

Datavetenskap

Johan Berntsson och Erik Johansson

**Distributionssystem för
kvalitetssäkringsinformation**

Examensarbete, C-nivå

2003:21

Distributionssystem för kvalitetssäkringsinformation

Johan Berntsson och Erik Johansson

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är vårt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Johan Berntsson och Erik Johansson

Godkänd, 2003-06-04

Handledare: Per Strömgren

Examinator: Stefan Lindskog

Sammanfattning

Rapporten beskriver dels framtagandet av ett informationspaket för kvalitetssäkring, dels utvecklingen av ett system för att behandla och distribuera informationen åt Volvo Cars Customer Service (VCCS). Anledningen till projektet var att VCCS inte hade något standardiserat informationspaket och att informationen behandlades på ett omständligt och ålderdomligt sätt. Projektet har innefattat allt från förstudie och analys av krav till konstruktion och implementation.

Slutresultatet blev ett framtaget informationspaket som uppfyller de krav som var ställda samt ett system som behandlar och distribuerar denna information.

Distribution system for quality assurance information

Abstract

This document describes partly the research involved in producing an information package for quality assurance, partly the developing process of a system that process and distributes information for Volvo Cars Customer Service. The reason for this project was due to VCCS's lack of a standardised information package and that the processing of the information was old-fashioned and long-winded. The project involved everything from research and requirement analysis to construction and implementation.

The project resulted in an information package that fulfils the given requirements and a system that process and distributes this information.

Innehållsförteckning

1	Problembeskrivning.....	4
1.1	Bakgrund	4
1.2	Problem.....	4
1.3	Mål.....	4
2	Metod	4
2.1	Förstudie	4
2.1.1	Kravinsamling	
2.1.2	Analys och utvärdering av krav	
2.1.3	Avstämning	
2.1.4	Kravspecifikation	
2.2	Konstruktion och implementation	4
2.2.1	Automatisering	
2.2.2	Inläsning av information	
2.2.3	Grafiskt användargränssnitt	
2.2.4	Databas	
2.2.5	Grafgenerering	
2.2.6	Distribution	
3	Tekniker.....	4
3.1	Objektmodeller	4
3.1.1	Data Access Objects	
3.1.2	Excels och Outlooks objektmodell	
3.2	Relationsdatabas	4
3.3	SQL.....	4
3.4	Visual Basic.....	4
4	Erfarenheter	4
5	Slutsats	4
	Referenser	4
A	DAO.....	A-4
A.1	Skapa och ändra databasdesign	A-4
A.2	Hämta, ändra och lägga till data i en databas	A-4
A.3	Hämta, länka eller ändra data i andra typer av databaser	A-4

B	Slutgiltig parameterlista	B-4
C	Kravspecifikation	C-4

Figurförteckning

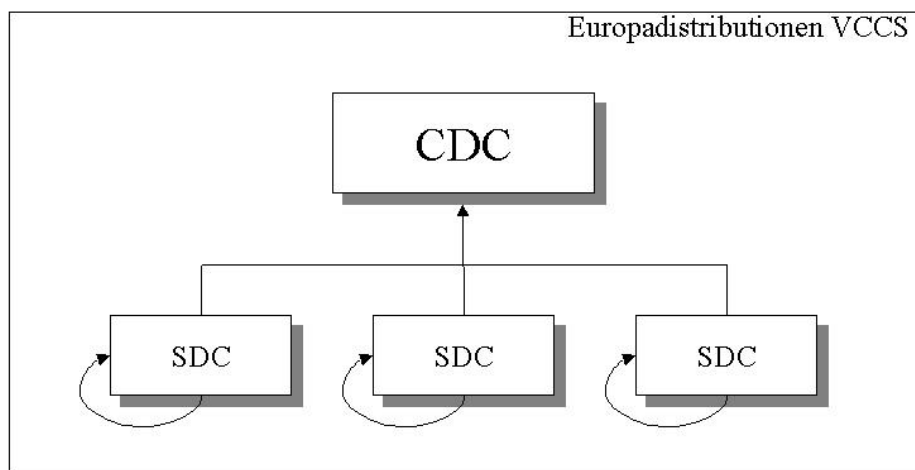
Figur 1.1: Redovisningsstruktur för reservdelsdistributionen i Europa.....	4
Figur 1.2: Hur uppföljningarna gjordes förut	4
Figur 2.1: Europe Operational Organization - Avgränsning.....	4
Figur 2.2: Systemöversikt	4
Figur 2.3: Huvudmeny	4
Figur 2.4: Relationsdiagram av konfigurationstabellerna.....	4
Figur 3.1: Excels objektträd.....	4
Figur 3.2: Outlooks objektträd.....	4
Figur A.1: Microsoft Jetdatabas objektträd.....	A-4

Tabellförteckning

Tabell 2.1: Parameterlista med minimikrav	4
Tabell B.1: Slutgiltig parameterlista	B-4

1 Problembeskrivning

1.1 Bakgrund



Figur 1.1: Redovisningsstruktur för reservdelsdistributionen i Europa

Volvo Cars Customer Service (VCCS) är den del av Volvo Cars som tillhandahåller och distribuerar reservdelar för alla Volvos bilar producerade de senaste 15 åren.

Europadistributionen VCCS består av ett Central Distribution Center (CDC) som är huvudlagret och ett antal Support Distribution Centers (SDC:er) som är mindre lager utplacerade i Europa för snabbare distribution till återförsäljarna. Varje SDC ansvarar för ett antal marknader. Exempel på marknader är Tyskland och Österrike. Det finns två olika typer av SDC:er, de som drivs av Volvopersonal och de som drivs av inhyrd personal (s.k. tredjepartslogistik).

