

HANDELSHÖGSKOLAN VID
GÖTEBORGS UNIVERSITET
INSTITUTIONEN FÖR INFORMATIK

Småföretags elektroniska handel över Internet

**Hur genomförs utvecklingen med ASP för att koppla en
befintlig Accessdatabas till Internet?**

Lena Gustafsson, Anette Svensson och Marita Åkerlind
EXAMENSARBETE 10p ingående i ADB-PROGRAMMET 80p
Vårterminen 1998
Handledare: Wera Tegner Johansson

Småföretags elektroniska handel över Internet.

Abstrakt

Vårt examensarbete beskriver metoder och verktyg för en klient, med syftet att användas inom elektronisk handel för småföretag, från en Access- databas till Internet mot EDI. Informationen hämtades från facklitteratur, Internet och handledning av personal från Sema Group. Vi utvecklade en modul i ett system för ordermottagning via Internet. Befintlig Visual Basic- kod omvandlades till VB- Script och HTML. Slutmålet var att beskriva tillvägagångssätt och utvecklingsmiljöer för att ta fram den fungerande modulen.

Småföretags elektroniska handel över Internet.

INNEHÅLLSFÖRTECKNING

1. INTRODUKTION	6
1.1 Bakgrund	6
1.2 Frågor/Problem.....	6
1.3 Avgränsningar.....	7
1.4 Syfte.....	7
1.5 Metod	7
1.6 Intervjuer.....	7
2. UTVECKLINGEN	8
2.1 Användbarhet	8
3. VÅRT TILLVÄGAGÅNGSSÄTT	9
3.1 Val av verktyg och miljöer.....	9
3.2 Kopplingar.....	9
3.3 Session- objekt.....	10
3.4 Händelseförfarande	10
3.5 Problemlösningar.....	10
4. TEKNIKEN	11
4.1 Vad måste göras för att skapa en koppling till databasen?	11
4.2 Komponenterna mellan databas och applikationen.....	11
4.3 Database Access Component	12
4.4 Hur används de huvudsakliga objekten i ADO?	12
5. EDI	14
5.1 Ramar för elektronisk affärskommunikation.....	15
6. KOPPLING MOT EDI	16

6.1 Meddelandets väg från sändare till mottagare via AMTriX.....	17
6.2 Definition av EDI- flöde	18
7. SÄKERHET MED AUTOMATISK IDENTIFIERING.....	19
8. VERKTYGEN OCH MILJÖERNA.....	21
8.1 MILJÖER.....	21
8.1.1 ASP	21
8.1.2 Jämförelse av en vanlig HTML- sida och en ASP- sida	21
8.1.3 Vad kan användas av ASP- sidor?	22
8.2 VERKTYG	23
8.2.1 Access	23
8.2.2 Microsoft VISUAL INTERDEV	23
8.2.3 VB Script.....	24
9. DISKUSSION	25
ORDLISTA	26
KÄLLFÖRTECKNING	27
Litteratur.....	27
WEBB- adresser	27
KODBILAGA Bilaga 1.....	29
RELATIONSSCHEMA	Bilaga 2
33	
INTERVJU MED UTVECKLARE	Bilaga 3
34	

1. INTRODUKTION

1.1 Bakgrund

I vårt examensarbete beskriver vi metoder och verktyg från en Access- databas till Internet mot EDI (Electronic Data Interchange).

Beskrivningen gäller småföretags elektroniska handel via Internet.

Vi blev tillfrågade om vi ville göra ett arbete hos Sema Group på en avdelning som heter Echonet. Där skulle vi koppla en existerande databas mot Internet. Eftersom vi aldrig har gjort något liknande i skolan ansåg vi det vara en intressant uppgift.

Idén med Echonet är att vara ett kunskapsföretag som ska utvecklas av och för små- och medelstora företagen. För de här målgrupperna utvecklas:

- Ett enkelt och billigt medium för företagets marknadsföring och information.
- Ett öppet forum för offertförfrågningar.
- Billig och pålitlig postgång direkt från kundens vanliga dataprogram.
- Förenklade rutiner för beställningar.
- Billig uppkoppling till EDI- systemen för elektronisk handel.

Echonet är utformat som en webb- plats på Internet och fungerar som en modern handelsplats. Echonets kärna är en server. Över den här servern förmedlas alla affärstransaktioner mellan köpare, säljare, banker och intressenter. Servern har även länkar till EDI. Säkerheten i Echonet sker genom kryptering av kommunikationen och lösningar för bl. a inloggning finns.

