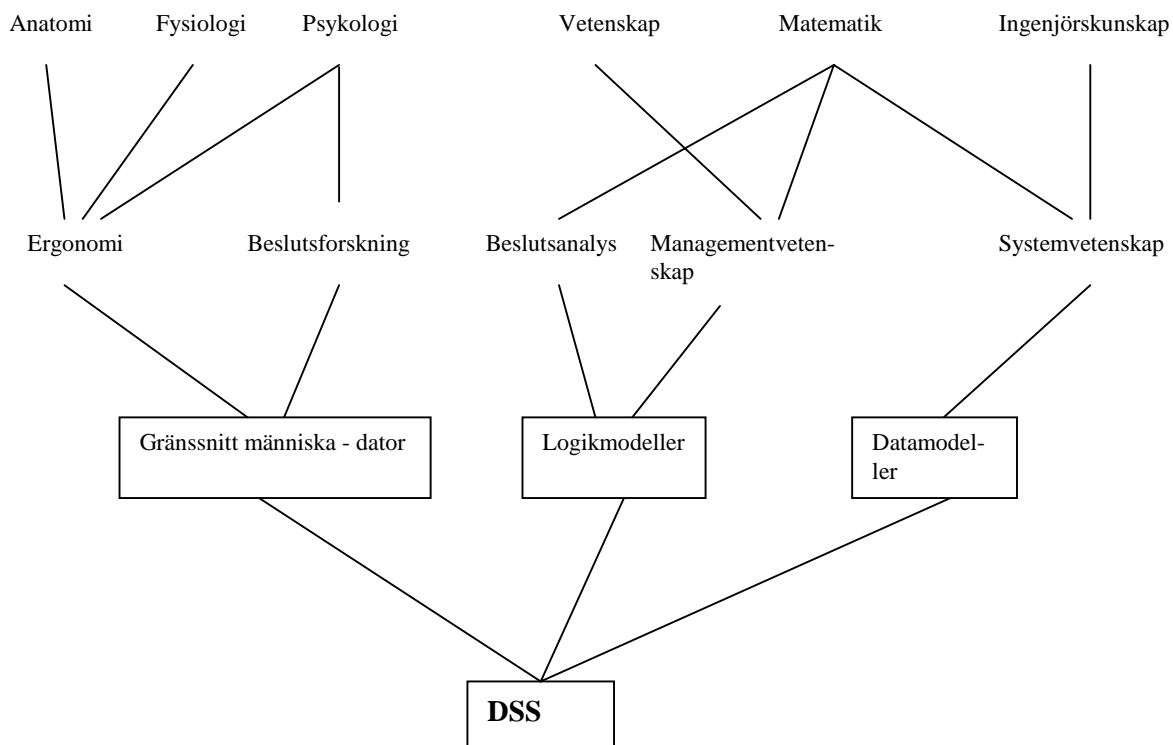
Ett annat sätt att få en djupare förståelse för vad ett DSS är kan enligt Finlay (1994) vara att i stället studera de fem vetenskaper DSS bygger på, nämligen:

1. **Management Science**, som har sin grund i vetenskapliga metoder och matematiska tekniker för att fatta affärsmässiga beslut. Målet är att fatta bättre beslut genom bättre metoder.
2. **Beslutsanalys**, som hanterar beslutsfattande i osäkra situationer med många alternativ. Beslutsanalys erbjuder ett ramverk i vilket beslutssituationen kan struktureras och tydlig-

göras.

3. **Datalogi**, som har gjort det möjligt att hantera stora mängder data.
4. **Ergonomi**, som är vetenskapen om människans interaktion med sin fysiska omgivning.
5. **Beslutsforskning**, som kretsar kring de kognitiva processer som är involverade i beslutsfattande. En nyckelprincip är att varje DSS skall stödja och förbättra befintlig form av beslutsfattande snarare än att införa en bättre beslutsform.

Sambanden mellan dessa vetenskaper och de olika delarna i ett DSS ses i figur 2. I figuren ses även de basläror som i sin tur ingår i de fem vetenskaperna ovan.



**Figur 2.** Vetenskapernas medverkan i DSS-utveckling (Källa: Finlay, 1994)

Att studera ett DSS utifrån de läror som gemensamt utgör dess ursprung kan ge en bredare bild av vad ett DSS är, eller med andra ord vad det består av. Det visar också tydligare varför det är svårt att sammanfatta DSS i en enda teoretisk ram - var och en av de fem vetenskaperna ovan har sina specialister, sina definitioner och sina teorier. I stället för att alltså tala om en enda teori för DSS skulle man kunna se de fem ovan nämnda vetenskaperna och deras ramverk av teorier och definitioner som en sorts bärande hörnstenar i ett bygge. För en utvecklare ger ändå detta en sorts stöd och medvetande om hur ett DSS fogas samman till en på alla de fem områdena fungerande enhet.

### Kunskap och DSS i detta arbete

Beslut fattas i vissa fall grundat på erfarenhet, i andra fall grundat på tabeller och grafer, svart på vitt i strukturerad form.

Jag hävdar att det i båda fallen handlar om kunskap enligt Finlays modell ovan. Det som kortfattat kan beskrivas som erfarenhet är i själva verket kunskap som formats av sorterade data – även om sorterandet har skett i någons hjärna och inte på rutiga papper. För en person som dväljs i databasernas, programmeringens och tabellernas rutiga och välordnat strukturerade och dokumenterade värld känns möjligen en så abstrakt resultatmängd (erfarenheten!) som lite osäker, men oavsett om man vill sätta tilltro till erfarenheten eller inte så kan man fortfarande se den som kunskap, baserad på sorterade data (information), i enlighet med Finlays kunskapsmodell.

Detta arbete vill bland annat underlätta och tydliggöra kunskapen genom att ta fram de konkreta bevisen för att erfarenhetens slutsats var korrekt.

Definitionsansatserna ovan kändes till att börja med dels alltför diffusa och spretiga, dels alltför ensidigt inriktade på beslutstöd för styrelse- och ledningsgrupper.

Naturligtvis har ett beslutstöd på operationell nivå intresse även för ledningen, och skulle då kunna beskrivas i Keen & Scott Mortons termer av impact, pay-off och relevans. I Turban & Aronsons sex olika typer urskiljer jag tre som utgör mer avancerat beslutsunderstödjande system, medan tre är karaktäriserade av vilket resultat de producerar. Finlay är på samma spår men skiljer endast mellan informationsskapande och kunskapsskapande – och han kallar alltihop för DSS.

Att som Turban & Aronson skilja mellan summerande rapporter och återkommande rapporter synes en smula vanskligt – summeras inte data i återkommande rapporter?, är aldrig summerande rapporter återkommande?

För att kunna göra en fullständigt klar analys av beslutstödens klassificering och definiering behövs ett djupare studium i ämnet än vad jag kunnat hinna med. Min otillräckliga insikt gör att jag vill stanna vid den enklaste och inte särskilt djuplodande definitionen:

Ett beslutstödssystem kännetecknas av att ett sådant system stödjer och förbättrar beslutsfattande (Turban & Aronson, 1998)

Denna definition är tillräcklig för att beskriva vad mitt arbete syftar till. Det är dessutom så att beslutstödssystem kan ses som en "paraply-term" för att beskriva vartenda datoriserat system som stödjer beslutsfattande på ett eller annat sätt, oberoende av vad som matas in eller hur presentationen av svaret ser ut, oberoende av komplexiteten i frågorna som ställs osv (Turban & Aronson, 1998). Jag har visserligen helt kort försökt ge en överblick över detta stora område, men min uppsats har inte för avsikt att ge en tung genomgång av områdets teoretiska bredd.

Vill man förtydliga begreppet något känns Finlays indelning efter producerat resultat naturligt, eftersom den inte låser definitionen varken vid någon nivå i organisationen eller vid någon speciell typ av problem, utan utgår från **hur** beslutsfattandet stöds.

I det här arbetet är det uteslutande system/rapporter av MIS-karaktär (enligt Finlays definition, således) som är aktuella, dvs framtagande av information, inte av intelligence.

#### 4.5. Beslutstöds utseende

Vilken typ av utseende skall ett beslutstöd ha för att ge bäst och mest information? Detta måste naturligtvis bero på vilken typ av beslutstöd man är ute efter.

I det här arbetet talar vi, som tidigare nämnts, om relativt enkla och inte speciellt komplexa frågor. Vi talar om den ena anläggningen eller den andra, den ena pumpen eller den andra, var har kostnaderna varit störst osv. Frågan som ställdes redan i inledningen var hur man skall ställa sig när man utformar själva programdelen. Skall det vara diagram eller tabeller?

Dickson, DeSanctis och McBride gjorde 1986 en serie av experiment för att testa hypotesen att effektiviteten av grafisk framställning i beslutstödsammanhang varierar beroende av problemområdets natur. De kom fram till följande.

En grafisk presentation av ett beslutsunderlag är inte alltid effektivare än en presentation i tabell-form (Dickson, DeSanctis & McBride, 1986). När det gäller problem som inte är av komplext slag, exempelvis enkla, ekonomiska rapporter, blir förståelsen och besluts kvaliteten varken bättre eller sämre med grafisk presentation.

När en enorm mängd data presenteras och antalet intryck materialet avser att ge är få är grafer att föredra framför tabeller.

Vidare kom man fram till att viktigare än val av graf respektive tabell var att bryta upp ett stort beslutsunderlag i mindre bitar.

I det här arbetet förekommer både sådana presentationer där enorma mängder data skall presenteras på ett överblickbart sätt och sådana där mängden data är mindre. För de senare betyder detta, enligt Dickson, DeSanctis & McBride (1986), att vi i princip kan presentera rapporterna i valfri form. Vi har därför valt att fråga användarna själva om vad de föredrar för de mindre komplexa rapporterna.

I SiRi kommer man ofelbart att handskas med stora mängder data om man väljer att ta fram någon som helst rapport som skall hantera alla anläggningsdelar. Dickson et al visade att ett stort underlag bör brytas ned i mindre bitar. I lösningsförslagen har jag tagit fasta på detta genom att i många fall visa upp några anläggningsdelar i taget och på ett markerat sätt låta användaren välja vilken nivå i anläggningsstrukturen man vill röra sig på. Se avsnitt 6.3.

#### 4.6. Programmeringsbegrepp

Programmeringen i detta arbete har skett med hjälp av Visual Basic 6.0, VBScript, JavaScript, ASP, HTML och viss mån DHTML. ActiveX-komponenter har programmerats.

I den mån dessa begrepp är okända för läsaren görs här ett försök att någorlunda kortfattat ge en beskrivning av dem. Det är dock en starkt förenklad beskrivning.



### HTML och DHTML

HTML, HyperText Markup Language, är i grunden ett sätt att med hjälp av olika "märken" (tag) styra upp hur ett dokument skall visas på bildskärmen. Skriv t ex "<small>Hej</small>" och ordet hej kommer att visas med liten text. Det är också ett sätt att få struktur på dokument - sidan har exempelvis head och body och det finns markeringar för olika rubriknivåer osv.

DHTML, Dynamisk HTML, kom med Explorer 4.0. Det nya var bland annat att man har större direkt tillgång till dokumentets olika beståndsdelar och element, och kan nå texturor och knappar och dynamiskt ändra innehållet i dem utan att gå via kommunikation med server eller använda programmerade objekt (som Java Applets eller ActiveX-kontroller). I DHTML finns även en kraftfull händelsehanterare som gör att programmeraren inifrån HTML-sidan kan fånga upp vad användaren gör inom browsern, och dynamiskt ändra utseende och innehåll på sin sida utifrån vad användaren gör (Isaacs, 1997).

### ASP

Active Server Page - en sida som inte endast innehåller HTML utan även aktivt kommunicerar med servern.

### VBScript och JavaScript

Ett script är en liten klick programmering, i det här fallet på en intranätsida. Scripten ger programmeraren möjlighet att lägga in funktioner och metoder som skall utföras varje gång sidan laddas, eller vid anrop exempelvis från en knapp på sidan. Kortfattat ger det alltså ökade möjligheter till dynamiska och interaktiva intranätsidor.

### ActiveX komponent

En ActiveX komponent kan förenklat sägas vara en inkapslad programmeringskod som utför någon eller några specifika uppgifter. Koden kompileras ihop till ett litet paket som kan placeras på servern i ett nätverk och anropas från klienterna i nätverket. Den kan även placeras lokalt hos klienten. Komponenten har ändelsen .dll, vilket står för "dynamic link library". Det är alltså en sorts litet extrabibliotek med anropbara funktioner.

Att använda komponenter av det här slaget har flera fördelar. Det underlättar programmeringen på ett intranät, eftersom endast ett anrop behövs för att aktivera en önskad procedur. Dessutom blir exekveringen snabbare, naturligtvis under förutsättning att servern är snabb. Applikationerna blir också mindre till storleken, eftersom de inte behöver innehålla alla funktioner utan kan anropa funktioner i komponenterna vid behov.

De komponenter som detta arbete har lett fram till är två dll-komponenter, programmerade i Visual Basic. Den ena komponenten tar emot ett anrop med parametrar (exempelvis anläggningsnivå, objekt-id och datum) och returnerar en resultatmängd med data att presentera på intranätet. Den andra komponenten tar emot en resultatmängd med data från databasen samt en rad information om hur önskad graf skall se ut och skapar sedan en Excel-graf. Denna graf lagras ut på serverns hårddisk i det bildformat som kallas jpeg. Jpeg-filen kan sedan användas som bild på intranät.

## 5. Underhållssystemet SiRi

SiRi är ett underhållssystem speciellt framtaget för Borås Energi AB, men transparent och generellt programmerat för användande i liknande miljöer på andra ställen. Systemet är programmerat i Visual Basic 6.0. Databasen är en relationsdatabas, Microsoft SQL-server. Till rapporterna/blanketterna har vi använt Crystal Report.

För att i någon mån ge en uppfattning om systemets storlek kan sägas att databasen idag (april 2000) innehåller 320 tabeller och att den är ca 370 MB stor. Den exekverbara filen är knappt 6 MB. All utskriftshantering ligger i en egen exekverbar fil, som i sin tur är drygt 4 MB stor.

Huvudprogrammet startas från SiRis egen hemsida på Borås Energis intranät. Vem som helst kan göra felanmälningar direkt via intranäts-funktioner, men det skall speciell behörighet till för att få starta huvudprogrammet. På intranätet finns också en speciell, förenklad html-version av SiRi för utförare (se nedan).

SiRi körs i nätverk på Borås Energi AB av idag 22 stycken installerade användare.

Här beskrivs några väsentliga begrepp i underhållssystemet SiRi, för att förhoppningsvis ge större möjlighet att följa med i resonemangen i resultatavsnittet.