SDC:erna redovisar sin produktivitet, effektivitet och kvalitet både internt och för chefen på Europaavdelningen VCCS. Denna chef sitter på Europakontoret på CDC (Se Figur 1.1) i Göteborg. Underlaget redovisningen tas från ett stordatorsystem lokaliserat i Sverige med hjälp av egenutvecklade program. Parametrar som mäts är till exempel packade rader, packade rader per timma och antalet leveransanmärkningar över en viss tidsperiod. Vi kommer inte att gå in närmare på vad dessa parametrar har för innebörd då detta inte har någon egentlig betydelse för rapporten. Mätdata skrivs sedan ut i form av listor hos SDC:erna och personalen får sedan sammanställa informationen som grafer och diagram i

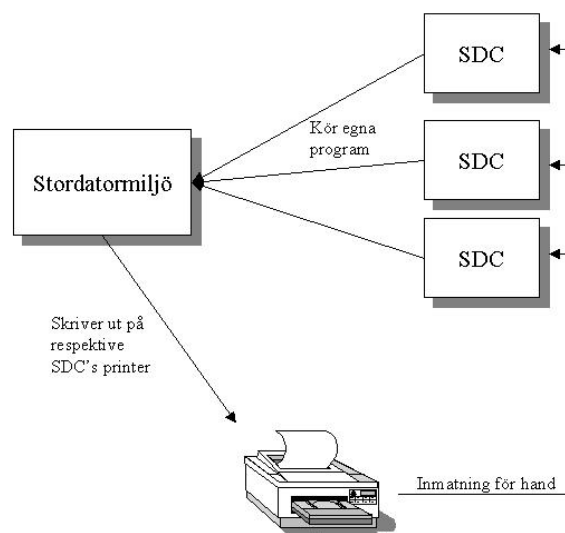
Exceldokument som sedan redovisas. Denna sammanställning sker veckovis, månadsvis och kvartalsvis.

1.2 Problem

VCCS har idag inte någon standardiserad metod för uppföljningar bland sina SDC:er, vilket gör det svårt att kontrollera att uppsatta mål uppnås. Programmen som används för att hämta ut informationen ur stordatorsystemet är ofta gamla och producerade på olika håll utan någon som helst standard vilket gör det mycket svårt att jämföra olika värden. Skapandet av Exceldokumenterna som skall användas för redovisning sker för hand vilket innebär en klar risk för felaktiga uppföljningar då parametrarna som skall matas in är många, denna process är också mycket tidskrävande (Se Figur 1.2).

Sammanfattning av de fem huvudproblemen.

- Ej enhetliga uppföljningar, vilket leder till svårigheter att tyda och jämföra uppföljningar.
- Gamla icke uppdaterade program.
- Risken för fel när inmatning sker för hand.
- Tidsåtgången för att mata in informationen för hand och för att tyda olika grafer och diagram.
- Uppföljningar sker ej inom uppsatt tid.



Figur 1.2: Hur uppföljningarna gjordes förut

1.3 Mål

Projektets mål delas upp i två delar, ett primärmål och ett sekundärmål. Anledningen till detta är att det är svårt att förutse hur stort projektet kommer att bli och vilka krav som kommer att ställas. Om bägge målen kommer att utföras bestäms efter att analys och utvärdering av krav utförts.

Primärmålet innefattar att ta fram ett gemensamt informationspaket för SDC:erna och en gemensam standard för att presentera informationen för chefen på Europaavdelningen. Informationspaketet skall innehålla ett antal bestämda parametrar, varje SDC skall använda samma informationspaket.

Sekundärmålet är att utveckla och implementera ett system som automatiskt skapar grafer utifrån informationspaketet och automatiskt distribuerar dessa till SDC:erna. Systemet skall uppfylla de krav som förstudien kommer visa.

Dessa mål skall göra det enkelt och effektivt att skapa uppföljningar och kontrollera att uppsatta mål uppnås då samma informationspaket kommer användas.

2 Metod

Projektet innehåller tre olika utvecklingssteg, förstudie, konstruktion och implementation, vilka beskrivs nedan.

2.1 Förstudie

Det huvudsakliga syftet med förstudien var att ta fram de parametrar som informationspaketet skulle innehålla. Det andra var att ta reda på om det fanns tillgänglig teknik och tid till att ta fram ett bättre sätt att leverera informationen till SDC-cheferna, till exempel färdiga diagram eller grafer via e-mail.

Det fanns många parter som ville säga sitt och dessa parter var vana vid att bestämma på egen hand så det krävdes lite arbete för att få alla att komma överens. De olika parterna såg också sin chans att få extra arbete utfört, de kom med förslag som låg utanför projektet, vilket medförde att en klar avgränsning fick göras.

2.1.1 Kravinsamling

Kravinsamlingen inleddes med att ett formellt e-mail skickades till SDC-cheferna där det förklarades vad projektet gick ut på och vad som förväntades av dem. I e-målet inkluderades

en lista med de obligatoriska minimikraven på parametrar (Se Tabell 2.1) som chefen för Europaavdelningen tagit fram och en förfrågan om ytterligare förslag till listan.

Key figures	Frequency
Packed Lines	weekly
Packed Value per Hour (all hours)	weekly
Packed Lines per hour (all hours)	weekly
Packed Value	weekly
Invoiced Lines	weekly
Invoiced Value	weekly
Leadtime	weekly
Availability	weekly
Cost Per Order Line (all hours WC & BC) at SDC	weekly
Zero Picks	weekly
Lines Binned	weekly
Value Binned	weekly
Binned Lines Per Hour	weekly
Refill Deviations	weekly
Parts Stocked	weekly
Inventory Remaining	weekly
Inventory Adjustments +/- Physical/Value	weekly
Picking Percentage Per Area (Mezzanine/Racking/Bulky)	weekly
Discrepancies 00/20/40/72	monthly

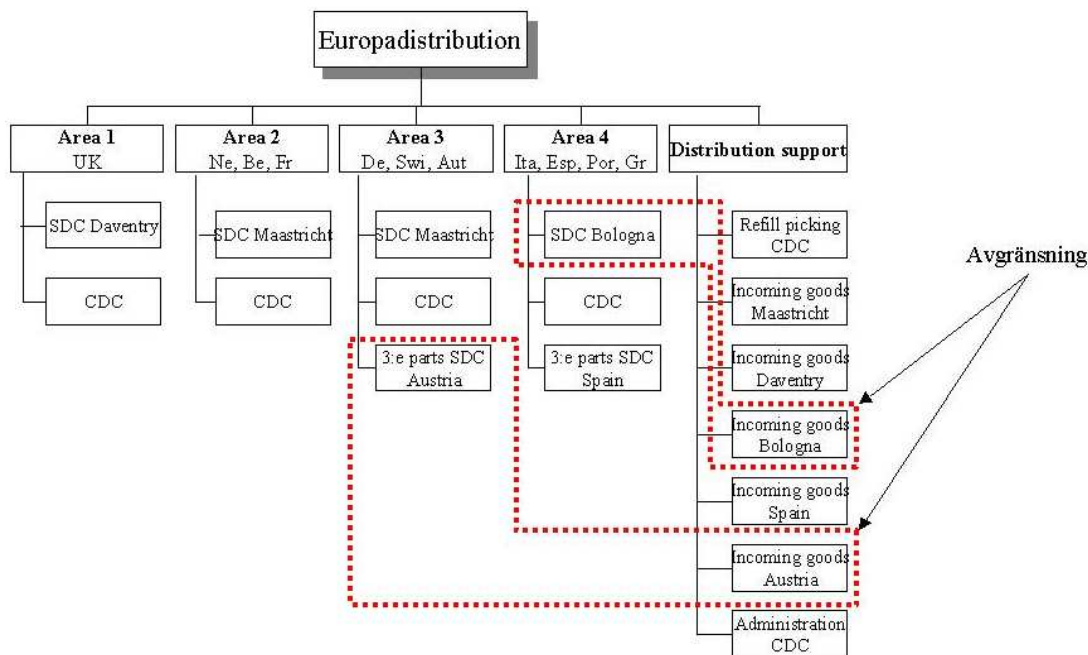
Tabell 2.1: Parameterlista med minimikrav