Idag finns så gott som inget material som beskriver just metoderna för att sköta elektronisk handel för små företag. Det enda som vi hittade var en uppsats från vårterminen 1996 som heter ” Från dokument till EDI- dataöverföringar”. Det måste vara ett billigt alternativ eftersom ett litet företag inte har de ekonomiska resurserna för att kunna investera i stora system. De ska kunna sitta hemma på sina små kontor och koppla upp sig till Internet och sköta sina beställningar eller ordermottagningar.

Genom att bygga en modul i ett fungerande system har vi fått inblick i arbetsmetoderna. Vi har även gått en heldagskurs i ASP (Active Server Pages) som ökade vår förståelse för utvecklingsmiljön. Vi har fått använda oss av olika verktyg och miljöer för att kunna göra kopplingarna och få ett fungerande system. De verktygen och miljöerna ger vi en beskrivning av. För att få en överblickbar bild av ämnet var vi även tvungna att lära oss lite om EDI. På Sema Group har vi fått en föreläsning om EDI som var mycket givande.

1.2 Frågor/Problem

Hur genomförs utvecklingen med ASP för att koppla en befintlig Accessdatabas till Internet?

Vilka verktyg och miljöer ska vi använda?

Hur fungerar verktygen och miljöerna?

1.3 Avgränsningar

Endast beskriva verktygen, miljöerna och metoderna till och från det lilla företaget.

1.4 Syfte

Syftet med uppsatsen är att beskriva metoder och verktyg för en klient, med syftet att användas inom elektronisk handel för små företag, från en Access- databas till Internet mot EDI.

1.5 Metod

Vi har studerat litteratur, gått en kurs och fått handledning av personal från Sema Group. Sema Group har även tillhandahållit oss med litteratur, rapporter, låtit oss delta i en kurs om ASP och gett oss en föreläsning om EDI, som har hjälpt oss att se helheten i vår uppgift. Sanningshalten i materialet vi har tagit från Internet är svårt att kontrollera. Vi har även intervjuat olika personer som arbetar inom det här området och tagit vara på deras erfarenheter.

Grunden för vår uppsats ligger till största delen av en metod vi själva fick utveckla. Det innebär att en hel del arbete har lagts ned på programmering till ett fungerande system. Verktygen och miljöerna som vi har arbetat i är Visual Basic, Access, InterDev och ASP. Koderna vi har använt oss av har varit VB- Script och en del HTML.

1.6 Intervjuer

Frågorna i vår intervju syftar till att ge svar på det praktiska förfarandet vid utvecklingen av ASP- sidor och hur utvecklarna upplever verktygen (se bilaga 3). Vi ville även ta reda på om det är svårt att följa med i utvecklingen eftersom den går fort framåt. Vi valde två utvecklare som arbetar på olika arbetsplatser och är rutinerade av programmering i ASP- miljö. Thomas Hartwig och Mark Gustavsson valdes p.g.a. att de arbetar på olika kontor på Sema Group. ASP är en ny miljö, vilket innebär att det finns lite att utgå ifrån, i form av t.ex. hjälp. Utvecklarna fick frågor som var utformade på så sätt att de fick berätta om hur de upplever verktygen, miljöerna och hur de utformar sina program och kopplingar.

Vi intervjuade:

Mark Gustavsson, Sema Group, Göteborg.
Thomas Hartwig, Sema Group, Malmö.

Thomas Hartwig har även endagarskurser om ASP för anställda på Sema Group. Vi deltog i en kurs som handlade om att skapa enkla ASP- sidor, användarinbyggda objekt i ASP, ADO- databasmodeller och komponentbaserade webblösningar.

2. UTVECKLINGEN

I och med den snabba utvecklingen av nya verktyg och metoder som är kompatibla med varandra händer mycket inom det här området. Från början var allt innehåll på webben statiskt hyperlänkat. Den statiska modellen innebär att HTML- dokumenten skapas och publiceras på en webbserver, bläddraren frågar efter en sida på webbservern, servern tar emot förfrågan och skickar tillbaka HTML- filen och bläddraren formaterar och visar innehållet för användaren. Innehållet på sidorna förblir oförändrat till dess att någon manuellt ändrar det. Det innebär att möjligheterna till interaktionen mellan klient och server blir begränsade. Olika standarder har utvecklats för att förbättra möjligheterna för interaktion mellan klient och server. Det finns CGI (Common Gateway Interface) och ISAPI (Internet Server API). CGI är långsamt och ISAPI DLL är svårt att programmera mot. Microsoft har därför utvecklat ASP (Active Server Pages). Miljön finns i IIS (Internet Information Server 3.0) och bygger på ISAPI. I ASP läggs exekverbara script till direkt i HTML- sidan. ASP tillåter användningen av dynamiska element i en HTML- sida. Webb- applikationer som är utvecklade med ASP är kompatibla med flera olika bläddrare. All bearbetning görs på servern och därefter skickas resultatet till klienten.