Hjärtat i SiRi kan kort sägas vara delarbetsordern. Det har med tiden uppstått en rad olika funktioner i systemet, men alla kretsar kring delarbetsordern.

### 5.1. Kort kring en delarbetsorders liv

En delarbetsorder uppstår antingen då någon gör en felanmälan eller då något återkommande, rutinmässigt arbete skall genomföras. Den vandrar sedan från person till person (skickas vidare elektroniskt) och byter status för varje steg, så att man alltid vet hur långt en delarbetsorder har hunnit i sin livscykel. Felanmälan har status 1, en avslutad delarbetsorder har status 13.

Först är det en **beredare** som går igenom delarbetsordern. Han bestämmer om det är lämpligt med en undersökning, en reparation direkt, en nyinvestering osv.

När beredaren är färdig skickar han delarbetsordern till en **arbetsledare**. Arbetsledaren planerar in arbetet, fördelar resurser osv.

Näste man är **utföraren**. Han gör jobbet. På delarbetsorderna bokar han in hur lång tid det faktiskt tog, vilket material han behövde använda och liknande.

Delarbetsordern kommer sedan att vända tillbaka och hamna hos beredaren igen, nu med åiterrapporteringsdelen förhoppningsvis ordentligt ifyllt. Om beredaren är nöjd med resultatet kan han avsluta delarbetsordern, och föra över den till historik.

### 5.2. Historik

I SiRi-sammanhang finns två typer av historik. Den första är mer direkt, ett handfast lagrande av datum, användarsignatur och händelse för en delarbetsorder, en skiftrondsdefinition eller



Finns inbyggnadsbeteckningar eller uppställningsplats läggs dessa beteckningar till på slutet av beteckningen.

Inte bara beteckningen är uppbyggd på detta vis, utan även databasen är hierarkiskt relaterad från verksamhetsområde ned till komponent. El, instrument, reglerteknisk- och datorteknisk utrustning kan ges en inbyggnadsbeteckning för att beskriva var komponenten är placerad, om den är placerad i ett skåp eller dylikt. Uppställningsplatser är i princip byggnader och rum.

Utöver detta finns för block, aggregat och komponenter något som kallas gemensamma block, gemensamma aggregat och gemensamma komponenter. Det är en praktisk funktion för att slippa ange samma karaktäristik för tusentals ledningar eller fjärrvärmecentraler osv. Det gemensamma objektet innehåller exempelvis fabrikat, typ och namn. Varje objekt på dessa tre nivåer är knuten till ett gemensamt objekt.

Ett aggregat kan också ha ett huvudaggregat. Det innebär att om man exempelvis ställer av ett aggregat, som i sig är huvudaggregat för flera andra aggregat, så kommer samtliga aggregat att ställas av utan att det behöver göras för vart och ett.

På alla nivåer finns två undernivåer som automatiskt skapas av systemet. Det är dels ODEF, som står för odefinierad och som kan användas exempelvis när någon vill felanmäla något han sett, men är osäker om på vilket block eller system osv han befinner sig. Det är också OGEM som står för gemensam och som används för sådant som är gemensamt för flera delar på överliggande eller innevarande nivå i hierarkin.

Aggregatet är den nivå som arbetet kretsar kring i SiRi. Varje delarbetsorder är knuten till ett aggregat, oavsett vilken nivå i anläggningen arbetet skall utföras på. Om arbetet avser ett block hänger delarbetsordern på aggregatet OGem på systemet OGem på det aktuella blocket.

## 6. Resultat

### 6.1. Faktorer som påverkar användarnas inställning till beslutstöd

#### Beslutstödet utformning

”Det skall bara vara en knapp” hörde jag flera gånger. Det var uppenbart att många av användarna upplevde sin datoriserade värld som en smula kaotisk just nu. De hade inte hunnit så långt med SiRi, men upplevde att ”det är för stort”. Att bygga på ytterligare saker ställde de sig tveksamma till i vissa fall. När de fick försäkringen att detta inte skulle innebära en utbyggnad i sig, utan just ”bara en knapp” för att förenkla deras arbete och frigöra tid för dem blev de mer positiva.

Det fanns ett självklarare och positivare accepterande av ett datoriserat beslutstöd hos chefspersonalen. Önskemålet om enkelheten, ”bara en knapp”, var dock detsamma.

#### IT-miljön

Ett annat återkommande ämne var att de är utsatta för den vanliga åkomsten inom stora organisationer – ”det finns så många system”. De har tillgång till viss ekonomisk uppföljning från organisationens ekonomisystem – men där får de endast vissa klumpsummor. Ett förråds-system för reservdelar håller på att implementeras (en koppling mellan SiRi och detta förråds-system är beslutad och skall genomföras). Det finns ett system som från början var ett simuleringsprogram för optimering av energiproduktionen, men som med tiden byggts ut till att omfatta såväl emissioner som bränslehantering och drifttider. Utöver dessa olika system används Excel flitigt.

Ytterligare funderingar gäller att själva maskinvaran i organisationen tyvärr inte riktigt har rätt kapacitet, vilket i vissa lägen kan innebära en irriterande tröghet. Detta gör att man ställer sig tvivlande till ”mer finesser”.

#### Kunskap

Den som praktiskt handhar en pump, en fjärrvärmecentral eller någon annan teknisk utrustning i sitt dagliga arbete bildar sig naturligtvis en uppfattning om utrustningen. Han har en känsla av vilka fabrikat som håller bäst, hur ofta filter måste bytas eller hur ofta skall man rycka ut med oljekannen.

I mina samtal med personalen var det tydligt att man såg olika på begreppet kunskap beroende på vilken nivå i organisationen man befann sig på. Ju närmare den rent praktiska nivån ju större var tilliten till erfarenhet. Ju närmare ledningen ju större var förtroendet för tabeller, grafer och beräkningar. Det beskrivs kanske bäst av en informant, som ställd inför något tjuvigt rapportförslag från mig svarade:

”... ? Kanske för den som kommer ny hit, efter mig, när jag slutar här... Jag behöver det inte, jag vet ändå.”

### Position i organisationen

Det fanns en annan markant skillnad beroende på vilken position man hade, en skillnad som kanske är så självklar att den inte ens borde nämnas. För den som arbetar dagligen med en viss bestämd del av verksamheten vill gärna världen bestå av just den delen, oavsett om ens arbetsområde är av det mindre eller större slaget. Det gör att det är svårare att se värdet av ett mer flexibelt beslutstöd för den som har ett mindre arbetsområde. Denna gradvisa utvidgning av horisonten, arbetsmässigt, var mycket tydlig.

Som redan nämnts var personal i chefsposition mer positiva till begreppet beslutstöd än personal i direkt operationell verksamhet. De hade också klara åsikter om vad de tyckte skulle vara positivt för deras underställda att veta.

## 6.2. Informanternas förslag på rapporter

### Allmänt

Här följer samtliga förslag till beslutstöd/rapporter och listor som kom upp under mina samtal med informanterna. Jag har fört samman önskemålen och räknat hur många informanter som kommit upp med idén respektive visat intresse för förslaget. Om något önskemål i sig har betytt en utvidgning av någon annans önskemål har jag fört upp det under samma punkt. Exempelvis ville en informant ha mantimmar per anläggningsdel på tio-i-topp-lista medan en annan ville ha samma sak, men dessutom fördelat på extern/intern personal. Detta är samma lista, har jag ansett, bara med eller utan ytterligare en fördelningsgrund.

I viss utsträckning liknar vissa tio-i-topp-listor andra listor, exempelvis periodrapporter, men de är upptagna som enskilda förslag här, eftersom de skulle komma att se annorlunda ut i periodrapportversion.

Informanternas förslag visas här i tabellform. För att det ska bli enklare att se förtydligandena av förslagen har jag delat upp tabellen i olika delar, och förklaringen kommer strax under.

För att ta reda på om förslaget är realiserbart har en kunskapsutvinning ur databasen gjorts för de olika förslagen. För att öka läsbarheten har denna kunskapsutvinning beskrivits under varje punkt för sig, under rubriken "Databasen". Som verktyg användes i första hand de verktyg som SQL-servern själv tillhandahåller. Med hjälp vybyggaren kan man bygga sql-satser på ett grafiskt sätt, genom att hämta fram rätt tabeller och klicka för de fält man vill ha tag på. Vybyggaren stödjer dock inte mer avancerade sql-satser, så efter ett första grundläggande bygge är det lämpligt att ta med sig sql-satsen till Query Analyzern. Här får man ingen grafisk hjälp, men kan köra sina sql-satser mot databasen och se vilket resultat de ger.

Även diagrammen i SQL-servern är till hjälp – här plockar man fram de tabeller man är intresserad av och kan grafiskt se vilka relationer de har.

I den mån de här förslagen givit upphov till ett rapportförslag i mina lösningsförslag nedan finns mitt förslags nummer angivet i listan.

## Tio-i-topp-listor

| Typ av underlag | Detaljnivå   | Önskat ggr | Lösning nr |
|-----------------|--|------------|------------|
| ”Tio-i-topp”    | Frekvens felanmälningar per fabrikat   | 6          |            |
|                 | Frekvens felanmälningar per anläggningsdel (med trendkurva, 1)   | 8          | 3          |
|                 | Störst ökning av kostnader eller störst ökning av antal ingrepp  | 3          | 4          |
|                 | Underhållskostnad i relation till nypris   | 4          |            |
|                 | Underhållskostnad i relation till producerad energi (GW)   | 3          |            |
|                 | Underhållskostnad i relation till producerad energi och nypris, fördelad på avhjälpande och förebyggande underhåll                           | 1          |            |
|                 | Underhållskostnad per anläggningsdel, totalkostnad i kr inkl bortfall och stoppkostnader, fördelat på arbetstid, material och stoppkostnader | 8          | 1          |
|                 | Vilka fel har tagit längst tid att laga. Mantimmar fördelat på extern personal/intern personal per anläggningsdel                            | 4          | 2          |
|                 | Kostnad i förhållande till tillgänglighet  | 1          |            |
|                 | Kostnad i förhållande till verkningsgrad   | 1          |            |
|                 | Frekvens ingrepp   | 1          |            |

Detta förslag har alla informanter kommit med. Förslaget innebär att man ska kunna se en lista över alla anläggningsdelar, sorterade efter angivna parametrar.

När man talar om en lista där något beräknas per anläggningsdel får man en flerdimensionell lista. Dels talar vi om att kunna se de dyraste / mest tidskrävande / mest felfrekventa överst i listan och de billigaste / minst tidskrävande / minst felalstrande nederst. Men vi talar också om att det skall vara per anläggningsdel, vilket innebär att vi, enligt den skisserade hierarkin i avsnitt 5, har att vandra från verksamhet, anläggning, block och system ned till nivån aggregat eller rentav komponent.

I korthet kan man säga att den allmänna åsikten är att man vill veta vad som var dyrt och vad som tog tid. Sedan går åsikterna lite isär om vilka parametrar som är användbara för detta.

**Databasen**

Dessa listor går i de flesta fall att ta fram ur databasen, med hjälp av ett intensivt räknande.

Ett problem när det gäller fabrikat är att underlaget har luckor – dvs att man angett fabrikatet som odefinierat eller allmänt. Som exempel kunde konstateras vid anrop till databasen att det för ögonblicket (slutet februari 2000) finns 98.774 aggregat i databasen. 93.570 av dessa aggregat anges ha ett fabrikat som är odefinierat (50.989 stycken) eller ”allmänt”. Den största posten med ett verkligt fabrikat angivet är 1.905 stycken aggregat av samma fabrikat. Man skulle vilja påstå att underlaget består av en enda stor lucka. Detta beror bland annat på att aggregat ofta är ihopsatta objekt där de olika delarna, komponenterna, kommer från olika leverantörer. På komponentnivå däremot är fabrikat mer användbart. I databasen finns i detta ögonblick 306.375 stycken komponenter. Tyvärr är drygt 300.000 av dem knutna till odefinierat fabrikat eller ”allmänt” fabrikat.

Några tusen komponenter eller aggregat kan tyckas vara ett relativt bra underlag för att göra jämförelser ändå. Det är också lite vanskligt att dra slutsatser om det bristande användandet av fabrikat i det här läget. Databasen har inte varit i bruk särskilt länge och huvuddelen av de anläggningsdelar som finns i den är autogenererade, överlästa från Excelark. Vi har vid överläsningarna i stor utsträckning kunnat ta tillvara de uppgifter som fanns, men det återstår en hel del översyn från användarnas egen sida. En sådan sak som att knyta komponenter till fabrikat är något som kommer att ske allteftersom, varje gång man använder en komponent.

Ett annat problem är om det ens är meningsfullt att jämföra fabrikat, på grund av de olikartade förhållanden objekten arbetar under. Två pumpar kan hantera olika flöden av olika temperatur och kvalitet och ha väldigt olika drifttider osv. En tio-i-topp-lista på felfrekvens per fabrikat skulle förmodligen inte alls ge någon rättvisande information.

Att gå hela vägen ned till komponent skulle även vara möjligt. Visserligen är varje delarbetsorder kopplad till ett aggregat, men när delarbetsordern skapas plockas arbetsbeskrivningar för aktuell komponent in från komponenten. Genom att titta på vilka arbetsbeskrivningar som har hämtats in kan man alltså ta reda på vilka komponenter man avsett att arbetet skall utföras på. Det är en omväg att hitta dem, men fullt möjligt. För skojs skull visas här sql-satsen för hur många fel som kan hänföras till speciell komponent:

```
SELECT distinct
  count(AnlKomponentTillaggGem.Fabrikat), AnlKomponentTillagggem.fabrikat,
  AnlKomponentTillagggem.namn, AnlKomponentTillagggem Gem.typnr
FROM DaoFelanmDao
  INNER JOIN DaoDelarbetsorder ON DaoFelanmDao.FelanmId = DaoDelarbetsorder.PKId
  INNER JOIN DaoDelarbetsorder1 ON DaoFelanmDao.DAOID = DaoDelarbetsorder1.PKId
  INNER JOIN HrDAOStatus ON DaoDelarbetsorder1.HrStatusBet = HrDAOStatus.Beteckning
  INNER JOIN AnlAggregat ON DaoDelarbetsorder1.AggregatId = AnlAggregat.PKId
  INNER JOIN AnlSystem ON AnlAggregat.MasterId = AnlSystem.PKId
  INNER JOIN AnlBlock ON AnlSystem.MasterId = AnlBlock.PKId
  INNER JOIN AnlAnlaggning ON AnlBlock.MasterId = AnlAnlaggning.PKId
  INNER JOIN AnlVerksamhet ON AnlAnlaggning.MasterId = AnlVerksamhet.PKId
  INNER JOIN AnlKomponent ON AnlAggregat.PKId = AnlKomponent.MasterId
  INNER JOIN DaoKompGemArbBeskr ON DaoDelarbetsorder1.PKId = DaoKompGemArbBeskr.DAOID
  AND AnlKomponent.PKId = DaoKompGemArbBeskr.KompId
  INNER JOIN AnlKomponentGemArbetsBeskr ON DaoKompGemArbBeskr.KompGemArbBeskrId =
  AnlKomponentGemArbetsBeskr.PKId
  LEFT OUTER JOIN AnlKomponentTillaggGem ON AnlKomponent.KompGemId = AnlKomponent-
  TillaggGem.PKId AND
  AnlKomponentGemArbetsBeskr.AnlKomponentGemID = AnlKomponentTillaggGem.PKID
GROUP BY AnlKomponentTillagggem.fabrikat,
  AnlKomponentTillagggem.namn, AnlKomponentTillagggem.typnr
```



Underhållskostnad i relation till nypris liksom underhållskostnad i relation till producerad energi (GW) är en svårare nöt att knäcka. I databasen finns i dag inga nypriser inlagda. Producerad energi finns inte heller. Att fördela kostnaderna på avhjälpande underhåll respektive förebyggande underhåll skulle vara fullt möjligt. Det finns ett stort antal hjälpregister som används för att kategorisera delarbetsordrarna på olika sätt. Bland annat finns ett hjälpregister för planerbarhet, som skulle kunna användas för detta ändamål.