Som svar på e-målet kom många nya förslag till parametrar som skulle läggas till listan. Det visade sig att det existerade en betydligt djupare komplexitet i organisationen än vad som först var väntat, organisationen var uppdelad i areor.

En area består av ett SDC samt en avdelning på CDC:et i Sverige som står för leverans av delar som SDC:et inte har i lager och ibland även en tredjepartslogistiker. De olika delarna av arean ansvarar tillsammans för en samling marknader. För att få korrekta mätvärden med denna modell krävdes bl.a. att SDC:ernas mätvärden summeras med det som producerats på CDC:et för SDC:ets räkning

För att göra en korrekt utredning i denna komplexa struktur krävdes betydligt mer tid än vad projektet innefattar vilket medförde att en avgränsning fick göras till att innehålla vad SDC:erna levererar samt vad de levererar per marknad, vilket visas i nedan.

Dagens uppföljningar följer dock inte heller area-modellen, vilket medförde att avgränsningen inte var något större bakslag för projektet.



Figur 2.1: Europe Operational Organization - Avgränsning

Efter många turer fram och tillbaka framkom en slutgiltig lista med ett trettiotal parametrar som alla berörda parter var nöjda med (se Bilaga B).

2.1.2 Analys och utvärdering av krav

Efter samtal med chefen för Europaavdelningen avgjordes det vilka av önskemålen från SDC-cheferna som var möjliga att genomföra inom tidsramen för projektet, de som ej var möjliga att genomföra sorterades bort.

En del av den information som krävdes från stordatorsystemet visade sig ej finnas inlagda där, utan istället hanterades internt på varje SDC. Detta löstes genom att skapa ett fördefinierat Exceldokument som SDC-cheferna fyller i med information om antalet timmar. Detta Exceldokument skall sedan sändas till systemet via e-mail varje vecka.

VCCS hade inte någon entydig definition av parametrarna som informationspaketet nu innehöll, detta medförde att intervjuer med flera kontaktpersoner inom organisationen var nödvändiga. En lista med de insamlade definitionerna producerades som slutligen kontrollerades och godkändes av chefen för Europaavdelningen tillsammans med SDC-cheferna.

2.1.3 Avstämning

Inför avstämningen fanns tre möjliga scenarier som kunde inträffa beroende på vilken information som fanns tillgänglig, finansaspekten och tidsåtgången.

1. Skriva kravspecifikation och konstruktionsbeskrivning till Volvo IT så att de kan skapa ett nytt program i stordatormiljö, i detta scenario skulle endast rådata distribueras till SDC:erna.
2. Skriva kravspecifikation och konstruktionsbeskrivning till Volvo IT så att de kunde skapa ett nytt program i stordatormiljön och även konstruera och implementera en separat applikation i Windowsmiljö som automatiskt skulle distribuera datan i graf- och diagramform.
3. Återanvända de program som redan fanns i stordatormiljön för de olika parametrarna och samla informationen i en separat applikation i Windowsmiljö och sedan distribuera informationen i graf- och diagramform.

De ansvariga på VCCS ville helst se att scenario 2 realiserades men efter möte med Volvo IT visade det sig vara svårt för Volvo IT att utveckla lämpliga stordatorprogram inom rimlig tid p.g.a. besparingar. Volvo IT kunde dock producera vissa program för att hämta informationen från stordatormiljön men inte alla. Eftersom många småprogram redan fanns färdiga på VCCS bestämdes det för en kompromiss där VCCS och Volvo IT skulle dela på jobbet. Detta gav oss en kombination av scenario 2 och 3, vilket innebar att också sekundärmålet skulle uppnås.

2.1.4 Kravspecifikation

Kravspecifikationen utformades med hjälp av en mall som tillhandahölls av vår handledare eftersom Volvo inte hade någon sådan mall.

Sammanfattningsvis innehåller kravspecifikationen följande punkter.

- Systemet skall ta emot information som Volvo IT tillsammans med VCCS tillhandahåller via e-mail från ett stordatorsystem.
- SDC:erna skall varje vecka fylla i ett formulär
- Informationen skall omvandlas till grafer och tabeller i form av Exceldokument enligt en viss standard.

- Exceldokumenten skall distribueras via e-mail till SDC-cheferna i förutbestämda intervall.
- Ovan nämnda punkter skall sköts med automatik.
- Systemet skall klara av att stå online dygnet runt.
- Ett grafiskt användargränssnitt skall finnas för att administrera systemet.
 - Ändring av grafstandarder.
 - Uppdatering av SDC:er.
 - Uppdatering av informationspaket.

Den fullständiga kravspecifikationen finns i Bilaga C.

2.2 Konstruktion och implementation

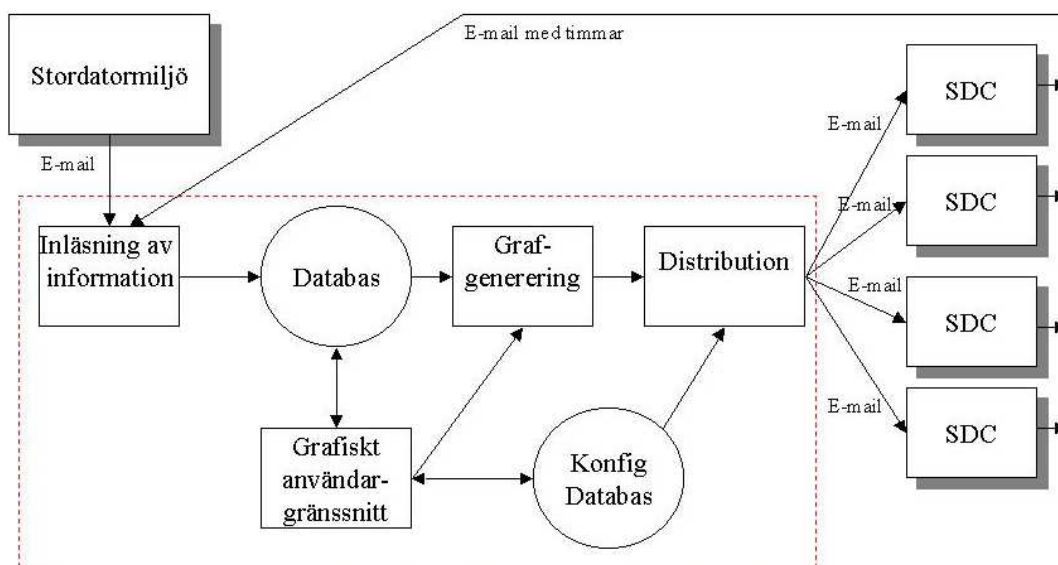
För att realisera kraven som ställdes i kravspecifikationen behövdes ett antal programdelar:

- Då graferna inte enbart skall innehålla information från det senaste intervall utan också information från tidigare tillfällen (oftast ska graferna vara årsvis), behövs en databas för att mellanlagra informationen på ett säkert sätt.
- Informationen kommer från stordatormiljön som bifogade filer i e-mail och då behövs en inläsningsdel som parsar ut informationen och sparar informationen i databasen.
- Ett grafiskt användargränssnitt där varje SDC med tillhörande parametrar och grafer specificeras. På grund av alla de inställningsmöjligheter som krävs sparas dessa inställningar också i databasen.
- En del i systemet som skapar diagram och grafer i form av Exceldokument från informationen som finns lagrad i databasen.
- En distribueringsdel som distribuerar excelokumenten via e-mail till SDC-cheferna.
- En automatiseringsdel som samordnar alla de andra delarna och utför rätt operationer vid rätt tidpunkter.

Sammanfattning av de delar som ska skapas (Se Figur 2.2):

- Automatisering.
- Inläsning av information.
- Grafiskt användargränssnitt.
- Databas.
- Grafgenerering.

- Distribution.



Figur 2.2: Systemöversikt

2.2.1 Automatisering

Automatiseringen är den styrande och samordnande delen i systemet. Automatiseringen bygger på en loop och två värden, importdatum och distribueringsdatum.

Importdatumet talar om nästa gång en parameter skall importeras från brevlådan som stordatorsystemet har levererat den senaste informationen om parametern till. Datumet sätts första gången vid konfigurationen av parametern och räknas därefter med hjälp av det senaste datumet och med vilket tidsintervall som parametern skall importeras. Om parametern till exempel skall importeras veckovis och det senaste datumet då parametern importerades var en måndag räknar systemet ut datumet för måndagen i nästa vecka och sätter importdatumet till detta.

Distribueringsdatumet fungerar på samma sätt som importdatumet fast det talar om när ett visst Exceldokument skall skapas och distribueras.

Loopen går hela tiden men utträttar inget förrän det finns ett import eller distribueringsdatum som är utgången, loopen fortsätter tills den blir stoppad genom det grafiska användargränssnittet (Se Figur 2.3).

Automatiseringsprocessen

1. Inläsning av data

Läs in parametrar från stordatorsystemet till databasen vars importdatum är utgången.

Uppdatera importdatumet.

2. Grafgenerering

Skapa grafer och tabeller i form av Exceldokument vars distribueringsdatum är utgången.

3. Distribution

Distribuera Exceldokumenterna via e-mail vars distribueringsdatum är utgången.

Uppdatera distribueringsdatumet.

2.2.2 Inläsning av information

Informationen kommer som bifogade filer via e-mail vid bestämda tidpunkter (Se Figur 2.2). Dessa e-mail kommer både ifrån stordatormiljön och SDC:erna. Den information som kommer från SDC:erna innehåller uppgifter om timmar som ej finns inlagda i stordatormiljön utan sparas internt för varje SDC. SDC-cheferna fyller i ett Excel-formulär varje vecka anpassat just för dessa timmar. Timmarna används för att räkna ut sådant som packade rader per timma m.m. (Se Tabell B.1).

Formatet på de bifogade filerna är semikolon separerade kolumner. Filerna består av ett varierande antal kolumner med värden. Alla bifogade filer, från en förutbestämd avsändare oavsett antal e-mail, sparas på disk. Kommunikationen mellan Microsoft Access och Microsoft Outlook sker genom användandet av Outlook's objektmodell.

Konfigurationen av vilka parametrar som finns i vilka filer samt var dessa ska sparas finns i databasen. Filerna öppnas i Excel som automatiskt öppnar filerna kolumnvis, där hämtas informationen och sparas i databasen.

Eftersom systemet ska stå online dygnet runt i obegränsad tid raderas både e-mailen och filerna, som är sparade på disk, efter att information sparats i databasen. Om detta ej görs innebär det en stor risk att systemet kraschar eftersom diskutrymmet är begränsat.