De verktyg och miljöer vi har arbetat med är utvecklade av Microsoft och kräver att Microsofts produkter används. Genom att använda Microsoft Visual InterDev kan dynamiska webbprogram byggas. Det är en integrerad utvecklingsmiljö med osynlig anslutningsbarhet till alla ODBC- baserade databaser.

2.1 Användbarhet

De nya verktygen och miljöerna för webbapplikationer som kommer ut på marknaden gör att bl.a. handeln över Internet kommer att öka väldigt mycket. Med de nya miljöerna och verktygen kommer nya möjligheter, som ger ökad funktionalitet. Kopplingen mellan Internet och databaser gör att administrationen förenklas och informationen kan uppdateras med den egna webbläsaren. När informationen läggs ut på Internet ger det ökad tillgänglighet. En stor fördel är att användaren slipper speciella installationer av programvaror och datafiler. Dessutom fungerar Internet mot alla plattformar och det är också kostnadseffektivt. Affärsnyttan med Internet är stor och det skapas fler och fler webbutiker, som erbjuder olika tjänster. Exempel på handel är måttbeställda kläder direkt från tillverkaren. Nackdelen med det är att arbetstillfällen i mellanleden försvinner. En annan tjänst är matvaruhandel. Via Internet, görs en beställning och den blir levererad till hemmet vid en bestämd tidpunkt. Här ses den motsatta effekten, då det här skapas fler arbetstillfällen.

Vi har byggt en modul i ett system där små företagare kan köpa en tjänst för en mindre summa. I framtiden kommer vår del att ingå i ett större system.

Enligt vår uppdragsgivare Sema Group fanns inte något klart system för den tjänsten tidigare.

3. VÅRT TILLVÄGAGÅNGSSÄTT

3.1 Val av verktyg och miljöer

För att överhuvudtaget kunna använda ASP vid utvecklingen är det nödvändigt att ha NT Server. Nedladdning av Internet Information Server (IIS) 3.0 eller 4.0 från Microsofts hemsida måste ske.

Det första att ta reda på var vilket eller vilka verktyg, som behövdes. Mycket tid gick åt till att leta efter information på Internet om de olika verktygen. Efter diskussioner med personal från Sema Group, blev slutresultatet att vi skulle använda Microsoft Visual InterDev.

Funderingar på att använda Frontpage fanns också, men anledningen till att valet blev Microsoft Visual InterDev var att undvika den extra kod, som automatiskt genereras i Frontpage. Även Mark Gustavsson som blev intervjuad sade –” Nackdelar med Frontpage är att det skapas mycket skräpkod, som är svår att överblicka”(se bilaga 3).

Som nybörjare kan det vara svårt att få en bra överblick på grund av den automatiskt genererade koden. En annan skillnad är att i Microsoft Visual InterDev skrivs mer kod av programmeraren själv, vilket ger mer lärdom.

En befintlig databas fanns redan tillgänglig. Den databas som vi har använt oss av är en Access- databas, men SQL Server är annars ett bra alternativ. Access- databasen, som vi har arbetat med, skall senare göras om i SQL Server.

När hämtning skett av de tillbehör som behövdes för utvecklingen, började arbetet i Microsoft Visual InterDev.

Vi började med att skapa en arbetsyta på servern. På den kan läggs bl.a. till ASP- filer och HTML- filer.

Microsoft Visual InterDev skapade en grundläggande kodsida med HTML, vilket gav en början att bygga vidare på.

3.2 Kopplingar

Eftersom den huvudsakliga uppgiften var att kunna hämta och skicka uppgifter till och från en Access- databas var ett av våra första steg att skapa en koppling till den befintliga databasen. Det gjordes genom att först skapa en ODBC- datakälla i kontrollpanelen, som gavs ett namn. För att utnyttja datakällan skapade vi en koppling till i en global fil i Microsoft Visual InterDev. Den globala filen (global.asa) är egentligen inte nödvändig, men är bra att använda. ASP läser filen när webbservern tar emot den första förfrågan på en ASP- sida då webbservern har startats.