Totalkostnad för underhåll i kronor, inklusive bortfall och stoppkostnader finns i databasen.

Vilka fel som har tagit längst tid att laga går att räkna fram. Felorsakerna finns i databasen, tidsåtgången likaså.

För såväl kostnads- som tidsberäkningar kan det finnas luckor i databasen, men de är utanför vår kontroll eftersom de i sådant fall beror på att man glömt mata in någon faktura eller något arbetspass. Det är inte möjligt för någon annan än exempelvis arbetsledaren att avgöra om ett belopp är för lågt eller ett tidspass för kort.

Mantimmar fördelat på extern personal/intern personal per anläggningsdel blir svårare igen. Mantimmar finns, per yrkeskategori (elektriker, mekaniker osv). Men om denna resurs består av extern personal eller egen personal framgår inte av delarbetsordern.

### Ökade kostnader

|  |  |   |  |
|--|--|---|--|
| Vilka anläggningsdelar har ökat mest när det gäller kostnad eller antal åtgärder under viss period |  | 1 |  |
|--|--|---|--|

Detta är ytterligare en "tio-i-topp". Den skulle utformas som en graf som börjar på verksamhet (eller kanske anläggning) och visar kurvor månad för månad för de exempelvis 5 anläggningsdelarna som ökat mest i fråga om kostnad eller antal åtgärder. Ökat mest bör förmodligen beräknas från kurvans lägsta punkt till dess högsta, inte från början av perioden till slutet. Sedan ska man kunna klicka på en intressant kurva och då i stället få upp nästa nivå nedåt, och de fem värsta i den gruppen. Osv! Man ska också kunna bläddra till nästa fem och nästa fem osv. När man kommer ned till aggregatsnivå och klickar på en aggregatkurva får man upp en lista över de delarbetsorder som passar in i vald period.

### Databasen

Denna graf är fullt möjlig att ta fram, med enkel matematik och ett intensivt räknande.

### Jämförelser

|   |  |   |   |
|---|--|---|---|
| Jämförelse mellan olika fabrikat              | Drifftid, belastning osv, hur länge höll de? Hur mycket fel och vad har de kostat? | 2 | 5 |
| Jämförelse mellan likvärdiga anläggningsdelar |  | 1 |   |

Sex informanter ville ha felfrekvens per fabrikat i form av tio-i-topp-lista, två ville kunna göra jämförelser mellan specifika fabrikat. Alla åtta förde vissa resonemang kring värdet av sådana jämförelser eller listor. Det fanns en viss form av motstridighet här – de ville verkligen kunna

jämföra fabrikaten, och såg samtidigt problemen i det. Vissa fabrikat täcker in en stor mångfald av olika apparatur. Att bara jämföra fabrikat rakt av kan bli en lustig jämförelse mellan myra och elefant. Vidare, att jämföra två pumpar av olika fabrikat låter sig inte heller göras rakt av, eftersom de möjligen arbetar under mycket olika förhållanden.

En informant föreslog jämförelser av likvärdiga anläggningsdelar. De kan då vara av olika fabrikat eller samma, men sitta på likvärdiga ställen i anläggningen.

### Databasen

En sådan jämförelse skulle vara möjlig. Här gäller samma problem som under tio-i-topp-rubriken – att underlaget har luckor. En jämförelse skulle ändå kunna vara rimlig, om man exempelvis låter användaren peka ut vilka två (eller flera) likvärdiga pumpar han vill jämföra. Det gäller då att komma ihåg att om samma pump förekommer flera gånger i anläggningen kan inte antalet felanmälningar jämföras med antalet felanmälningar på en pump som bara förekommer en gång.

### Arbetsrapporter – statistik över delarbetsorder/felanmälningar

|  |   |   |      |
|--|---|---|------|
| Information kring delarbetsorders och felanmälningar | Statistik över hur det går, felanmälningar respektive alla ingrepp. Hur många nya, hur många avklarade osv. | 3 | 8, 9 |
|  | Hur många felanm med vilken prioritet?  | 2 |      |
|  | Hur många återrapporterade  | 1 |      |
|  | Hur många vilande felanm?   | 1 |      |
|  | Hur många har vi klarat av?   | 1 |      |

Tre personer i ansvarsställning tyckte det kunde vara viktigt att se statistik över hur lång tid det tar från felanmälan till avslutad delarbetsorder. Intressanta frågeställningar var också hur många nya felanmälningar som har kommit, hur många som man har klarat av, hur många som ligger och dammar, osv.

### Databasen

Tid från felanmälan till färdig delarbetsorder kan mycket enkelt beräknas med hjälp av start- och sluttid på delarbetsorder. Att räkna nyinkomna felanmälningar respektive delarbetsorders möter inte heller några problem. Ett enkelt sätt att se om de är återrapporterade, avslutade eller finns någonstans i det aktiva förloppet är att titta på dess status. Är status 13 är den återrapporterad.

### Händelserapport

|                |  |   |   |
|----------------|--|---|---|
| Vad-har-gjorts | Underlag som visar vilka åtgärder som gjorts på anläggningsdel under viss period | 1 | 6 |
|----------------|--|---|---|

Detta är en enkel listning av berörda delarbetsorders.

#### Databasen

En delarbetsorder relaterar alltid till ett aggregat. Men i det fall delarbetsordern inte avser arbete på aggregatet utan på ovanliggande system är aggregatet angivet till OGEM. Om arbetet skall utföras på blocknivå är delarbetsordern knuten till aggregatet OGEM på systemet OGEM på aktuellt block. På det viset kan rätt delarbetsorders sorteras fram. Vill man veta vad som har gjorts på nivån under aggregatet, nämligen komponentens är det lite tricksigare. Det är dock möjligt på ett indirekt sätt. Arbetsbeskrivningar är alltid knutna till en komponent. För varje komponent under aggregatet kan det finnas en lång rad arbetsbeskrivningar. När delarbetsordern skapas kommer beredaren att hämta in de arbetsbeskrivningar som är aktuella, och man kan på det viset se vilken komponent delarbetsordern berör.

Ett problem är naturligtvis i det sammanhanget att delarbetsordern kan avse arbete på flera olika komponenter på aggregatet. Det innebär att man kan se vad som gjorts på komponentnivå, man kan få en listning på aktuella delarbetsorders, men man kan inte göra någon beräkning av kostnaderna för en specifik komponent. Delarbetsorderns kostnadsberäkning kan innefatta kostnader för flera komponenter.

#### Vad sysslar vi med för typ av arbete?

|   |                                 |   |   |
|---|---------------------------------|---|---|
| Planerbarhet, hur har de utvecklats beträffande kostnad/antal |                                 | 4 | 7 |
| Planerbarhet, hur fördelar de sig under viss period           | Vad sysslar vi med för arbeten? | 4 | 7 |

Det finns i SiRi en parameter som kallas planerbarhet. Det finns i princip idag tre olika grader eller typer av planerbarhet, nämligen akut, planerat, utryckning/beredskap. Dessutom en "okänd" och en "odefinierad" som här får gå som "övriga". Här önskar man sig en graf (kurvor) med antal eller kostnad på y-axeln och tid på x-axeln. Grafen visar kurva för akut, planerat, beredskaps (arbetstyper) och övrigt för viss period. Börjar akutkurvan stiga skulle det visa att något är fel. Eventuellt skulle man kunna välja t ex max 4 andra parametrar till denna graf från teknikområde, typ av arbete och övriga hjälpregister.

"Vad sysslar vi med", det vill säga en graf som i stället för planerbarhetstypernas utveckling visar deras fördelning under viss period.

#### Databasen

Varje delarbetsorder är märkt med en planerbarhetstyp – det är bara att räkna! Ett problem i sammanhanget kan vara om många delarbetsorder knyts till typen "okänd" eller "odefinierad", vi får då en sorts lucka i materialet.

## Periodrapporter

|                 |                                    |   |  |
|-----------------|------------------------------------|---|--|
| Periodrapporter | Drifftid per anl.del               | 3 |  |
|                 | Bränsleförbrukning                 | 1 |  |
|                 | Mantimmar                          | 3 |  |
|                 | Vilken tid har lagts ned per andel | 1 |  |
|                 | Totalkostnad per andel             | 2 |  |
|                 | Intern/extern personal             | 1 |  |
|                 | Material                           | 1 |  |
|                 | Var prel felorsak den rätta        | 1 |  |
|                 | Var tidsplanering rätt             | 2 |  |

Arbetsledare var inte speciellt intresserade av periodrapporter. Det var däremot beredare och ansvariga för beredning.

Här återkom i stort sett samma punkter som under tio-i-topp-listan. Drifftid, mantimmar fördelat på extern och intern personal, bränsleförbrukning, materialkostnad osv – allt per anläggningsdel.

Här kom också upp en del statistik över hur det går. Man ville veta hur många felanmälningar man haft under perioden, hur många man klarat av, hur många som ligger vilande respektive är pågående.

Vidare fanns det intresse av att ta reda på hur bra man är på att diagnostisera – har förmodad felorsak i ursprunglig felanmälan visat sig vara den faktiska felorsaken? Med en sådan kunskap skulle man kunna planera en mer riktad vidareutbildning för personalen. Listan skulle inte vara ned på personplan, men om man kunde konstatera att den förmodade felorsaken kanske alltid var fel för ett visst aggregat skulle man kunna dra slutsatsen att personalen vet för lite om just detta aggregat.

På motsvarande sätt fanns det intresse av att ta reda på hur bra man är på att planera tidsåtgången. Stämde resursplaneringen, tog det längre tid eller kortare tid i själva verket?

### Databasen

Drifftid per anläggningsdel kommer i viss utsträckning att finnas, men är beroende av att det verkligen knappas in. Möjligen skulle dessa siffror kunna hämtas från en annan databas, ProdSim. Vad gäller bränsleförbrukning finns inga sådana fält i SiRis databas – även där skulle man möjligen i framtiden kunna hämta information från ProdSim.

Mantimmar finns i SiRi, återigen med förbehållet att det inte framgår om arbetad tid är arbetad av intern eller extern personal.

Statistiken över felanmälningar är en enkel operation. Hur många som finns, hur många som är vilande, hur många som avslutats osv under viss period räknas enkelt ut. Delarbetsordern är försedd med en dynamisk status som ändras automatiskt allteftersom den förs framåt genom systemet. Det är alltså bara att studera status.

Tid, totalkostnad och material per anläggningsdel är relativt omfattande räkneoperationer men alla önskvärda fält finns i databasen.

Utredning kring eller jämförelse mellan förmodade respektive faktiska felorsaker och förmodad respektive faktisk tidsåtgång är relativt enkla räkneoperationer direkt ur databasen.

### Drifftidsrelaterad fellista

|                             |   |   |  |
|-----------------------------|---|---|--|
| Drifftidsrelaterad fellista | Felen på en anläggning under viss drifftid. Är felet beroende av drifftid? Mängd bränsle och flödesmätning kan också inverka. Har vi likadana fel? Vad kostar det att laga? | 1 |  |
|-----------------------------|---|---|--|

Det här förslaget handlar om en rapport som visar felen på en anläggningsdel under viss drifftid. Men felen är inte alltid beroende av drifftid - mängd bränsle och flödesmätning kan också inverka. Rapporten måste därför även ta med sådana parametrar.

Vidare skulle den här rapporten ge svar på frågor som: har vi samma fel om och om igen och i sådant fall – i vilket förhållande står dessa fel till drifftiden? Vad kostar det att laga dessa upprepade fel?

Vad man är ute efter här är att försöka få hjälp att hitta underhåll som borde bli förebyggande i stället för att man ständigt är tvungen att göra avhjälpan underhåll.

### Databasen

Denna lista är mer komplex. Felen på en anläggning under viss drifftid kan beräknas, så länge drifftiden finns inknappad. Om felet beroende av drifftid kan inte hämtas ur databasen, varför mängd bränsle och flödesmätning med flera liknande parametrar behöver tas med. Dessa finns endast i viss omfattning (flödesmätning) i SiRi. Bränsleförbrukning kan i en framtid tänkas hämtas från ProdSim.

Om konstaterade fel, faktiska fel, under angiven period är av samma slag kan naturligtvis visas – eller snarare i hur hög grad de är av samma slag. Kostnaden för dessa fel och tidsåtgång för att åtgärda felen kan också räknas fram ur databasen.

### Presentationens utseende

|                 |                    |    |  |
|-----------------|--------------------|----|--|
| Form av rapport | Grafer             | 12 |  |
|                 | Tabeller           | 12 |  |
|                 | Intranät           | 12 |  |
|                 | Utskrifter         | 12 |  |
|                 | Siffror till Excel | 3  |  |

Samtliga informanter önskade sig grafer för att kunna presentera för sina medarbetare och chefer, tabeller för att kunna se mer i detalj för egen räkning och möjlighet att skriva ut såväl grafer som tabeller på papper.

Det fanns också i några fall önskemål om att kunna få ut siffrorna i Excel, för att kunna pula vidare på egen hand.

Man önskar sig full valmöjlighet när det gäller utseendet på rapporter / listor / underlag.

Tabeller på intranät är inga problem. När det gäller grafiken kan den åstadkommas med en ActiveX-komponent som skapar grafer i Excel och lagrar på hårddisken i lämpligt bildformat.

Utskrifter kan alltid fås direkt från web-sida, med låsta fontstorlekar. Ett alternativ är att koda utskriften i Visual Basic, med hjälp av Crystal Reports och låta web-sidan köra ett exekverbart program.