2.2.3 Grafiskt användargränssnitt

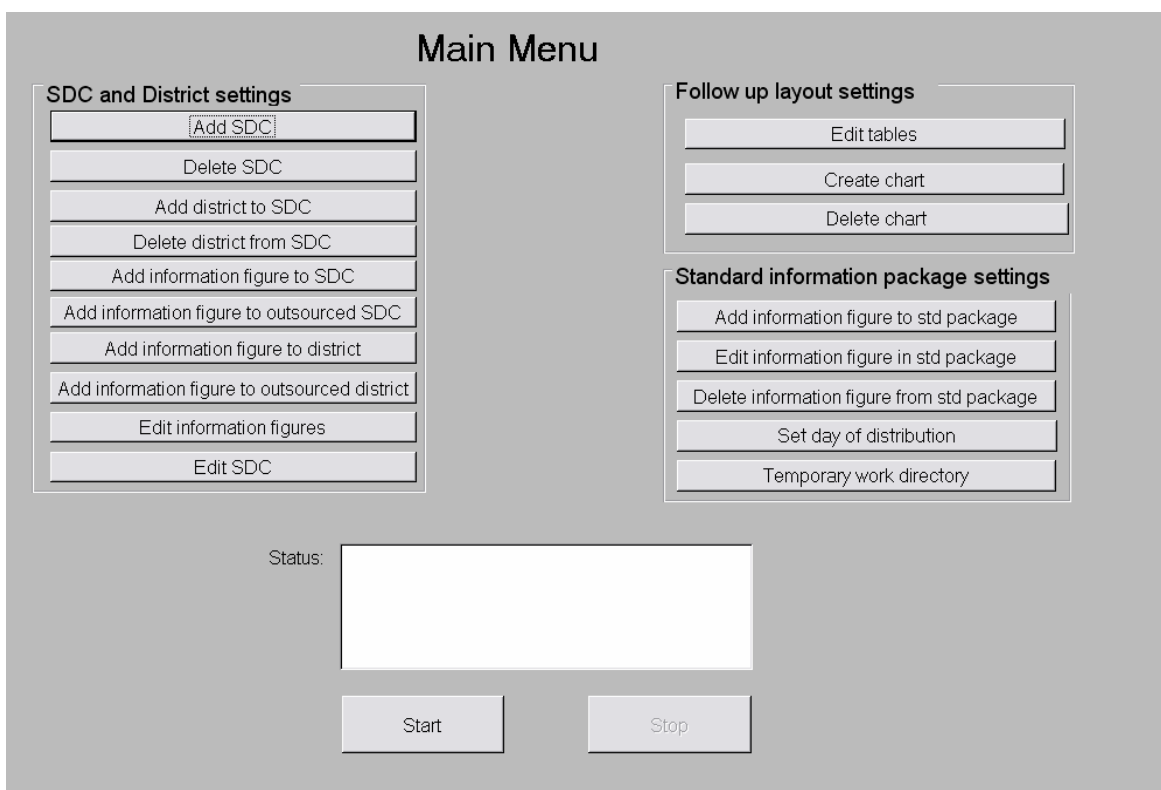
Användargränssnittet behövs för att göra systemet flexibelt och användbart i längden då rutiner och krav på redovisning av information ständigt ändras på VCCS.

Användargränssnittet är anpassat för endast en användare; administratören som skall göra uppdateringar i konfigurationen av systemet och återställa systemet efter eventuella systemkrascher.

Funktionerna i användargränssnittet

- Lägg till ett nytt SDC (tillsammans med distributionsadress).
- Ta bort SDC.
- Lägg till marknad till SDC.
- Ta bort marknad från SDC.
- Lägg till en parameter från informationspaketet till ett specifikt lager.
- Lägg till en ny parameter i informationspaketet.
- Ta bort en parameter från informationspaketet.
- Lägg till värden i tabeller om det skulle fattas information från stordatorsystemet.
- Konfigurera grafer för Exceldokumentet.

Inställningar sparas i tabeller i databasen



Figur 2.3: Huvudmeny

2.2.4 Databas

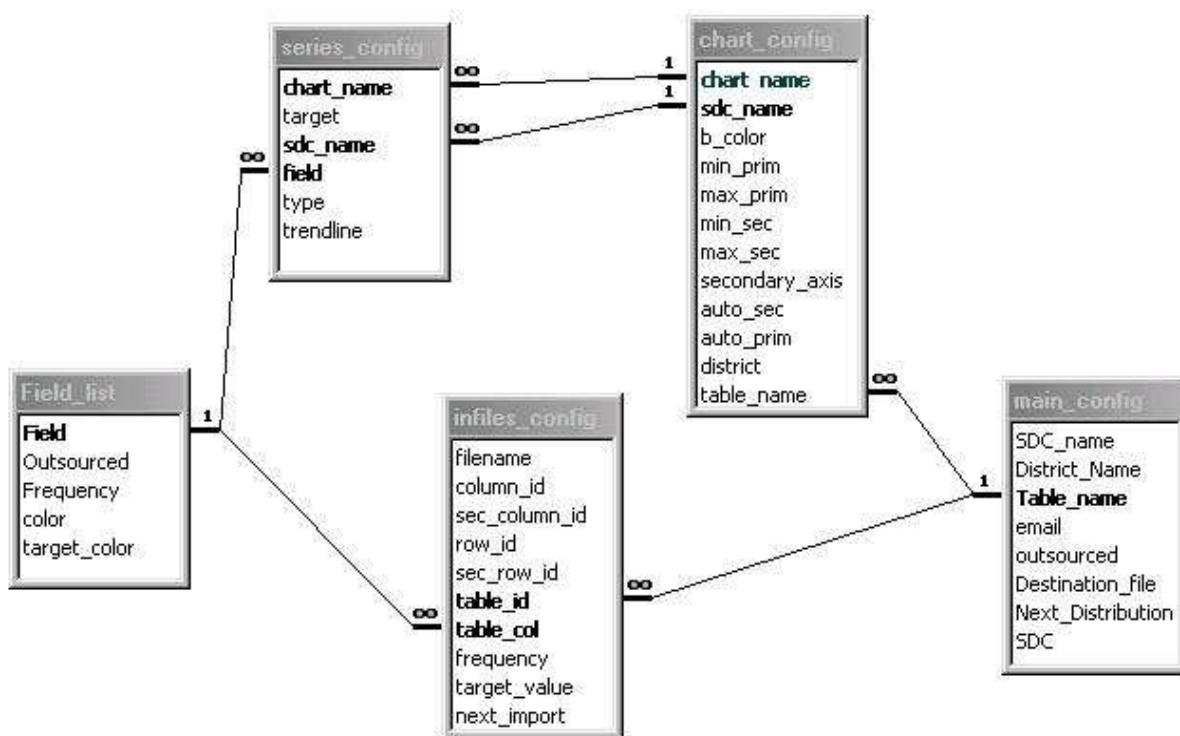
Databasen är kärnan i systemet, här sparas inte bara informationen som skall distribueras utan också inställningar för alla de andra delarna av systemet.

Anledningen till att en databas har använts i systemet är att det är alldeles för riskabelt att lagra uppföljningar i Exceldokument mellan uppdateringarna.

För uppföljningarna skapas en ny tabell för varje uppföljningsfrekvens. Varje SDC kommer alltså ha en separat tabell för de parametrar som skall följas upp veckovis, månadsvis och kvartalsvis. Varje rad i dessa tabeller har unika identifierare i form av år och vecka, månad eller kvartal så tabellerna kan innehålla uppföljningar för flera år.

Det finns fem konfigurationstabeller:

- `main_config`
Huvudtabellen som innehåller information om SDC:erna.
- `infiles_config`
Tabell för konfiguration av inläsningen av värden från stordatorsystemet, har en rad för varje parameter tillhörande ett visst SDC.
- `field_list`
Definition av alla parametrar i informationspaketet.
- `chart_config`
Innehåller inställningar för graferna i Exceldokumentet, en rad för varje graf i ett Exceldokument.
- `series_config`
Innehåller inställningar för varje serie i en graf. En serie är en sekvens med mätvärden, till exempel antalet packade rader för ett antal veckor.