Datakällan skapades med hjälp av Connection- objektet. En kopplingssträng skapades med de parametrar som angav var datakällan fanns, namnet och så vidare. Kopplingssträngen användes sedan av Connection- objektet.

3.3 Session- objekt

När databasen anropats i den globala filen med ett Session- objekt i programmet, kunde vi därefter utnyttja Session- objektet i våra andra filer och med hjälp av det öppna kopplingen till databasen. Session- objektet används för en användare och avslutas när tiden gått ut eller när användaren avslutat sin uppkoppling mot Internet.

För att kontrollera en användare skapades en inloggningssida, där användaren får ange sitt användarnamn och lösenord. Användarnamnet och lösenordet bestämmer vilka uppgifter från databasen som användaren kommer åt, eftersom uppgifterna i databasen är kopplade till ett bestämt användarnamn och lösenord.

Med hjälp av Session- objektet kan kontroll av användarens användarnamn och lösenord på efterföljande sidor ske, så att användaren får rättighet till korrekt information.

3.4 Händelseförfarande

Om användaren har angivit ett giltigt användarnamn och lösenord kommer han eller hon in till huvudsidan. Här väljs olika alternativ från en meny.

OrderEnligtKund är en sida, på vilken användaren söker efter en order med antingen kundnr, ordernr eller leverantörsnamn. För att hämta uppgifterna skapades texttrutor med hjälp av HTML, som tilldelades värden om användaren tryckt på en av de valfria sökknapparna. Eftersom sökning skulle ske med utgångspunkt från de olika sökvalen skapades select-satser som hämtade värden från databasen med hjälp av en parameter, som valdes beroende på tryckt sökknapp.

En knapp för Ny sökning laddade om OrderEnligtKund- sidan och hämtade alla kundnr, ordernr och leverantörsnamn från databasen.

På en sida kallad Order fördes det över uppgifter från OrderEnligtKund- sidan. På den sidan gjordes sökningar på ordernummer för den enskilde användaren.

Från Order kunde ett annat formulär öppnas, där utskrift på en speciell order kunde ske.

3.5 Problemlösningar

Vår tanke var att vi med hjälp av en befintligt Visual Basic- applikation skulle kunna ”översätta” den största delen av koden för att använda den på Internet. Dock är det mycket som skiljer Visual Basic och Visual Basic Script åt. Därför fick vi tänka om till viss del när det gällde exempelvis händelser, som inte fungerar på samma sätt i Visual Basic Script. Det var också skälet till att tre sökknappar skapades i det fall då vi ville söka på olika alternativ.

En blandning av HTML och Visual Basic Script var nödvändig för att utföra flera moment.

Informationen från databasen hämtade vi genom att använda SQL- frågor.

4. TEKNIKEN

4.1 Vad måste göras för att skapa en koppling till databasen? ¹

För att kunna hämta information från eller till databaser på en ASP- sida måste först en databaskoppling till ODBC skapas.

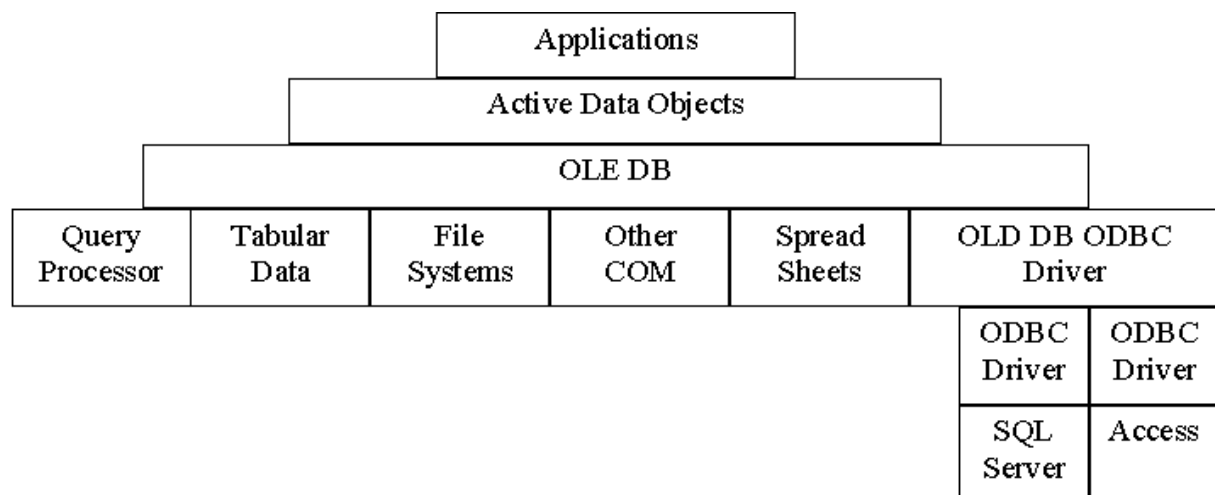
En DSN (Data Source Name) beskriver en datakälla, och standardinställningar som används vid koppling till den.