Att få resultatet av sin förfrågan i form av siffror på ett Excel-ark låter sig göras med hjälp av ActiveX-komponent.

### 6.3. Lösningförslag

Ur informanternas förslag plockades följande 13 rapporter fram, såsom varande bland de mest önskade. Listnummer 10 och 11 slogs ihop till en rapport. Listnummer 3 och 5 utgick, se kommentarer nedan. Listnummer 7 ingår delvis i listnummer 13. De nio förslag som detta resulterade i presenterades vid ett möte där informanterna fick ytterligare en chans att göra sin röst hörd. Siffran inom parantes visar hur många som önskat just denna lista.

#### "Tio-i-topp"

1. Störst kostnad i nominellt belopp (8)
2. Mest använd tid (4)
3. Frekvens felanmälningar per fabrikat (6)
4. Störst antal felanmälningar (8)
5. Störst kostnad i förhållande till nypris (4)
6. Ökning av kostnader eller ökning av antal åtgärder (4)
7. Frekvens ingrepp (inte bara fel) (3)
  
8. **Jämförelse mellan olika fabrikat**, högst fyra objekt (fel, drifttid, kostnader...) (8)
9. **Vad har gjorts på en anläggningsdel under viss period?** (1)
10. **Arbetstyper, hur har de utvecklats beträffande kostnad/antal?** (4)
11. **Arbetstyper, hur fördelar de sig under viss period?** (4)
12. **Felanmälningar: aktuellt antal, nya under viss period (fördelade på status/prio)** (3)
13. **DAO: aktuellt antal, nya under viss period (fördelade på status/prio)** (3)

Följande kommentarer gavs: Frekvens felanmälningar per fabrikat var ofta nämnd. En sådan tio-i-topp-lista skulle förmodligen i praktiken inte ge vad man tänkt sig. Två pumpar av olika fabrikat kan t ex ha kolossalt olika förutsättningar. En direkt jämförelse mellan två specifikt utvalda pumpar med likvärdiga förutsättningar i anläggningen borde däremot kunna ge sådan information. Se i stället lösningförslag nr 8.

Idag finns inte fält för nypris i databasen. Listan kunde dock vara intressant, eftersom den förmodligen ger en bättre uppfattning om ett belopps proportioner.

Många ville se arbetskostnader fördelat på extern och intern personal. Detta är inte möjligt i SiRi idag.

#### Förslag 1 – 4. "Tio-i-topp"

För samtliga tio-i-topp listor gäller att man väljer en period och en nivå i anläggningsstrukturen att starta på.

Man kan därefter bläddra sig vidare nedåt i strukturen. Från verksamhet till anläggning, från anläggning till block, från block till system osv.

För samtliga val nedan gäller även att man får upp diagram för de fem översta i det val man gjort, men man kan därefter bläddra sig vidare nedåt till nästa fem, och nästa....

Alternativ

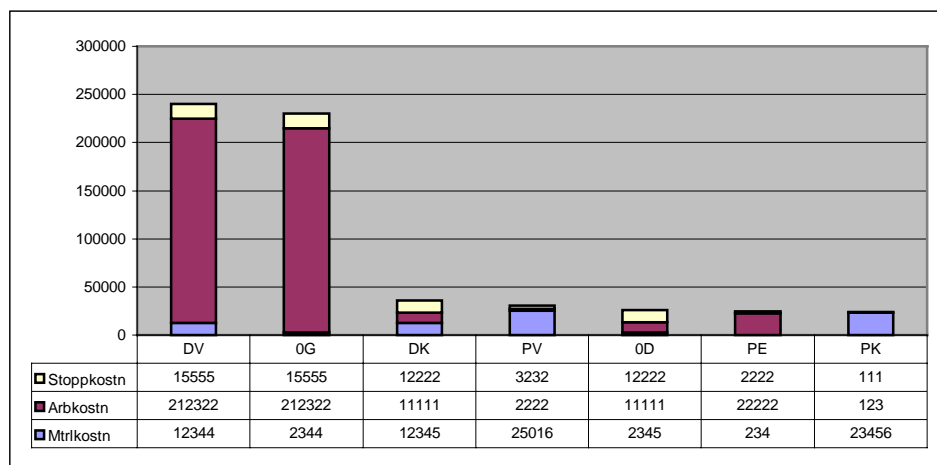
- \* Störst kostnad i nominellt belopp
- \* Mest använd tid
- \* Störst antal felanmälningar
- \* Ökning av kostnader eller antal åtgärd

OBS! Alla siffror är kolossalt påhittade!!

Vi väljer perioden och får till att börja med upp en lista och två bilder.

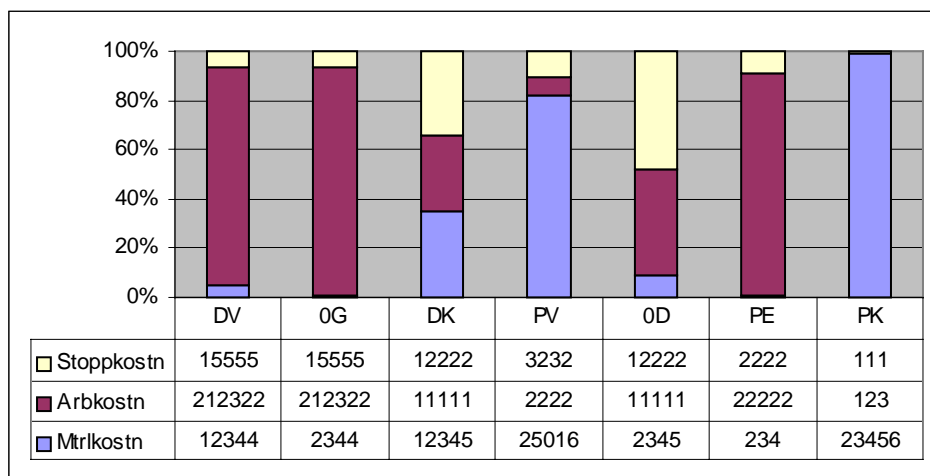
Listan visar ordningsföljden på verksamheterna. Här klickar man på önskad rad för att bläddra vidare.

| Verksamhet                | Kostnad |
|---------------------------|---------|
| DV Distribution värme     | 240 221 |
| OG Gemensam verksamhet    | 230 221 |
| DK Distribution kyla      | 35 678  |
| PV Produktion värme       | 30 470  |
| OD Odefinierad verksamhet | 25 678  |
| PE Produktion el          | 24 678  |
| PK Produktion kyla        | 23 690  |



I det första diagrammet ses alla verksamheter (eftersom de är så få) med den totala kostnaden fördelad på stoppkostnad, arbetskostnad och materialkostnad.

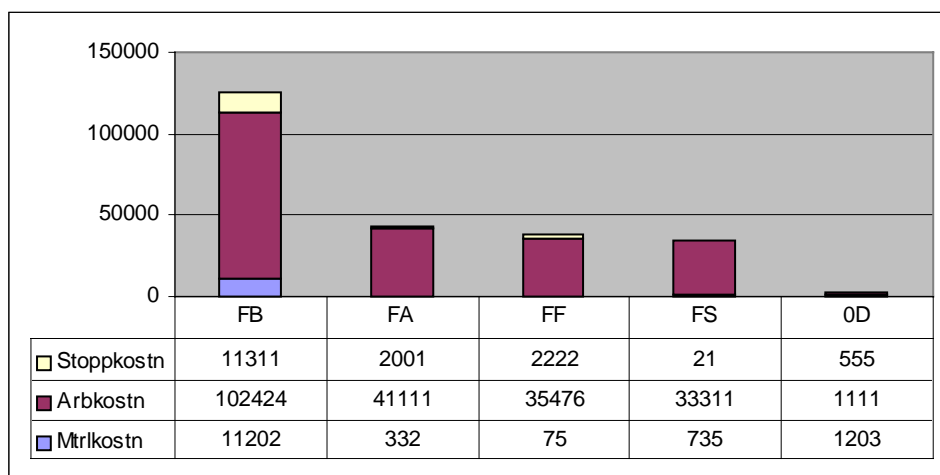




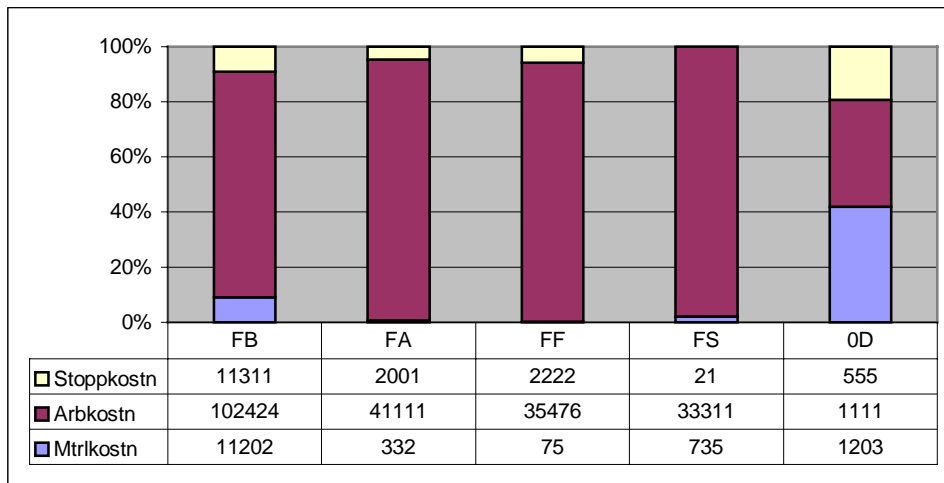
I det andra diagrammet ges ungefär samma information, men här ser vi andelen stoppkostnader, arbetskostnader och materialkostnader i stället.

Nu undrar man intresserat vad som hänt på DV, och väljer därför att gå vidare med att titta på anläggningarna på DV. Vi får återigen en lista att så småningom välja vidare i. För de fem högsta värdena får vi motsvarande två diagram som för verksamheterna:

|    |                        |         |
|----|------------------------|---------|
| FB | Borås                  | 124 937 |
| FA | Dalsjöfors             | 43 444  |
| FF | Fristad                | 37 773  |
| FS | Sandared               | 34 067  |
| OD | Odefinierad anläggning | 2 869   |
| OG | Gemensam anläggning    | 2 284   |



Totalkostnaderna fördelade (ovan) som tidigare och andelen (nedan).



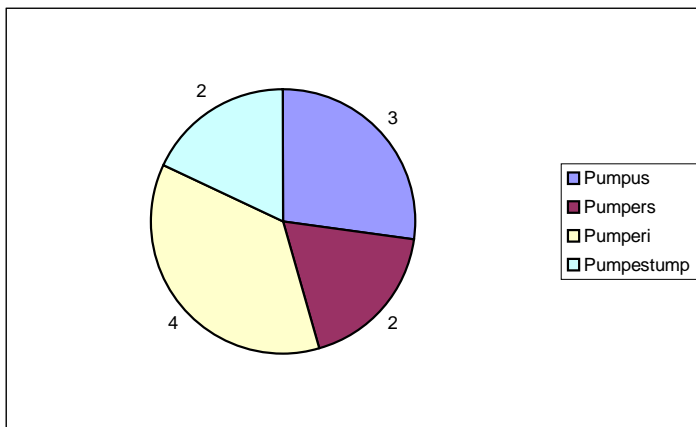
På samma sätt kan man fortsätta till blocknivå, systemnivå och aggregatnivå. När listan över aggregaten kommer upp kan man fortfarande välja det aggregat som intresserar mest, men i stället för att få upp lista över komponenterna får man här en lista över de delarbetsorders (lustigt ord, men korrekt enligt svenska akademins ordlista!) som åstadkommit kostnaden.

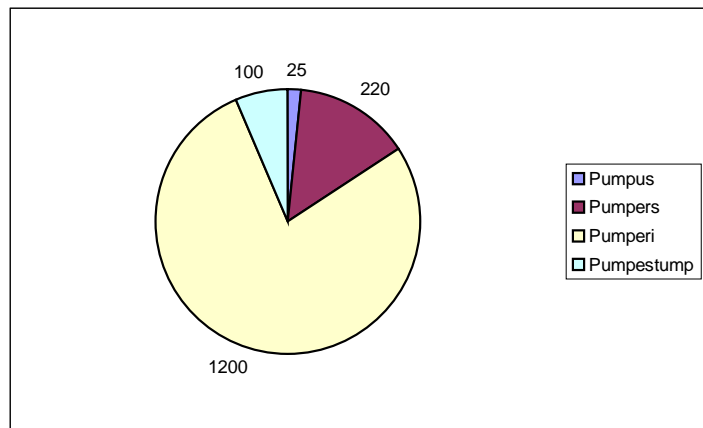
### Förslag 5. Fabrikatjämförelser

Tanken är här att man kan välja ut högst fyra komponenter ur en lista över gemensamma komponenter och få jämförelsematerial för dem. På det viset kan man välja att jämföra komponenter som man vet är likvärdiga och utför ett likvärdigt arbete.

Resultatet skulle kunna bli:

### Antal felanmälningar på fyra pumpar under vald period



**Underhållskostnad för fyra pumpar under vald period****Förslag 6. Vad har gjorts på viss anläggningsdel under viss period?**

Efter att ha valt önskad period väljer man anläggningsdel i en lista och får upp en lista över periodens delarbetsorder. Eftersom en felanmälan eller delarbetsorder kan skapas under perioden men arbetet vara ogjort när perioden är slut tas endast sådana som avrapporterats under perioden med.