Figur 2.4: Relationsdiagram av konfigurationstabellerna

2.2.5 Grafgenerering

Grafgenereringen skapar slutprodukten av systemet vilket är grafer och tabeller i Exceldokument. Det finns ett specifikt antal typer av grafer som skall produceras med olika attribut som till exempel mållinje och trendlinje. För att specificera vilken data som skall representeras i graferna, var graferna skall lagras och vilken typ av serier som grafen skall innehålla används konfigureringsstabellerna chart_config och series_config. Konfigurering av dessa attribut sker genom användargränssnittet. Traverseringen av konfigureringsstabellerna och skapandet av grafer samt exportering av tabeller sker med hjälp av Data Access Objects (Se 3.1.1 om DAO) och Excel's objektmodell (Se 3.1.2).

Varje Exceldokument har ett distribueringsdatum som sätts senare i distribueringsdelen. Ett Exceldokument produceras bara om dess distribueringsdatum är utgånet.

2.2.6 Distribution

Distributionsdelen av systemet bifogar Exceldokumentet i e-mail och skickar dem till de e-mailadresser som konfigurerats i användargränssnittet som finns sparade i tabellen main_config.

Konfigurerings Tabellen traverseras med hjälp av DAO och ett e-mail för varje fil som har ett utgången distribueringsdatum skickas ut. Distribueringsdatumet uppdateras till ett nytt korrekt datum efter att e-målet har sänts.

Eftersom distributionsdelen kommer efter grafgenereringsdelen i automatiseringsprocessen och ett nytt datum för distribution sätts sist av allt skyddar detta mot att en distribuering går förlorad vid en systemkrasch. Efter en systemkrasch kommer eventuella ej färdiga grafgenereringar och distribueringar att fångas upp eftersom deras datum fortfarande är utgången. I värsta fall så kraschar systemet efter att e-målet har skickats men innan distribueringsdatumet har uppdaterats, då kommer ytterligare ett e-mail att skickas efter omstart. Detta anses dock bättre än att en uppföljning går förlorad.

Skapandet och skickandet av e-mail görs genom Outlook och dess objektmodell.

3 Tekniker

3.1 Objektmodeller

Alla program som ingår i Microsoft Office; Access, Excel, Word och Outlook har en struktur som är uppbyggd av objekt som går att ändra och läsa på olika sätt genom Visual Basic programmering. Dessa hierarkiska strukturer kallas objektmodeller.

Längst upp i hierarkin finns application-objektet som är föräldern till alla andra objekt, utan application-objektet kan inte några andra objekt existera.

Varje objekt har ett innehåll och en eller fler funktioner, en workbook i Excel har till exempel ett antal sheets som innehåll och funktionen close. En funktion till ett förälderobjekt appliceras på alla dess barnobjekt, om till exempel en workbook stängs, stängs även alla sheets för denna. För att kunna ändra eller läsa ett specifikt värde ur ett objekt så måste hierarkin traverseras tills man når det objekt där värdet finns. De högre nivåerna i trädet innehåller vanligtvis en mängd objekt där dessa adresseras genom att ange ett indexnummer eller namn.

Det enda som krävs för att använda en objektmodell är programmet ifråga och att det finns en referens till objektbiblioteket innehållande alla objekt och dess funktioner i projektet.

3.1.1 Data Access Objects

Data Access Objects (DAO) är en objektmodell som gör det möjligt att kontrollera och manipulera vilket databasobjekt eller databas som helst, tillhörande en Microsoft Jet Databas. Detta är möjligt genom applikationer som innehåller Visual Basic for Applications som till exempel Access, Word och Excel. Ett DAO objekt kan antingen representera strukturen i en databas eller själva datan. Access innehåller en Jet databas.

Huvudfunktionerna i DAO är:

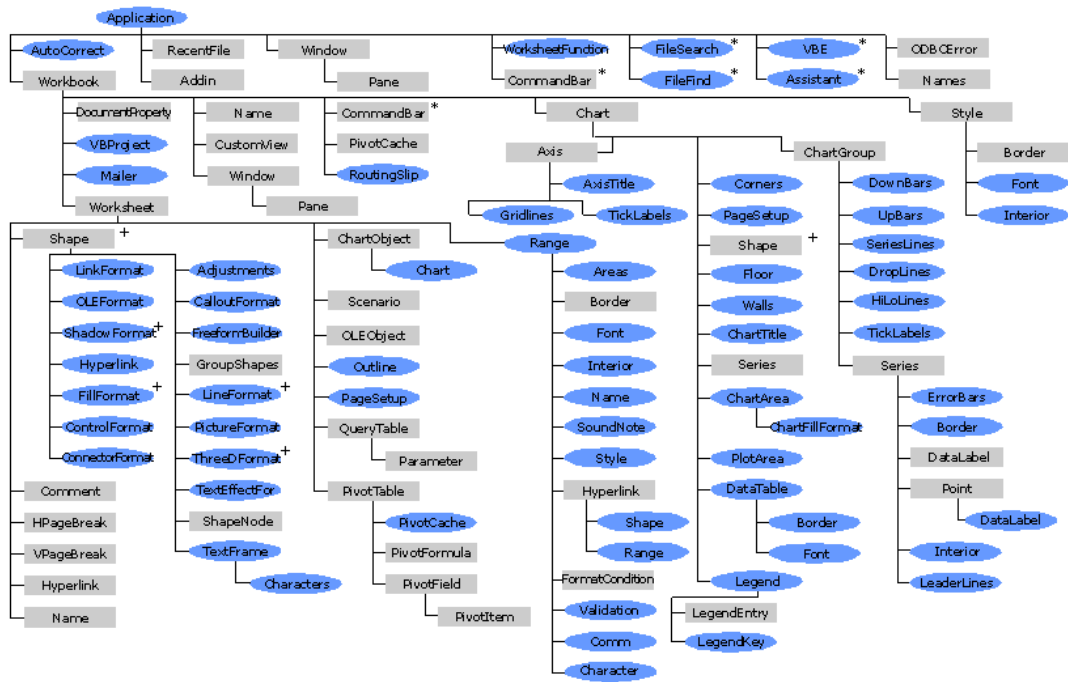
- Skapa och ändra en databasdesign.
- Hämta, ändra och lägga till data i en databas.
- Hämta, länka eller ändra data i andra typer av databaser.

För vidare information om DAO se bilaga A eller [1].