Det finns fyra typer av datakällor, Användar- DSN, System- DSN, Fil- DSN och Maskin- DSN².

- Användar- DSN är till för en speciell användare på en maskin.
- En System- DSN används för att de som utnyttjar en speciell dator skall komma åt kopplingen.
- Fil-DSN tillåter användare att dela på en DSN.
- Maskin- DSN visar användare och de olika System- DSN som finns tillgängliga på den lokala maskinen.

För att skapa en koppling till en Microsoft Access- databas används Microsoft Access Driver, som drivrutin. Att ange den nya datakällans namn och vilken databas kopplingen skall gälla för, är också nödvändigt.

4.2 Komponenterna mellan databas och applikationen



Bilden visar de olika komponenterna som tillsammans skapar kopplingarna för dataöverföring mellan databas och applikation.³

¹ <http://iis3.activeserverpages.com/aspsamp/tutorial/lessons/database.asp> (1998-04-05)

² <http://www2.fyrisskolan.uppsala.se/books/vid/ch31.htm> (1998-05-18)

³ Homer, Professional Active Server Pages, Handbok

4.3 Database Access Component

En komponent som ingår i ASP är ”the Active Database Access component”. Den består av flera objekt, som kallas ADO.

I ADO finns flera olika egenskaper och metoder, som gör att användaren kan komma åt information från alla sorters databaser. Data hämtas vanligtvis från en relationsdatabas i en klientapplikation, men kan även hämtas från andra källor, exempelvis ett kalkylblad i Excel. ADO används för att arbeta mot databaser via ODBC och kan användas med en datakälla om den har en ODBC- drivrutin.

Eftersom vårt arbete har handlat om att visa hur information hämtas från en Access- databas mot Internet och EDI är det i huvudsak denna ODBC- Driver som beskrivs. Access- ODBC använder syntaxer som många andra ODBC- drivrutiner inte kan använda.

ADO är baserat på ett skikt som kallas OLE- DB. P.g.a. att skiktet består av olika källor måste det specificeras när en datakälla ska länkas.

Det räcker inte att endast känna till drivrutinens mjukvara, t.ex. ODBC. Att känna till vilken källa som datan hämtas ifrån är också ett måste. ODBC är den mest använda OLE- DB- källan.

4.4 Hur används de huvudsakliga objekten i ADO?

De huvudsakliga objekten i ADO är Connection, Recordset och Command.

Connection- objektet används för att skapa en aktiv koppling, vilket gör datan tillgänglig. Server- objektet har en metod för att skapa ett objekt, som heter Createobjekt. Med hjälp av den här metoden kan en instans skapas för en koppling.

Beroende på hur kopplingen ska användas kan den skapas varje gång en datakälla anropas eller så skapas den en gång på en sida och används flera gånger.

Det går också att skapa en instans som tar upp applikationsutrymme eller etapputrymme. Något att ta hänsyn till vid val av instansen för kopplingen är att en koppling ger bättre prestanda vid åtkomsten av data.

I ODBC- version 3.0 finns en metod för att kunna ha kopplingar för flera användare (connection pooling). Används den här metoden är det bäst att öppna och stänga datakällan på varje sida, som använder den för att förbättra effektiviteten.

- Connection- objektet har flera metoder och egenskaper, som kan användas för att arbeta med objektet.
- Open- metoden används för att öppna en koppling till en datakälla. I den här metoden anges en kopplingssträng, eventuellt användarnamn och eventuellt lösenord. Kopplingssträngen kan antingen innehålla en DSN eller mer specificerade kopplingsparametrar. De parametrar som finns är ”Provider”, ”File Name”, ”Data Source”, ”User” och ”Password”.