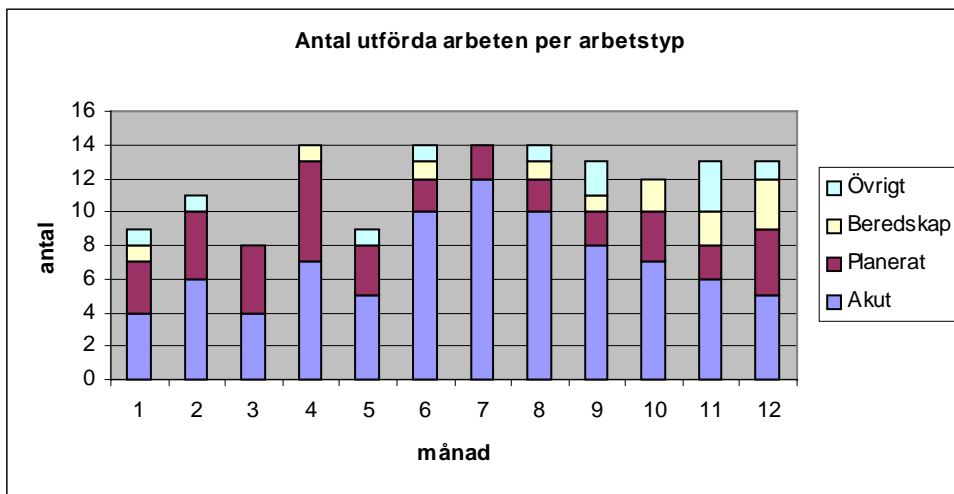
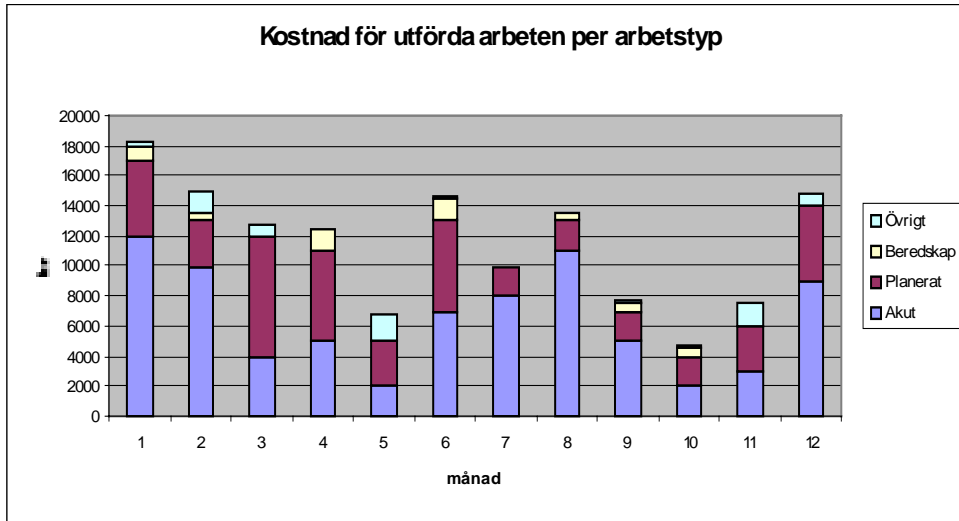
Resultatet kan kanske se ut så här:

**Åtgärder utförda på PV|SD=HP1|11|PP0001, Returledn.cirk.pump.HP1**

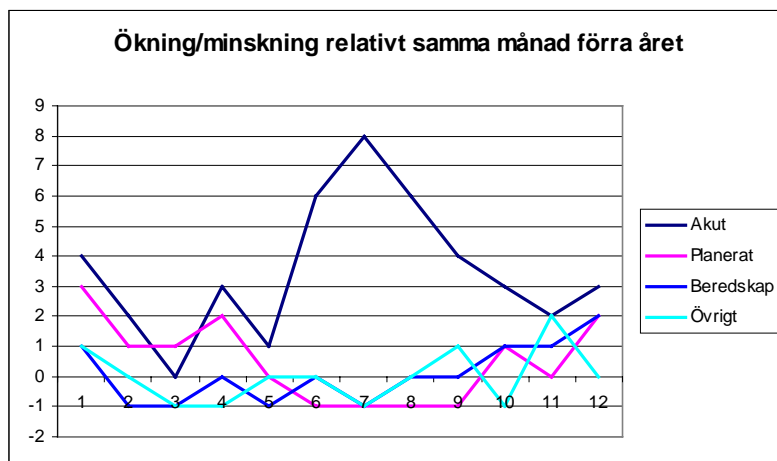
| Aonr  | Avslutad datum | Åtgärd                  |
|-------|----------------|-------------------------|
| 11111 | 000301         | Byte av kylvattenventil |
| 11021 | 000305         | Översyn                 |
| 11021 | 000307         | Läckage                 |

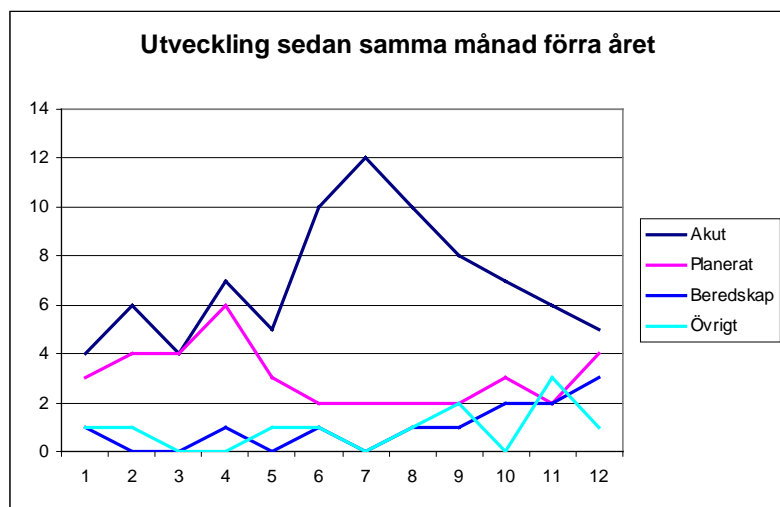
**Förslag 7. Vilken sorts arbete har vi hållt på med?**

Den här rapporten visar vilken typ av arbete man har ägnat sig åt de senaste tolv månaderna. Man kan alltså inte välja rapport, men naturligtvis anläggningsdel. Förslagsvis kan man också välja mellan att se antal utförda arbeten eller kostnaden för de olika arbetstyperna. Detta för att ett enda stort arbete kan kosta många gånger mer än tjugo småarbeten i vissa fall. Att endast titta på antal eller endast på kostnad kan ge ett felaktigt intryck.



För att lättare kunna se utvecklingen av de olika arbetstyperna kan man också åskådliggöra utvecklingen dels i relation till första månaden i denna serie, dvs för ett år sedan, dels utvecklingen i faktiska siffror sedan samma månad.





### Förslag 8. Felanmälningar och förslag 9. Delarbetsorder

De här två förslagen är mycket lika varandra. I båda fallen föreslås cirkeldiagram som visar aktuellt läge, totalt antal felanmälningar/delarbetsorder fördelat på status respektive prioritet. Dessutom cirkeldiagram som visar nyinkomna felanmälningar/delarbetsorder under vald period, även här fördelat på status respektive prioritet.

### 6.4. Respons

Som redan nämnts var inte responsen överväldigande. Några tyckte att ”det här har vi aldrig haft förut, så vad ska vi med det till? Det har väl alltid gått bra ändå?”. Arbetsgruppsdeltagarna säger att på båda avdelningarna upplevs det som en svårighet att ha åsikter om något som än så länge bara finns på papper. Man vill kunna se det ”på riktigt”, klicka och prova. Vidare har man en känsla av att man kommer att få behov av de här rapporterna ”längre fram”, när det faktiskt finns underlag i SiRi. Man har en smula svårt att se framåt.

Det som vägt tungt vid urvalet har varit att hitta de rapporter som ger en snabb överblick. Man har tyckt att de rapporter som direkt visade ekonomiska faktorer (pengar, tid, felfrekvens) var viktigare än de mer strategiska som handlade om jämförelser, intern funktionalitet osv.

### 6.5. Rapportval

De rapportförslag man fastnade för var förslagen nr 1 – 3 som rör störst kostnader, mest använd arbetstid och flest felanmälningar per anläggningsdel i tio-i-topp-version. Dessutom förslag nr 6, dvs vad som gjorts på viss anläggningsdel, och förslag nr 7 som presenterar arbetstidens fördelning mellan olika typer av arbete.

D-avdelningen hade gärna sett en tio-i-topp-lista över frekvens ingrepp, där alltså inte bara felanmälningar utan även andra arbeten fanns med. Denna variant fanns inte med bland de slutliga förslagen.

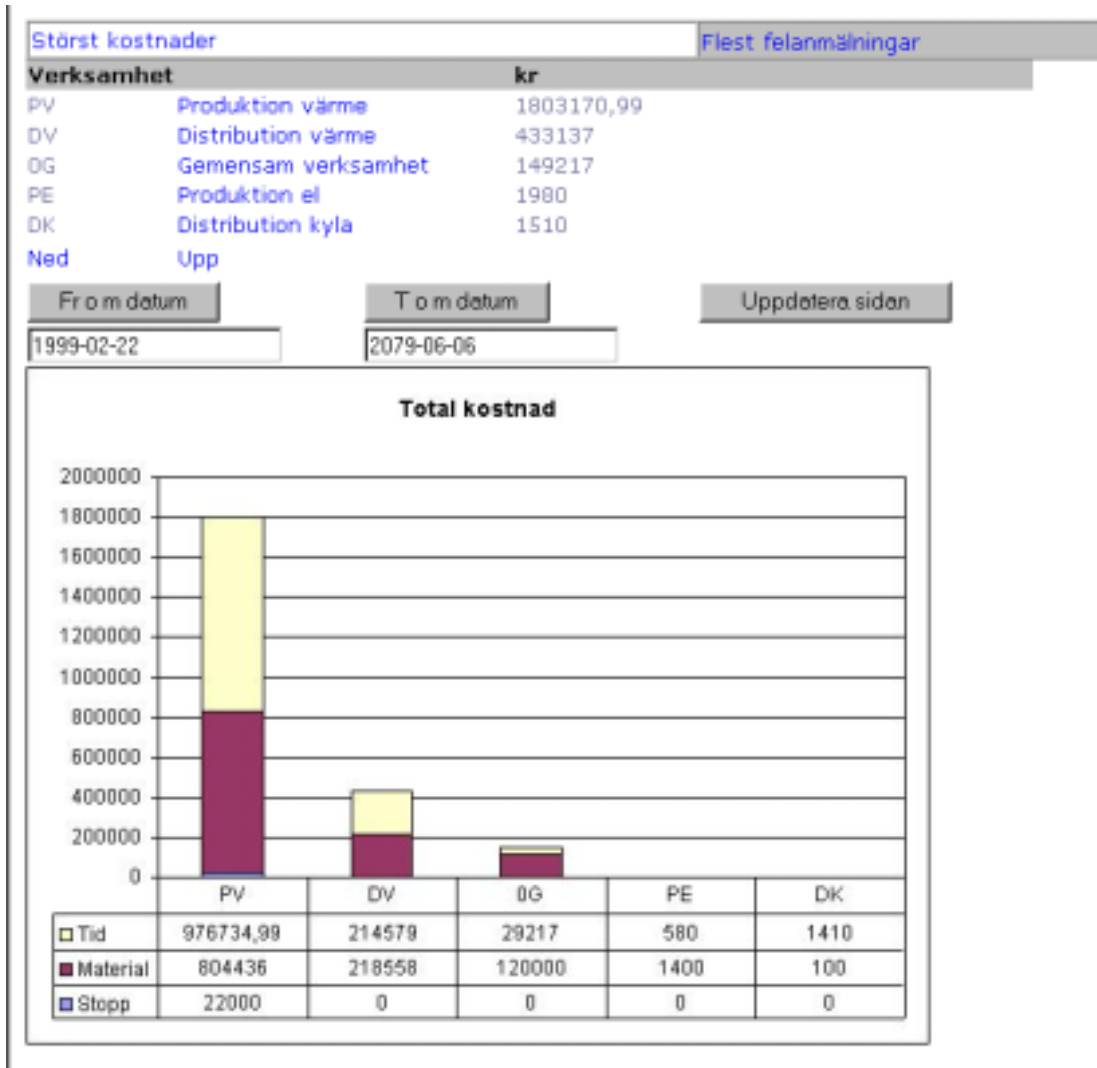
## 6.6. Rapporterna

I skrivande stund finns rapporterna i en första version på Borås Energis intranät. Svårigheterna att få något intryck från pappers-förslagen har gjort det nödvändigt att få fram en första intranät-version och låta SiRi-användarna titta på den. Dessa rapporter kan nu testas av var och en och förhoppningsvis kommer detta ge upphov till respons och/eller en fortsatt dialog om hur rapporterna slutligen skall se ut. Den här första versionen får möjligen betraktas med ett visst överseende - den är inte fullständigt "klar". Detta beror främst på att det tagit mycket längre tid än förväntat att få respons på lösningsförslagen.

Så här ser rapporterna ut nu. Först ses integreringen med befintliga SiRi – vi hittar rapporterna under rubriken "Rapporter" på SiRis hemsida.

| Category               | Date and Time      | Description  |
|------------------------|--------------------|--|
| <b>Startsida</b>       | 2000-05-03 00:28:2 |  |
| <b>Huvudprogram</b>    | 2000-05-03 00:28:2 | Med kopiera DAO, nya och enskild DAO direkt i listan |
| <b>Utförarprogram</b>  | 2000-05-03 00:28:2 | åtgärdade, vid problem Ri                            |
| <b>Felanmälan</b>      | 2000-05-02 15:29:3 |  |
| <b>Delarbetsorder</b>  | 2000-05-02 15:29:3 | SiRi stoppas vid två tillfälle                       |
| <b>Anläggning</b>      | 2000-05-02 15:29:3 | vi för att uppgradera minn                           |
| <b>Beskrivningar</b>   | 2000-05-02 15:29:3 | vi för att läsa in ledningsr                         |
| <b>System</b>          | 2000-04-27 02:34:3 |  |
| <b>Rapporter</b>       | 2000-04-27 02:34:3 | Driftorder fjärrvärme, OBS                           |
|                        | 2000-04-27 02:34:3 | Hälsningar Birgitta :))                              |
| <b>Tio-i-topp</b>      | 2000-04-27 02:32:4 |  |
| <b>Utförda arbeten</b> | 2000-04-27 02:32:4 | Nya urvalsalternativ i anlä                          |
| <b>Arbetstyper</b>     | 2000-04-27 02:32:4 | komponent / Hälsningar B                             |
|                        | 2000-04-27 02:32:4 | (mer minne kommer snar                               |
|                        | 2000-04-27 02:30:3 |  |
|                        | 2000-04-27 02:30:3 | Max urval och en fil/för d                           |

De tre "Tio-i-topp-rapporterna" är av samma typ, bara ur tre olika perspektiv – störst kostnad, flest felanmälningar respektive mest ianspråktagen tid. Dessa tre fick därför en sida, där man enkelt väljer vilket perspektiv man är intresserad av.



Utgångspunkten var att rapporterna skulle likna varandra så att inte varje ny rapport kräver nya ansträngningar att förstå hur man ska bete sig. Alla fem rapporterna har därför samma sökmetod överst. Man kan klicka på någon verksamhet och får då upp en ny sida med den verksamhetens alla anläggningar, och en graf för dem. Vill man längre ned i strukturen klickar man på en anläggning och får upp den anläggningens block, osv. Om det finns fler verksamheter än fem kan man klicka på "Ned" och kommer nedåt i listan över verksamheter.

Vilken tidsperiod rapporten skall visa väljs genom att man klickar på en knapp och får upp en liten kalender.