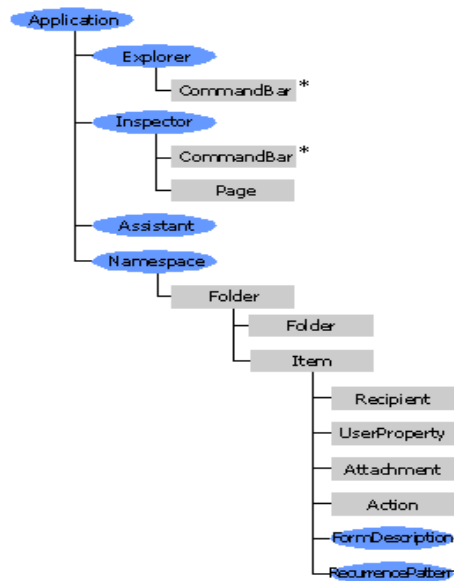
Med ISAM drivaren installerad är det möjligt att till exempel gå igenom sin mailbox i Outlook genom DAO eller länka den till en tabell i en Access databas men man kan inte skapa nya objekt på ett enkelt sätt. Därför används inte DAO-teknik i detta system för kommunikation med program utanför databasen utan bara för hantering av tabeller i databasen. Kommunikationen med externa program sker istället med deras egna objektmodeller.

3.1.2 Excels och Outlooks objektmodell

Alla objekt och funktioner som finns tillgängliga i respektive användargränssnitt finns också tillgängliga i objektmodellen. Detta gör det mycket praktiskt och enkelt att automatisera manuella arbetsuppgifter.



Figur 3.1: Excels objekträd



Figur 3.2: Outlooks objekträd

3.2 Relationsdatabas

Relationsdatabas (RDBMS - Relational Database Management System) är den mest spridda typen av databas idag. En relationsdatabas är uppbyggd av tabeller som består av fält (kolumner) och poster (tupler). I varje tabell måste varje tupel vara unik, det vill säga det får

inte finnas några dupletter. Genom att använda primära nycklar fås på ett enkelt sätt unika tupler, en primär nyckel kan bestå av en eller flera fält.

De olika tabellerna i en relationsdatabas har som namnet antyder relation till varandra. De kopplas samman via gemensamma fält.

3.3 SQL

Structured Query Language (SQL) är ett språk som används för att kommunicera med en relationsdatabas. SQL är en standard som bestämts av ANSI (American National Standards Institute), vilket betyder att ingen äger rätten till språket och kan därför användas av vem som helst. På grund av att det är en öppen standard och att allt fler producenter av DBMS (Database Management System) inkluderar SQL, är SQL idag en mycket spridd standard.

Frågorna kan innehålla instruktioner för att skapa, läsa, ändra eller radera data. Dessa skickas till databasen som sedan tolkar frågan och ger tillbaka ett svar som består av en delmängd av databasen.

Frågorna ställs i en form som är ganska nära människans sätt att tänka till exempel

”SELECT COUNT(personnummer) as antal FROM Sverige WHERE kön = man;”

vilket betyder; räkna ihop alla män i Sverige och returnera svaret. På detta sätt kan de mest komplexa frågorna skrivas.

För mer information om SQL se [2].

Exempel på några kommandon i SQL-språket.

- SELECT
- UPDATE
- DROP
- DELETE
- MODIFY

3.4 Visual Basic

Visual Basic (VB) är ett händelsestyrt programspråk. Med händelsestyrt menas att applikationen väntar på att någon händelse skall inträffa. När händelsen inträffar exekverar programmet en viss kod och sedan går tillbaka till vänteläge.

VB kan även hantera procedurbaserade applikationer, det vill säga där applikationen själv styr i vilken ordning koden skall exekveras.

För grundligare beskrivning av VB kan läsaren gå in på [1].

4 Erfarenheter

De erfarenheter vi har fått genom att göra vårt examensarbete på Volvo är många. När vi först kom till VCCS var vi tvungna att snabbt sätta oss in i vår uppgift. Då denna information var väl utspridd bland anställda på företaget lärde vi oss att prata med okända människor och hur viktigt det var att ställa rätt frågor. När vi inte ställde helt tydliga frågor fick vi inte heller några tydliga svar.

När vi under förstudien försökte samla in vilka parametrar SDC-cheferna ville att informationspaketet skulle innehålla fick vi väldigt många svar, en stor del av dessa var onödiga och överflödiga. Detta lärde oss att folk gärna passar på och önskar mer än vad uppgiften kräver och att man inte skall ge personer allt för fria händer men ändå samtidigt hålla trevlig ton. Vi märkte också att förändringar i rutiner kunde vara svårt för vissa, somliga tyckte att vårt projekt var en alldeles utmärkt idé medan andra tyckte mindre om de förändringar och arbete projektet medförde.

När vi först skrev kravspecifikationen visade det sig att denna var otillräcklig. Dokumentet beskrev både vad och hur systemet skulle byggas och var väldigt ospecifik. Detta har givit oss en djupare insikt av hur en kravspecifikation skall vara utformad.

Vår tid på VCCS har varit väldigt givande. Vi har varit med om en utvecklingsprocess som innefattat allting från kravinsamling till realisering av kraven i form av ett fungerande system.

5 Slutsats

Vi har inte haft några större motgångar under projektets gång. Men i början av projektet fanns det gott om dödtid då vi väntade på svar från kravinsamlingen och tid för möten. Hur man skall kunna utnyttja den tid bättre är dock en svår fråga då man inte vet så mycket om projektet och om vad som skall göras vid denna tidpunkt. Om vi däremot hade givit användarna, som vi samlade in kraven från, lite mindre fria händer kunde man kortat ner denna tidsperiod och därigenom kunna undvika några av de onödiga diskussionerna som låg utanför projektet.

Projektet har i slutändan löst flera av de problem som fanns på VCCS. Den huvudsakliga uppgiften med att ta fram ett gemensamt informationspaket har lösts, vilket var primärmålet. Detta har också haft en positiv bieffekt då vi var tvungna att ta fram en entydig definition av varje parameter, något som inte tidigare fanns på företaget. SDC-cheferna behöver inte längre underhålla uppföljningsdokumenten vilket har lett till att risken för felaktiga uppgifter blivit minimala.

Sekundärmålet som innebar att konstruera ett system har också lösts. Systemet konstruerades även med ett grafiskt användargränssnitt som uppfyllde de insamlade krav. Det automatiska skapandet av uppföljningar och distribuering av dessa har också lett till tidsbesparingar.

Allt blev dock inte riktigt lika bra som det var tänkt från början, Volvo IT misslyckades att utveckla alla program för att ta fram parametrar ur stordatorn som vi hade specificerat. Systemet är dock konstruerat så att det är möjligt att lägga till önskade parametrar till informationspaketet vid ett senare tillfälle genom det grafiska användargränssnittet.