- Execute- metoden används för att hämta data med hjälp av en lagrad procedur, ett SQL-uttryck eller en tabell. Förutom någon av de här parametrarna kan ytterligare parametrar läggas till för att förbättra effektiviteten.
- Close- metoden används för att stänga en koppling till en datakälla. För att ha lite mer kontroll på en hel kedja av händelser som sker vid ändring i en databas kan transaktioner utnyttjas. BeginTrans- metoden och CommitTrans- metoden avgränsar en transaktion. De här metoderna är bra att använda till exempel om det blir fel. Då används metoden RollbackTrans för att ångra de ändringar som gjorts.

Genom att använda ADO vinnns bl.a. hög hastighet och förenklad användning.¹ Objekt kan skapas oberoende av varandra, vilket innebär att förflyttning inte behöver ske genom en hierarki för att skapa objekt. Därför skapas och spåras de objekt som behövs, vilket leder till att färre ADO- objekt och även ett mindre arbetsområde utnyttjas.

¹ <http://www.microsoft.com/msdn/sdk/inetsdk/help/ado/idx01.htm> (1998-04-05)

5. EDI

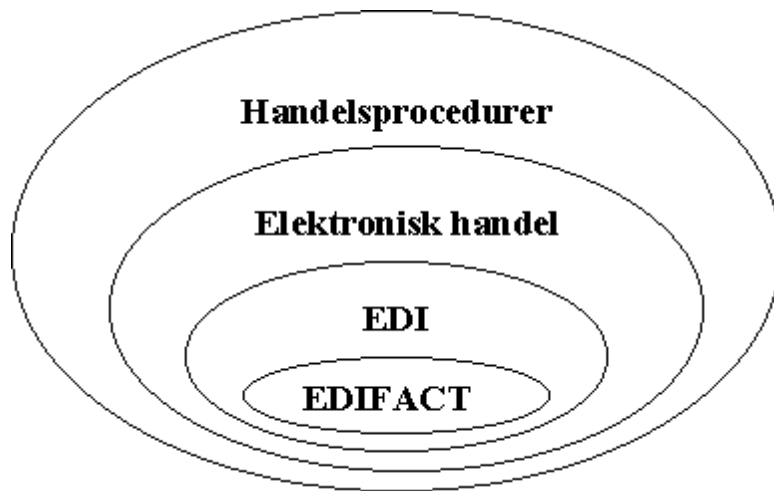
EDI står för Electronic Data Interchange och innebär att affärssystem, t.ex. order-, lager-, fakturerings- och ekonomisystem, kan utbyta information direkt, med ingen eller begränsad mänsklig inblandning. En särskild EDI- programvara sköter översättningen mellan EDIFact-standardens format och det format som används av det interna administrativa systemet. EDIFact- standarden kan liknas vid ett språk som gör att datorsystemen förstår varandra, oavsett om de två systemen som utbyter information finns i olika datormiljöer. De olika informationssystemen får med EDI en ”lös koppling” till varandra, d.v.s. , att de är sammankopplade men ändå oberoende.

Typiskt för EDI är att det används mellan organisationer som har regelbundet återkommande kommunikation av strukturerad information. Det kan vara beställningar mellan ett varuhus och dess leverantörer, leveransplaner mellan en monteringsfabrik och dess underleverantörer, betalningsorder mellan ett företag och en bank, uppgifter mellan ett företag och en myndighet eller provsvar mellan ett laboratorium och en vårdcentral. I normalfall finns ett affärsavtal eller liknande överenskommelse mellan parterna som EDI- kommunikationen baseras på. EDI, åtminstone i sin nuvarande utformning, passar inte för användning där transaktionerna sker sällan. Vår uppgift var att utveckla en modul i ett system över Internet, för att anpassas till det lilla företagets behov.

Då det inte finns någon formell definition av EDI, bör följande kriterier vara uppfyllda för att en överföring ska kunna kallas EDI- kommunikation.