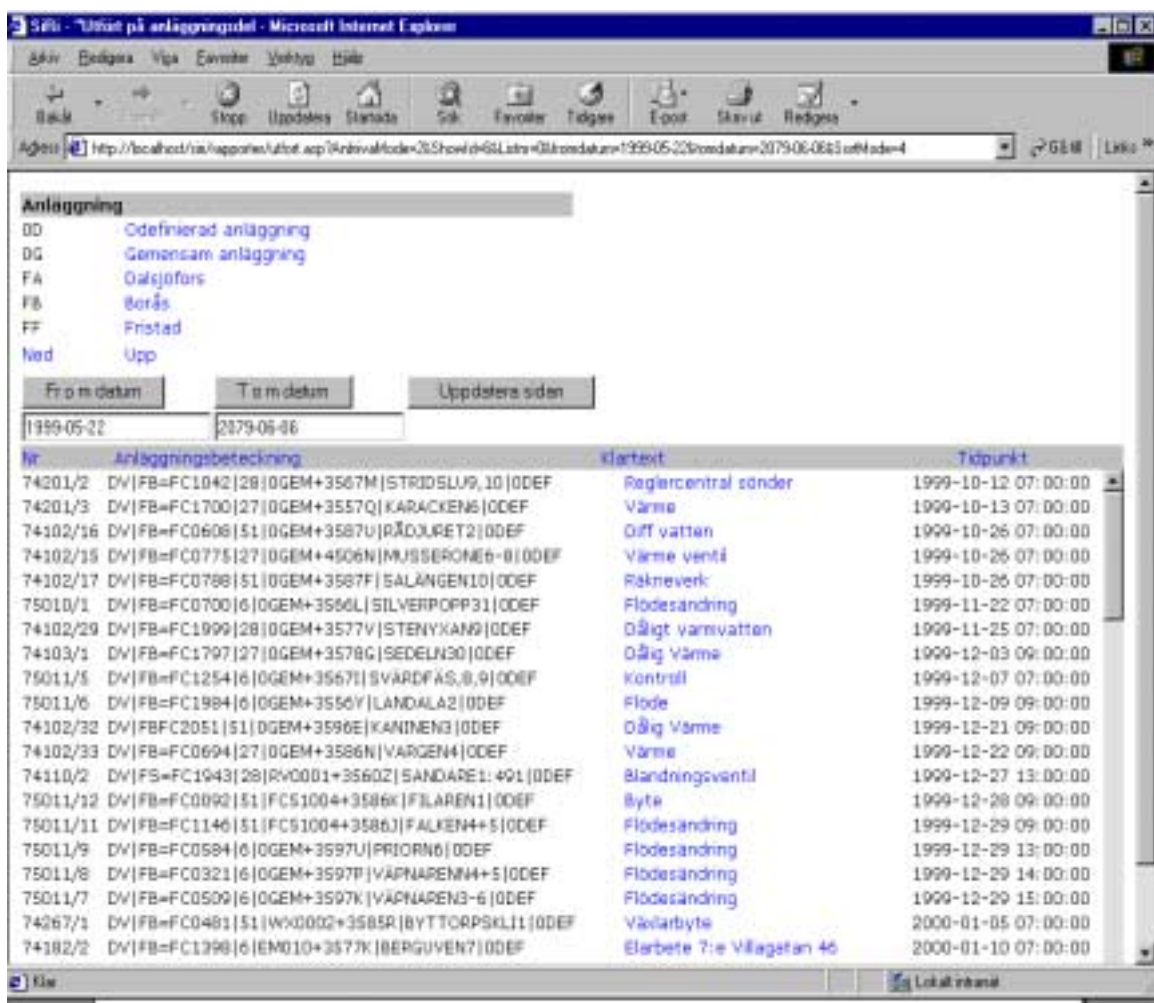
Vid uppstarten av samtliga rapporter visas en period som startar 30 dagar före dagens datum och slutar 2079-06-06. Samtliga rapporter startar överst i anläggningsstrukturen med verksamheterna.

Vad som utförts på någon speciell anläggningsdel blev en relativt enkel lista. Delarbetsordrarna listas scrollbart i en egen avgränsad region på sidan, så att anläggningsinformationen hela tiden kan ses överst på sidan. Rubrikerna är klickbara för ändring av sorteringsordning. Anledningen till att rubrikerna över delarbetsorderlistan hänger lite på svaj för tillfället är att rubrikerna och själva listan fick placeras i olika avsnitt på sidan,

för att scroll-barheten inte skulle innefatta rubrikerna. Naturligtvis ska det inte se ut så utan kommer att åtgärdas snarast.

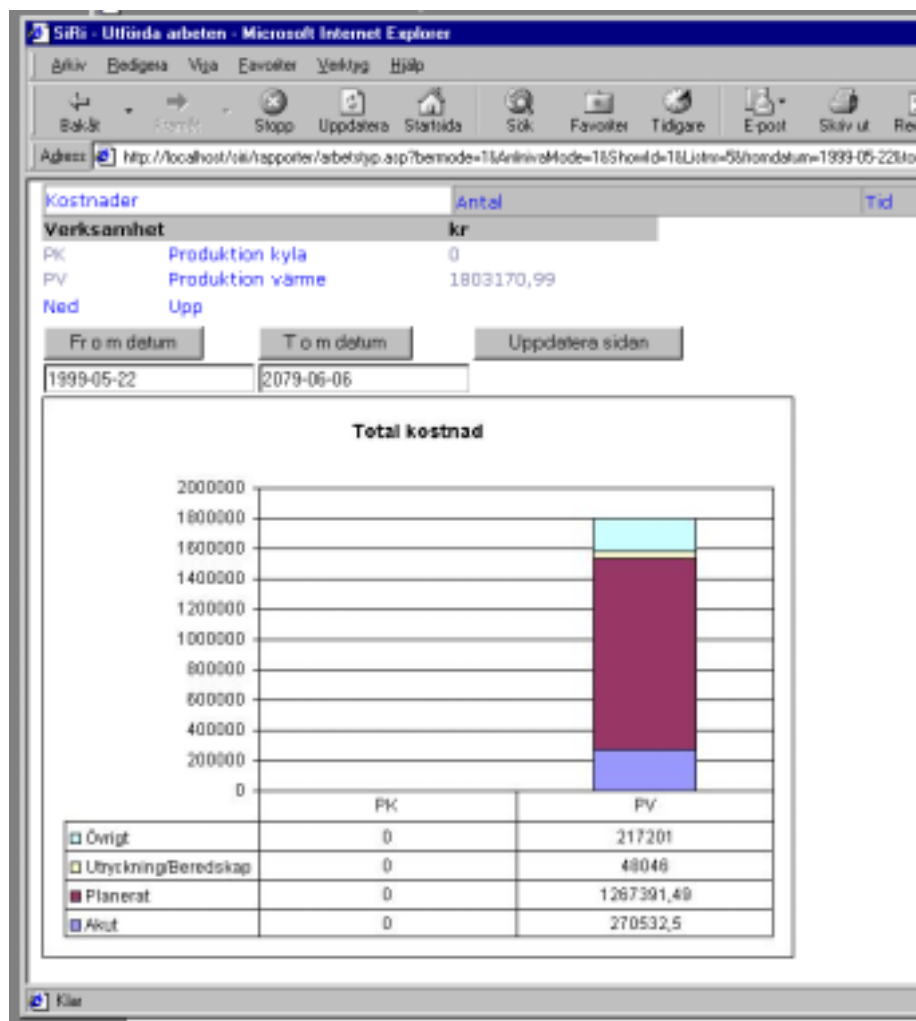
Endast avslutade delarbetsorder tas med i listan.

Vill man se någon specifik delarbetsorder kan man klicka på klartexten och få upp delarbetsordern i detalj i ett eget fönster.



Slutligen har vi listan över vilken typ av arbete man har ägnat sig åt. Som synes är den väldigt lik de övriga. Här finns det i princip tre listor i en, på samma sätt som i tio-i-topp-rapporten. Man kan välja att se kostnaderna, antalet delarbetsorder eller hur mycket tid man lagt ned fördelat på de olika arbetstyperna.





## 7. Diskussion

### 7.1. Beslutstödetts innehåll

Ett DSS kan sägas bestå av en datamodell, en logikmodell och ett användargränssnitt (Finlay, 1994). Detta är också en beskrivning av resultatet i denna uppsats. Vi har datamodellen som består av SiRis stora datamängder. Vi har logikmodeller, som utgörs av olika frågor ställda till databasen – i det här fallet oftast i form av sql-frågor. Till slut har vi också ett användargränssnitt, som utgörs av ActiveX-komponenterna och sidan på Intranätet.

Intressant att notera är att vi fick rapporter av det slag som Finlay kallar MIS – nämligen rapporter som producerar information. Här finns inga slutsatser dragna eller förslag på lösningar. Här finns inte ens några direkta förklaringar. Det man får är information om vad som är dyrast, tagit mest tid osv. Tanken var inte att finna svåra mönster i databasen eller dra avancerade slutsatser utifrån våra beräkningar. Men även så är det intressant att informanterna inte på någon punkt föreslår något sådant. Deras förslag och deras slutgiltiga val av rapporter rör sig kring de allra enklaste summerande rapporterna.

Detta väcker två frågor hos mig. Den ena är helt enkelt om jag har påverkat dem åt det hållet – omedvetet, eftersom jag ansåg att mitt arbete inte rörde de mer komplexa frågorna. Den andra frågan anknyter till något som Bo Dahlbom har sagt (föreläsning hösten 1999): nämligen att även om användarnas medverkan är av största betydelse behöver systemutvecklaren förstå bättre än användaren själv vad han vill ha och behöver. Det kan låta lite arrogant att säga sig veta mer än användaren om hans behov och önskemål, men denna vetskap baserar sig på en större kunskap om möjligheterna i systemet och tekniska möjligheter. Systemutvecklaren kan möjligen bättre se möjligheterna, för hon ser det hela ur en lite annan synvinkel.

Fyra av fem logikmodeller i det slutliga resultatet är rent summerande. Vi summerar kostnader, tid, felanmälningar och delarbetsorders med viss arbetstyp angiven. Den femte visar egentligen bara upp vilka ingrepp som gjorts på en viss anläggningsdel under en viss period. Den femte är lite speciell på det viset att den information som ges är möjlig att få på andra sätt i SiRi, men man har ändå funnit den viktig att ha lättåtkomlig i rapportmodulen. Detta poängterar åter igen den betydelse snabbhet och lättåtkomlighet har för användarna.

Poängteras bör också att önskelistan från användarna bestod i hög grad av rapporter för operationella beslut snarare än ledningsbeslut. Dessutom valde man när rapportförslagen kom att låta personalen på operationell nivå besluta om vilka rapporter som skulle byggas. Man kan säga att man valde att i första hand lägga beslutstödet på operationell nivå. Genom att bygga beslutstödet baserat på anläggningsstrukturen blev resultatet ändå ett stöd som kan tillfredsställa informationsbehovet på alla nivåer i organisationen. Ledningen kan titta på verksamheternas kostnader medan operativ personal kan gräva sig nedåt i materialet till en detaljnivå som rör enstaka ventiler.

Innehållet i ett beslutstöd skall enligt Little (Turban & Aronson, 1998) bland annat ge "viktig och relevant information". Den sortens uttryck förefaller så självklara när man först hör dem. Det är inget att ens resonera omkring, kanske man tänker. Men så fort man börjar arbeta med en riktig situation inser man att det inte är så självklart och enkelt. Principen att informationen skall vara relevant och viktig är nog så självklar, men vad *är* viktig information? Visa kvinnor och män skriver tjocka böcker om företagsekonomiska begrepp, organisationsförståelse och

---

data mining och för den som är expert på sitt speciella område urskiljs förmodligen långt fler viktiga informationskällor i en databas än vad de flesta användare ser.

Är då sådan komplex, specialiserad och kanske företagsledningsfilosofisk information viktigare eller mer relevant än summerande rapporter eller mer jordnära jämförelser? Det är en omöjlig diskussion att föra, enligt mitt sätt att se. Möjligen skulle man kunna plocka fram en lång rad förslag på rapporter och säga att de *borde* vara viktiga och relevanta. Men var det inte just det man gjorde i beslutstödens barndom, när beslutstöden inte blev använda? De som var tänkta att använda beslutstöden ställde sig inte "rätt" frågor - eller snarare, beslutstöden besvarade inte "rätt" frågor.

Om innehållet i ett beslutstöd ska kunna betraktas som viktig och relevant information är, anser jag, helt beroende av vilka behov användarna har.

Naturligtvis är man på hal is om man påstår att något är oviktigt bara för att ingen bryr sig om att veta det. Att vårda vår miljö var viktigt för oss människor långt innan tanken dök upp hos någon. I systemutvecklarens roll ligger bland annat att visa användaren på möjligheter som finns (Andersen, 1994), men det är användarens uppfattning om vad som är viktigt som måste gälla.

Man skulle kunna ställa frågan om hur det skulle gå till att aktualisera frågeställningar som ur exempelvis företagsekonomisk synpunkt *borde* vara viktiga - men det är snarare en fråga om vidareutbildning av personalen än en systemutvecklingsfråga. Låt oss bara åter slå fast att om ett beslutstöd skall vara användbart skall det svara på frågor som är av betydelse för användaren (Finlay, 1994).

## 7.2. Beslutstödet utseende

Informanterna ville inte utesluta någon form av presentation när de blev tillfrågade. De ville ha såväl diagram som tabeller, gärna möjlighet till utskrift och dessutom möjlighet att föra över data till Excel. Detta för att kunna använda presentationsmaterialet i flera olika sammanhang. De uppfattade diagram som ett bra verktyg för att ge snabba översikter framför allt till sina chefer – de ville ha tabeller och/eller siffror till Excel för att själva kunna gå djupare in i materialet. Detta är i överensstämmelse med tesen att få intryck ur ett stort datamaterial ges lättast via en graf (Dickson et al, 1986).

Vi har tillmötesgått deras önskemål på det viset att Excel-diagrammen har utrustats med datatabeller. Utskrift på papper och/eller till Excel är inte aktuellt i dagsläget. Detta är, som så ofta, en ekonomisk fråga. Naturligtvis kan man som alltid högerklicka på ett diagram på intranätssidan, kopiera det med sig för inklistring i exempelvis Word och på det viset få en utskrift.

Önskemålen om det enkla klickandets princip diskuteras mer nedan – dock kan här nämnas att det egentligen är en underlig omständighet att man ens behöver tala om att beslutstödet skall vara integrerat i det befintliga systemet. Även för systemutvecklaren är det enklare att skapa en modul i det användargränssnitt som redan finns.

Som den uppmärksamme läsaren kanske lagt märke till försvann begreppet beslutstöd relativt snart ur kommunikationen med användarna. I dess ställe kom begreppet rapporter.