Referenser

- [1] www.msdn.microsoft.com/library/ (t.v. i meny välj: Office Solutions Development->Microsoft Office->Microsoft Office 97->Product Documentation->Microsoft Office 97/Visual Basic Language Reference)
- [2] www.sql.org

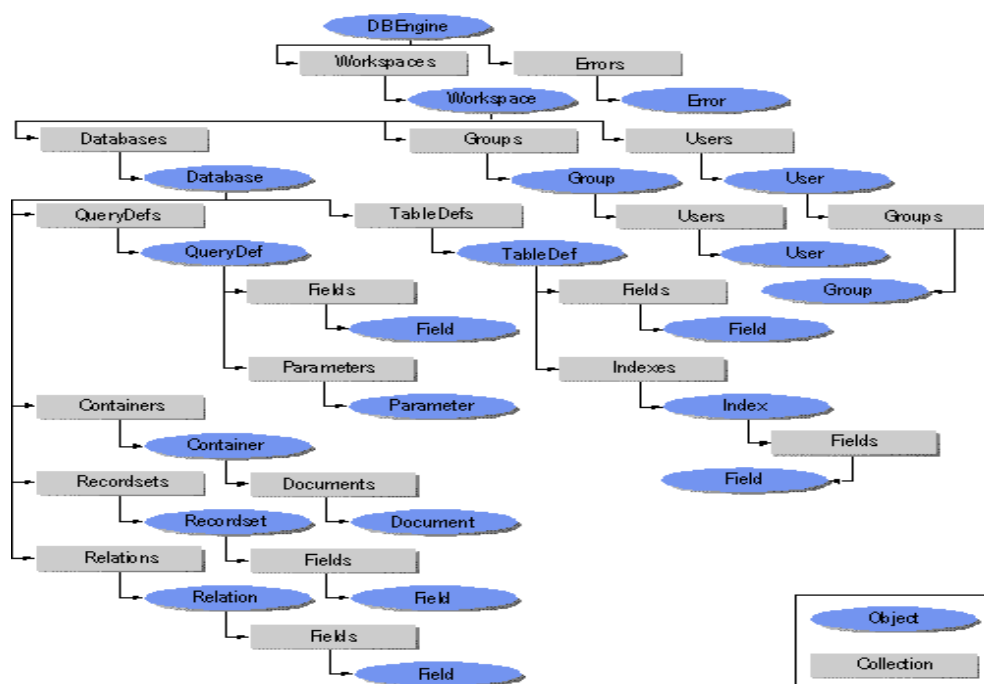
A DAO

Denna bilaga är en fördjupning av Data Access Objects (DAO) från kapitel .

A.1 Skapa och ändra databasdesign

Det är möjligt att skapa och ändra designen i en databas helt och hållet från Microsoft Visual Basic for Applications med hjälp av DAO, men det är betydligt omständigare än att använda användargränssnittet i Microsoft Access för detta ändamål. Alla databasfunktioner som till exempel att skapa tabeller eller SQL-frågor är tillgängliga, den enda funktionaliteten som går förlorad med den här metoden är användandet av formulär och rapporter men om detta skulle vara ett nödvändigt behov kan man använda Access objektmodell.

I objektmodellsträdet är databasmotorn i roten och själva databasen finner man först en bit ner i trädet i mängden databases.



Figur A.1: Microsoft Jetdatabas objektträd

Genom att traversera det hierarkiska trädet så kan man modifiera så väl designen av tabeller som rättigheterna för användare.

A.2 Hämta, ändra och lägga till data i en databas

För att kunna modifiera eller lägga till data i en databas måste man precis som när man vill ändra designen för ett objekt ha en referens till den önskade datan i databasen och även här används samma hierarkiska träd.

A.3 Hämta, länka eller ändra data i andra typer av databaser

Med hjälp av en drivare vid namn Indexed Sequential Access Method (ISAM) kan man arbeta med andra typer av databaser genom DAO.

Databaser som stöds av ISAM drivaren är:

- Microsoft FoxPro
- dBASE
- Paradox
- Microsoft Excel worksheets
- Microsoft Exchange/Outlook
- Lotus 1-2-3 spreadsheets
- Delimited och fixed-width textfiler i tabell format
- Tabeller i HTML filer

B Slutgiltig parameterlista

Detta är den slutgiltiga parameterlistan som togs fram under förstudiefasen. Systemet skall alltså tillhandahålla denna information i de Exceldokument som distribueras.

Parameter	Frekvens
<i>Packed lines</i>	weekly
<i>Packed weight</i>	weekly
<i>Packed volume</i>	weekly
<i>Packed cases</i>	weekly
<i>Packed value</i>	weekly
<i>Invoiced lines</i>	weekly
<i>Invoiced value</i>	weekly
<i>Packed value per hour (all hours)</i>	weekly & monthly
<i>Binned lines per hour</i>	weekly
<i>Logistical turnover rate from SDC</i>	weekly
<i>Packed lines per hour (all hours)</i>	weekly & monthly
<i>Leadtime</i>	weekly
<i>Lines binned</i>	weekly
<i>VOR lines from SDC</i>	weekly
<i>Service level</i>	weekly
<i>Refill leadtime from CDC</i>	weekly
<i>Total Value Refill (daily basis)</i>	weekly
<i>Availability</i>	weekly
<i>Backorder lines</i>	weekly
<i>Inventory status standard cost (Active parts)</i>	weekly
<i>Inventory status number of partnumbers (Active parts)</i>	weekly
<i>Inventory status standard cost (Passive parts)</i>	weekly
<i>Inventory status number of partnumbers (Passive parts)</i>	weekly
<i>Inventory value</i>	weekly
<i>Inventory adjustments +/- physical/value</i>	weekly
<i>Nil picks</i>	weekly
<i>Superseded parts</i>	monthly
<i>Empty Locations</i>	weekly
<i>Parts stocked</i>	weekly
<i>Refill deviations</i>	weekly
<i>Discrepancies 00/20/4X/6X/72</i>	monthly
<i>Lines binned per area (mezzanine,racking,bulky)</i>	monthly
<i>Picking percentage per area (mezzanine/racking/bulky)</i>	quarterly
<i>Received Lines</i>	weekly
<i>Not Invoiced not packed lines</i>	weekly
<i>Not Invoiced packed lines</i>	weekly

Tabell B.1: Slutgiltig parameterlista

C Kravspecifikation