Detta är måhända bara semantik, men har samtidigt betydelse för användbarheten av beslutstödet. Vi har som informatiker vårt eget språk - liksom alla ämnesspecialister. Men liksom läkaren inte bör ge diagnosen till patienten i form av medicinska termer bör inte heller informatikern dundra fram över användaren med nya och underliga termer. Jag kritiserar inte terminologin som sådan, inom yrket behövs den naturligtvis. Inte heller menar jag att på något vis nedvärdera användarnas förmåga att förstå exempelvis ordet beslutstöd.

För användaren, som många gånger står inför något nytt och ovant, underlättas ändå processen om det nya inte i onödan beskrivs med nya främmande termer. Det är en människas referensram, dvs exempelvis hennes kunskaper och erfarenheter, som bestämmer vad hon kan ta till sig (Andersen, 1994). Om till och med själva benämningen på ett nytt datoriserat hjälpmedel är främmande har man redan där satt ett hinder för acceptans.

### 7.3. Beslutstödet resultat - information

Den dolda information som finns i en stor databas är i sig endast en mängd data utan direkt synliga samband. Förhoppningsvis är mängden data inte utan betydelse och samband - den har sin ursprungliga betydelse i sitt ursprungliga sammanhang.

Att finna dold information i en databas handlar mycket om att vrida och vända på perspektivet; kanske summera lodrätt genom en datatabell som annars summeras vågrätt eller göra jämförelser, hitta likheter eller olikheter, osv.

Ett sådant arbete kan i bästa fall resultera i viktig och relevant information. Kunskap uppstår hos individen när mottagen information medvetet eller omedvetet bearbetas och jämkas in i hans begreppsvärld (Andersen, 1994).

Det kan diskuteras om man skall skilja på "kunskap" och "erfarenhet", eller kanske på den dokumenterade vetenskapen och den uppbyggda erfarenheten. Min tanke är att man borde kunna stödja erfarenheten med tabeller och grafer, och dra nytta av erfarenheten vid skapandet av tabeller och grafer.

Det beslutstöd vi talar om här, ett enklare sammanställande eller en guidning i SiRis samlade data, kräver i grund och botten erfarenhet, åtminstone i viss utsträckning. "En knapp att trycka på" har begärts fram - men fortfarande måste man ändå veta vilken anläggning man vill titta på, vilken period som kan tänkas extra besvärlig osv.

Kunskap är kopplad till en bestämd människa och beroende av den människans referensram (Andersen, 1994). Det betyder att kunskapen vi uppnår med beslutstödet som hjälp inte blir exakt densamma när olika personer använder beslutstödet. Ändå uppnår vi förhoppningsvis en bättre kunskapsnivå med hjälp av ett beslutstöd, eftersom informationen annars finns isolerad hos enskilda personer - oavsett om det är i form av erfarenhet eller egengjorda Excel-sammanställningar. I och med beslutstödet finns samma information att tillgå för alla, och det är en mer konkret och förmodligen mer rättvisande information, eftersom datorn inte glömmer eller räknar fel.

Det kan påpekas att med en mer avancerad data mining av databasen skulle den här typen av erfarenhet minska något i betydelse för beslutstödet, eftersom ett data mining verktyg självt finner mönstren och framför allt abnorma förändringar i mönstren.

#### 7.4. Beslutstödet underlag - databasfrågor

Att på detta vis leta information utifrån specifikt ställda frågor visade sig fungera utmärkt för min del, med hjälp av redan nämnda verktyg. Jag upplevde det inte som svårt och behövde inte heller lägga orimlig tid för att få svar på de frågor jag försökte ställa till databasen. Risker är givetvis: man kanske missar viktiga mönster i datan. Vill man gå från andra hållet och först finna mönster och avvikelser från mönster och utifrån det man finner formulera frågor handlar det i stället om att använda något verktyg för data mining.

Fyra av rapporterna kan bli missvisande på grund av luckor i data. Om flest antal felanmälningar kan naturligtvis sägas att småfel kanske aldrig anmäls utan bara fixas. Kostnader som inte kommer med kan vara externa fakturor som inte blir inmatade. Arbetstid kan saknas för man glömmer att mata in den. Vad gäller arbetstyperna är problemet att planerbarheten av slentrian kan anges till 0Def, dvs odefinierad. Om vi får en stor mängd arbeten av typ odefinierad får den här rapporten minskat värde ur en aspekt – men man kan också se det som en varningsklocka att systemet inte används som det borde. Ingen av dessa luckor i data är sådana att man kan kontrollera dem eller åtgärda dem inom SiRi. Att filtrera bort 0def lönar sig inte. Vad man skulle kunna göra är en åtgärd av statistiskt slag. Om man vet att arbetstypen akut förekommer på 15 % av de arbeten som har specifik arbetstyp angiven skulle man kunna tänka sig att hänföra 15 % av de odefinierade till arbetstypen akut. Bäst vore dock om typen odefinierad användes sparsamt och endast i de fall man verkligen inte kan avgöra typen.

Det är inte rimligt att tänka sig att den vanlige användaren på detta vis ska söka svar i databasen på sina frågor. Det skulle inte vara omöjligt, naturligtvis. Med viss kunskap om databaser och sql-frågor i botten kan man komma långt, om man använder sig av Sql-serverns verktyg. Det stora problemet skulle förmodligen i sådant fall vara att databasen SiRi idag innehåller drygt 300 tabeller. Visserligen kan Sql-serverns diagrambyggare visa vilka relationer tabellerna har till varandra, men om man inte har någon kännedom om SiRis databas från början kan det vara svårt att veta i vilken ände man skall börja.

#### 7.5. Faktorer som påverkar användarnas inställning till beslutstöd

##### Datorvana

Min erfarenhet är att i många fall av systemutveckling kommer själva datorn med sina applikationer och sin teknik att stå i centrum. Speciellt gäller detta då användarna inte är så vana vid datorer. Detta är ett stort hinder i arbetet, där fokus inte alls borde ligga på burken och dess knappar, utan på att försöka finna vilka problem som behöver lösas, vilka mål man har eller hur arbetet/rutinerna ska kunna förenklas/förbättras.

Samma fenomen återkom i detta arbete. I stället för att diskutera vilken information som skulle vara viktig att få, oavsett hur den togs fram eller presenterades, hade informanterna en tydlig tendens att fastna i resonemang kring sin egen ovana vid datorer i allmänhet och SiRi i synnerhet. De ansåg att de skulle kunnat svara bättre "längre fram" när de använt SiRi en längre tid. Detta är, tror jag, uttryck för den något överdrivna respekt för datorer som den ovane gärna har. Att flytta datorn från piedestalen och in i de vardagliga sysselsättningarna borde vara ett viktigt mål för alla systemutvecklare. Att förvandla datorn från något som är besvärligt och i sig självt en sorts självtillräcklighet till det hjälpmedel och verktyg underställt såväl användare som rutiner är svårt, men mycket viktigt. Så länge datorn tillåts vara en

krävande och främmande extra arbetsuppgift kommer varje nytt systemutvecklingsprojekt ha att kämpa med den ovanes motstånd.

Svårigheten att uttala sig om beslutstödet har i viss mån berott på detta missförstånd. Trots upprepade förklaringar från min sida att det inte var åsikter ur IT-synpunkt jag sökte, utan att jag ville ha ett resonemang kring önskad information kom man hela tiden att utgå från SiRi och därmed den begreppsvärld man inte tyckte sig behärska.

### Begreppet beslutstöd

Beslutstödet upplevs inte som efterlängtad av personal på operationell nivå, möjligen i många fall beroende på att man inte riktigt till att börja med ser möjligheterna. I intervjuerna upptäckte jag ganska snart en sorts motstånd mot begreppet som sådant, och övergick till att tala om rapporter och uppföljningar.

Informanterna vet exakt vad de vill ha och vad de inte vill ha när jag kommer till den delen av intervjun när jag börjar ge förslag. En del får inspiration av mina idéer och utvecklar dem eller förändrar dem något. Andra säger bara att det är intressant, respektive att det inte är intressant – men de vet precis vad de vill.

Ett hinder för hela konceptet är dock själva IT-miljön.

### IT-miljön

Naturligtvis är det alltid olyckligt om systemen är många och användarna har att irra runt från det ena till det andra. Värst blir det när användarna, som delvis i det här fallet, är ovana datoranvändare och kanske rentav motvilligt ”datoriserade”. I organisationen finns funderingar om en stegvis sammansmältning av systemen, i den mån det är möjligt (ekonomisystemet är externt administrerat). En gemensam server, ett gränssnitt mot användarna, oavsett varifrån informationen kommer.

Vad gäller maskinvara är det onekligen ett problem. Om vi ser endast till SiRi så är det ett krävande system. Lägsta möjliga kapacitet på datorutrustning för att köra SiRi har ännu inte uppnåtts överallt inom organisationen, vilket naturligtvis väcker viss irritation hos användarna.

## 7.6. Har användarna tydliga frågor?

Det viktigaste att påpeka, med risk att bli tjatig, är vikten av att användarnas frågor uppmärksammas vid utvecklandet av ett beslutstöd. Som nämns redan i avsnitt 1.2 påpekar Finlay (1994) hur viktigt det är att ett beslutstöd inte kräver för mycket extra inmatad information, att det är enkelt att använda och svarar på användarens frågor. Andersen (1994) talar om att svaren måste ges med hänsyn till användarens referensramar och kunskaper. Dessa båda saker tycker jag mig ha kunnat se ännu tydligare genom det här arbetet. Det fanns en tveksamhet till det som man inte kände till, ett motstånd mot det som kunde verka avancerat. Hos informanter på en något högre position i hierarkin finns en vana vid diagram och tabeller och också en vana vid att begära fram exakt den information man önskar sig. På operationell nivå är vanan vid diagram och tabeller inte alltid lika stor och ej heller vanan att överblicka vad man behöver / önskar sig för underlag. Det betyder inte att man sväljer vad som helst i form av fina presentationer – tvärtom. Om ett beslutstöd skall bli användbart och verkligen användas är det

---

kolossalt viktigt att användarna inte bara känner att de kommer få svar på viktiga frågor när de använder beslutstödet utan också att det känns naturligt att använda det.

### Användarmedverkan i utvecklingsprocessen

Grundtanken i denna uppsats var att prova om användarna har tydlig uppfattning om vilket beslutstöd de behöver. Utgångspunkten var kortfattat att intervjua dem, framlägga förslag och därifrån förhoppningsvis nå ett resultat i form av en användbar beslutsmodell på intranätet.

Detta kan låta enkelt och rättfram, men det finns en del problem i denna utvecklingsprocess.

Ett problem är tidsaspekten. Vi måste vara realistiska här. I mitt fall har det varit störande men inte katastrofalt att tiden bara rusade iväg utan att användarna kom till skott med några beslut kring rapportförslagen. Men detta är en akademisk uppsats, om än kring en verklig situation. I ett verkligt, fullt kommersiellt projekt med heltidsanställda systemutvecklare är det ett betydligt större problem. Ofta kan man använda tiden i väntan på respons till arbete med andra delar av systemet, men även dessa andra delar sker förhoppningsvis med användarmedverkan.

Användarmedverkan *tar tid*. En viktig slutsats blir därför att det vore att föredra att en process liknande den jag medverkat i den här terminen ska dras igång i ett mycket tidigare skede av systemutvecklingen.

"Delaktighet föder engagemang och acceptans!" utropar Goldkuhl och Röstlinger (1988) entusiastiskt. Det är inte alldeles enkelt och självklart. Jag är en stark förespråkare för en omfattande användarmedverkan i all systemutveckling. De egenskaper hos ett system som intresserar användaren, de yttre, skall också bestämmas av användaren, även om det är systemutvecklarens uppgift att förse användaren med material som gör det möjligt för henne att ta ställning (Andersen igen, 1994).

Min erfarenhet från detta arbete är att det är en bra mycket mer komplicerad, känslig och svårhanterlig process än vad Andersen eller Goldkuhl/Röstlinger ger sken av. Delaktighet föder inte alls automatiskt engagemang och acceptans. Ett hinder kan vara att användaren känner att han blir tillfrågad om och är delaktig i något hans begreppsvärld inte riktigt rymmer.

Ett annat problem kan vara om delaktigheten är "beordrad". Risken är stor att man då är involverad men inte delaktig. Ur sådan medverkan föds inte engagemang.

Vi har ett problem av allvarligt slag. Alla modeller för systemutveckling innehåller någon sorts tankar kring arbetsfördelningen mellan systemutvecklare och användare (Andersen, 1994). Under min studietid och i den litteratur jag läst har jag hela tiden fått intrycket att man försöker inpränta i oss systemutvecklare vikten av att släppa in användarna i arbetet. Ingenstans står det något om att *släpa* in dem. Man förutsätter att de är ivriga att få vara med och endast hålls borta av självgoda IT-nördar. Det är därför en smula snopet att upptäcka att användarna kanske inte alls vill vara med.

Problemet kvarstår - systemutvecklaren är expert på utvecklingsarbete medan användaren är expert på det arbetsområde systemet skall stötta. En seriös systemutvecklare kan inte genomföra sitt arbete utan användarens expertmedverkan (Andersen, 1994).

Hur man ska lösa det problemet och hur man gör för att engagera användaren och göra honom verkligt delaktig är frågor som kräver vissa kunskaper i beteendevetenskap. Utan sådana kunskaper får man ändå försöka att med stort tålamod och stor lyhördhet göra sitt yttersta för att få samarbetet att bli ett samarbete.

### Informanternas förslag

Vi har tyvärr bitvis en stark påverkan från min sida. Kanske hade en skickligare intervjuare lyckats locka fram idéerna och förslagen utan att själv komma med dem, men jag kunde det inte alltid. En del informanter kom själva med sina förslag, men övriga behövde mer eller mindre tydliga förslag från min sida att ta ställning till.

Tio-i-topp-listorna av olika slag är det underlag som uppfattas som någon sorts stöd för beslut, medan periodrapporterna snarare är någon sorts uppvisande dels av om det går bättre nu (färre felanmälningar exempelvis) och dels rent generellt hur mycket man haft igång. Periodrapporterna lämnas därhän för ögonblicket. De hamnar lite utanför beslutstödsknappen på intranätet.

Kan man verkligen säga att önskemålen var deras, inte mina? Jag skulle vilja uttrycka det som att önskemålen/förslagen i slutändan blev en sorts sammanbakning av informanternas större kunskap om sin arbetssituation och min större kunskap om databasen. Trots att det ibland var jag som uttryckte förslagen var responsen på dessa förslag av ett så klargörande och medvetet slag att jag inte uppfattade det som att min inverkan var särskilt övervägande. Om mina förslag var av intresse togs de dessutom genomgående upp med någon ändring eller något tillägg. Jag kunde föreslå underhållskostnad per nypris och höra ”nej, det är inte intressant, men underhållskostnad per drifttid är mycket intressant”.

### Hur skall man presentera förslagen?

Att presentera förslagen i dynamiska demoversioner i stället för statiskt pappersformat skulle vara optimalt ur användarmedverkan-synpunkt.

Det är, har jag förstått, av olika anledningar svårt att föreställa sig en rapport eller ett programavsnitt bara genom att få det beskrivet för sig. Att beskrivningen åtföljs av illustrationer har åtminstone inte i det här fallet hjälpt mycket. Särskilt en ovan datoranvändare har svårt att se möjligheter och brister i ett så pass förenklat förslag.

Men detta är återigen ett dilemma. Att programmera tar tid och kostar därmed pengar. Att som i mitt fall åstadkomma alla 8 förslagen i någorlunda fungerande rapporter på intranätet skulle vara ekonomiskt orimligt för alla andra än en uppsatsskrivande student. Ja, det skulle faktiskt vara orimligt för en sådan person också – av hänsyn till den tidsbegränsning man har.

Å andra sidan skulle man då kunna dra ned på antalet rapportförslag, men i gengäld lägga mer arbete på varje förslag.

Detta skulle i mitt fall kanske ha betytt att jag presenterade fem förslag i dynamisk intranätversion. Dessa fem förslag skulle då ha varit baserade endast på intervjuerna. Med en sådan metod skulle man förlora en bit av användarnas medverkan. De fem förslag jag i sådant fall hade presenterat skulle ha blivit en utgångspunkt för vidare diskussioner, men det är i en sådan situation stor risk att man låser sig vid just de fem rapporterna. Kanske lägger man till parametrar eller ändrar utseendet, men man kommer att utgå från grundidéen i varje förslag.



### Mina slutsatser

Användarmedverkan tar lång tid. Den är inte självklart enkel i alla lägen eftersom användarna kan tycka det är svårt att ge respons. Bristande erfarenhet, bland annat av de tekniska möjligheterna och därmed svårigheter att se olika lösningar framför sig kan vara orsaker till detta. Bristande engagemang, kanske på grund av att man inte upplever sig verkligt delaktig, kan vara en annan orsak.

Att helt och hållet bygga DSS utifrån användarnas önskemål kan, anser jag, leda till att man visserligen får ett användbart, enkelt och effektivt beslutstöd, men det kan tyvärr också innebära en fattigdom i resultatet. Kanske missar man spännande lösningar som skulle leda till ett än mer effektivt DSS. Användare med smalare arbetsområde eller mer praktiska arbetsuppgifter kan av förklarliga skäl ha svårare att se vad man skulle kunna ha nytta av.

Systemutvecklaren måste förmodligen vara mer påstridig i sin jakt på respons och samtidigt lyhörd och tålmodig.

Att responsen är så svår att få kan möjligen ge en antydning om en viss tveksamhet hos användarna när det gäller beslutstödet värde. Man tycker att det skulle "vara bra att ha" – men det är inte speciellt viktigt.

### 7.7. Svårigheter i intervjusituationen

Fjorton intervjuer (som det från början var tänkt) kan tyckas vara onödigt många. Å andra sidan var informanterna representanter för tre olika avdelningar, som delvis arbetar på olika sätt.

Att den intervjuade personalen kom från olika nivåer i organisationen var en stor fördel, eftersom det gav mig ganska skilda perspektiv på datorisering i allmänhet och beslutstöd och rapporter i synnerhet.

Frågeställningarna rent allmänt omnämndes i mitt brev. Tanken var att intervjuerna inte skulle hållas som en frågestund från min sida, utan som samtal där min roll skulle vara den lyssnande.

Det visade sig vara svårt. Till att börja med var jag förmodligen lite för lyssnande, vilket resulterade i att jag nu vet allt om informanternas barnbarn och arbetskamrater, besvikelser och funderingar kring arbetet i allmänhet och SiRi i synnerhet. Jag har fått ett stort antal önskemål och frågor som snarare handlar om systemet SiRi än om vad jag var där för – och antecknade naturligtvis dem också för vidare befordran. Uppenbarligen uppfattades jag som tämligen ofarlig – till mig kunde man säga sitt hjärtas mening om både arbetet och SiRi, eftersom jag kom utifrån och inte på något vis alls varit personligen närvarande i något arbete. Bikan AB tidigare har gjort i Borås.

Den delen av samtalen var både trevliga och ibland lite mindre trevliga naturligtvis, och mina första försök att styra samtalen rätt var nog lite taffliga. Med varje intervju tyckte jag själv att jag klarade det bättre, vilket var en trevlig känsla.

Andra svårigheter var tuffare att få någon ordning på. Det fanns ett motstånd mot att yttra sig, kände jag, för informanterna kunde inte riktigt frigöra sig från SiRi. Det betydde att de ägnade en hel del tid åt att generat förklara att de inte hunnit använda SiRi så länge (vilket är förståeligt eftersom SiRi inte varit igång särskilt länge!) och att de inte heller hunnit gå utbildningen och därför inte "kan" så mycket. Mina upprepade försäkringar att det inte var viktigt just nu, för min del, utan att jag bara ville veta vad de behöver, vad de vill ha, hjälpte inte stort.

Många av dem använder idag Excel till att ställa samman rapporter av olika slag. Dessa rapporter innefattar åtminstone i vissa fall data som kommer från SiRi. Men det var svårt för mig att få reda på vilka dessa rapporter var, eftersom informanterna upplevde sina rapporter som något självklart som redan finns, och som inte hade med mina frågor att göra.

## 7.8. Komponenter

Ett resultat som inte lätt låter sig visas upp eller diskuteras är naturligtvis ActiveX-komponenterna. Rent allmänt kan man väl citera min mamma som brukar säga "övning ger färdighet". Det finns bara ett sätt att lära sig programmera och det är att göra det. Sen är naturligtvis goda böcker i ämnet till stor nytta.

Ett för mig tidigare okänt bekymmer var att komponenterna placerades på intranät-servern. Tydligt berodde det på att web-sidorna behövde dem där för att hitta dem. Kanske finns det någon inställning för att använda dem från annan server, men det har jag inte hunnit utröna. Nu uppstod i stället frågan om hur mina komponenter skulle hitta SiRis databas, som fysiskt finns på en annan server. Detta löstes genom en liten Access-databas med samma namn som komponenten. Denna lilla databas innehåller en tabell med ett fält och en post, i vilken man skriver in den rätta adressen till SiRis server och databas. Komponent, som vet vad den själv heter, kan hämta fram innehållet i den lilla databasen och under förutsättning att man skrivit adressen rätt, vilket i sig kan ha sina sidor, så fungerar det utmärkt.

Mina begränsade kunskaper i ASP och HTML gjorde att programmeringen tog längre tid än den skulle ha gjort för en mer van programmerare.

## 7.9. Varthän...

Arbetet med de här fem framtagna rapporterna kommer att gå vidare. De enkla men fungerande intranät-versionerna kommer att presenteras på ett arbetsgruppmöte för kommentarer.

I ett vidare perspektiv är den naturliga fortsättningen på detta arbete ett försök att lägga en större kraft bakom ett lika enkelt beslutstöd. Det vill säga, varför inte utveckla beslutstöd för operationell miljö – fortfarande "bara en knapp" men med avancerade statistiska beräkningar bakom.

## 7.10. På annat sätt

... fanns det en del jag kunde gjort. Min oerfarenhet som intervjuare var ett hinder. Jag borde ha styrt upp intervjuerna lite mer. Det hade förmodligen kunnat ge ett mer omfattande resultat med tanke på att varje intervju tog en timme.

För ett testprojekt kring beslutstöd kunde man möjligen ha nöjt sig med den inte fullkomliga men redan på marknaden existerande grafikkomponenten i stället för att programmera en egen.

Jag borde också, som nämnts, ha skickat ut rapportförslagen några dagar innan presentationen.

Ett annat, möjligen bättre, sätt kunde ha varit att kalla till ett nytt möte någon vecka efter utskicket av rapportförslaget. Allra bäst hade varit om man kunnat samla ett större möte med fler användare från de olika avdelningarna. Ett sådant möte, om man hade fått tillåtelse att sammankalla det, hade inte behövt störa det dagliga arbetet med mer än någon timme, men kunde eventuellt ha gett mer direkt respons.

## 8. Slutsats

Vi har visat att det finns dold information i databasen. I de flesta fallen var den sökta informationen av relativt okomplicerat slag, exempelvis rättframt summerande rapporter med få parametrar.

Användarnas frågor var möjligen initialt något mer omedvetna än förväntat. Det fanns inte heller ett så stort engagemang hos användarna som man kunde ha förväntat sig. Detta kan bero på en rad olika saker, men en viktig orsak i det här fallet har varit att datorernas värld är en aning främmande för många av dem.

Slutsatsen är att man kan ta fram ett enkelt men användbart beslutstöd inom en redan befintlig applikation. Det viktiga är att vara mycket noggrann med återknytningen till användarna. Användarmedverkan tar tid och är betydligt mer komplicerat än att "gå ut och fråga", men den är nödvändig i ett seriöst utvecklingsarbete.

## 9. Förkortningar

|       |                                  |
|-------|----------------------------------|
| DSS   | Decision Support Systems         |
| EIS   | Executive Information Systems    |
| ES    | Expert Systems                   |
| KDD   | Knowledge Discovery in Databases |
| MINTS | Management Intelligence Systems  |
| MIS   | Management Information Systems   |
| MSS   | Management Support Systems       |
| TPS   | Transactions Processing Systems  |

## 10. Referenser

### Böcker/artiklar

Abrahamsson, Bengt & Andersen, Jon Aarum (1996). *Organisation – att beskriva och förstå organisationer*. Malmö. Liber-Hermods

Andersen, Erling S (1994). *Systemutveckling – principer, metoder och tekniker*. Lund: Studentlitteratur.

Dahlbom, B. & Mathiassen, L. (1993). *Computers in Context. The Philosophy and Practice of Systems Design*, Cambridge and Oxford: NCC Blackwell.

Dickson, Gary W, DeSanctis, Gerardine & McBride, D. J (1986). "Understanding the Effectiveness of Computer Graphics for Decision Support: A Cumulative Experimental Approach". *Communications of the ACM*, January 1986, Volume 29, no 1

Easterby-Smith, M & Thorpe, R & Lowe, A (1991). *Management Research*, London: Sage Publications Ltd

Eriksson, Lars Torsten & Wiedersheim-Paul, Finn (1997). *Att utreda, forska och rapportera*. Malmö: Liber Ekonomi

Fayyad, Usama, Piatetsky-Shapiro, Gregory & Smyth, Padhraic (1996). "The KDD Process for Extracting Useful Knowledge from Volumes of Data". *Communications of the ACM*, November 1996, Vol 39, No 11

Finlay, Paul (1994). *Introducing decision support systems*. Hartnolls Ltd, GB

Goldkuhl, Göran & Röstlinger, Annie (1988). *Förändringsanalys*. Lund, Studentlitteratur.

Hughes, J., King, V., Rodden, T. & Andersen, H. (1994). Moving out of the Control Room: Ethnography in System Design. *Proceedings of Computer Supported Cooperative Work (CSCW)*, pp 429 – 439

Isaacs, Scott (1997). *Inside Dynamic HTML*. Washington, Microsoft Press

Lomax, Paul (1998). *VB & VBA In a Nutshell*. Sebastopol, USA. O'Reilly & Associates Inc.

Lundgren, E. (1992), "Att uppenbara det dolda", *Kvinnovetenskaplig Tidsskrift*, nr 2, s 3-14  
Kompendium Institutionen för Informatik i Göteborg.

Mouwitz, Lars (1997). *Filosofi – en lärobok*. Malmö. Gleerups Förlag.

Roman, Steven (1999). *Writing Excel Macros*. Sebastopol, USA. O'Reilly & Associates Inc.

Turban, Efraim & Aronson, Jay E (1998). *Decision Support Systems and Intelligent Systems*, 5<sup>th</sup> ed. New Jersey, USA. Prentice-Hall Inc.

#### **Artiklar/länkar på Internet**

Brand, Estelle & Gerritsen, Rob (1998). *Data Mining and Knowledge Discovery*. Miller Freeman, Inc. <http://www.dbmsmag.com/9807m01.html> (1999-12-21)

#### **Magisteruppsatser 20 p**

Arvidsson, D & Johansson, T (1999), "The Hummingbird as a Support for Mobile Group Awareness – An Evaluation". Institutionen för Informatik, Göteborgs Universitet

**Relevant litteratur**

Banville, Claude & Landry, Maurice (1989). "Can the Field of MIS be Disciplined?" *Communications of the ACM*, January 1989, Vol 32, No 1

Glymour, Clark & Madigan, David & Pregibon, Daryl & Smyth, Padhraic (1996). "Statistical Inference and Data Mining". *Communications of the ACM*, November 1996, Vol 39, No 11

Wilson, Francis A & Wilson John N & Smith, Anne M (1993). "Computer-based systems: A Discussion of their Application to Managerial Decision-Support". *Communications of the ACM*, 1993.

Andersson, Catrin och Elfström, Annika (1998), "'Data Mining', En revolution eller ännu ett analysverktyg?", Institutionen för Informatik, Göteborgs Universitet



---

Intervjuerna hölls snarare som öppna samtal och frågorna nedan var endast tänkta som ett stöd för mig själv att föra samtalet framåt. Jag ställde alltså i många fall inte dessa frågor direkt, men försökte se till att varje samtal berört dessa ämnen.

- Vilka är dina huvudsakliga arbetsuppgifter?
- Vilka sorts beslut, små eller stora, behöver du ta i ditt arbete?
- Finns det någon information som skulle kunna tänkas underlätta dina beslut?
- Finns det någon speciell information du skulle vilja kunna visa din närmaste chef i en rapport?
- Hur skulle du vilja att den informationen såg ut (papper/bildskärm, detaljerat i siffror/grafier....?)

Mot slutet av samtalen hade jag en del egna förslag som jag framförde. Det var:

- Arbetsbelastningsrapport för olika yrkesgrupper (elektriker, mekaniker osv)
- Tidsåtgångsstatistik av olika slag, ex hur lång tid det tar från felanmälan till avrapportering, eller hur ofta kommer det felanmälningar på speciella komponenter/anläggningsdelar
- Hade fel/skada kunnat undvikas om komponenten varit med i rondering?
- Frekvens felanmälan på speciellt fabrikat
- Blev ett fel snabbare åtgärdat om prioriteten satts högre än om den satts lågt?
- Hur mycket mindre kostnader per år skulle det innebära om viss anläggningsdel skrotades?