- Direkt kommunikation mellan informationssystem, internt eller externt, via telekommunikation eller lokala nätverk. Det sker utan manuella arbetsinslag och inte via disketter eller band.
- Standardiserat format, t.ex. EDIFact. Det är inte ett egendefinierat filformat.
- Strukturerad information, typiskt sådant blanketter tidigare använts till och alltså inte personliga meddelanden och andra ostrukturerade informationer.
- Processbart hos mottagaren. Det ska inte behövas manuella insatser för att få data registrerat i det mottagande administrativa systemet.
- Oberoende av hårdvaruplattform, operativsystem, datatyp, typ av informationssystem, tid och kommunikationsmetod hos de olika parterna.
- Tillförlitligt är ett nyckelord. EDI innebär ofta att system utbyter data automatiskt utan mänskliga kontroller vilket gör att eventuella fel kan orsaka stor skada. Aspekter som inbyggda kontroller i system och olika slags säkerheter i överföringen blir därför mycket viktiga.
- Baserat på avtal, det vill säga mellan parter som redan har ett etablerat samarbete.

5.1 Ramar för elektronisk affärskommunikation



EDIFact är en del av EDI- konceptet, vilket i sin tur är en av teknikerna för elektronisk handel. Alla metoderna ska i sista änden förbättra ett företags handelsprocedurer.¹

¹ Fredholm, Elektroniska affärer, Studentlitteratur

6. KOPPLING MOT EDI

Stora företag och förvaltningar har idag gjort stora investeringar för hantering av EDIFact. De lösningarna är oftast bra.

Målsättningen nu är samma funktionalitet med bryggor mellan Internet och befintliga EDIFact- lösningar. Hälften av alla kunder som är små företag kan ej använda EDI idag. Systemintegration måste ske där data flyttas mellan olika system.

Ett exempel är lastbilstransporter. Om alla låg on- line med olika ordermottagningar skulle planeringen bli betydligt bättre. Eftersom många lastbilar går tomma åt ena hållet skulle mycket pengar kunna sparas med bättre planering. Idag sitter telefonister och tar emot preliminärbokningar som sedan inte alltid stämmer när hämtning sker av godset. I nuläget innebär det att lastbilar ofta körs halvfyllda. Det här innebär större kostnader för både åkare och kund. Idag är koppling mellan preliminärbokningen och lastplaneringen i det närmaste obefintlig. Med hjälp av GPS som är en satellitnavigering kan sambandscentralen hålla reda på var lastbilarna befinner sig och på det sättet utnyttja de som är närmast en nya beställningar. Används EDIFact och/ eller Internet blir planeringen betydligt bättre.

Många branscher skulle kunna använda sig av den här lösningen. Behovet av utveckling inom det här området är mycket stort.

Nackdelarna är att nu rationaliseras människor bort inom förvaltning vilket tidigare skett inom produktion. Den nya inriktningen på olika arbeten kommer att bli utveckling.

Kopplingar mot EDI innebär att två system automatiskt via ett nätverk ska kunna kommunicera med varandra. För att kunna utnyttja de kopplingar som behövs krävs mycket stor säkerhet med kvittenser, speciellt när det gäller stora företag.

När det är små kunder som själva sköter utväxlingen av meddelanden krävs inte lika hög säkerhet.

Stora företag har så kallade Inhouse- system, som är ekonomiska system. Inhouse- filerna ligger inom företagets eget system. Dessutom är det vanligast att de större företagen har en egen EDIFact- konverterare. Konverteringar sker från Inhouse- systemen så att de blir till EDIFact- meddelanden. De meddelandena skickas sedan vidare.

SFTI (Single Face To Industry) är en vanlig standard för att konvertera meddelande till EDIFact. En annan standard är Odette (Organization for Data Exchange by Tele Transmission in Europé).

Konvertering sker från EDIFact till en semikolon- separerad ASCII - sträng. En EDI- växel som heter AMTriX sköter konverteringen. I AMTriX som tar emot meddelanden finns olika mappar för konvertering. En av dem heter Cleaning House. Den konverterar också EDIFact- meddelanden.

Om det är t.ex. två kommuner som skickar meddelanden till varandra behövs ingen Cleaning House användas, eftersom deras EDIFact- meddelanden stämmer överens med varandra.

Odette är speciellt vanlig inom bilindustrin. T.ex. Volvo och Scania använder sig av den för bl.a. att kunna kommunicera internationellt mot t.ex. Renault. Konvertering sker t.ex. mellan Odette och SFTI.

Det finns även många andra standarder som bl.a. ICA och KF har. Ett exempel är KF:s standard som heter KF- bas. Många företag har sina egna standarder p.g.a. sina interna kommunikationer.

Kunden som använder sig av det system i vilket vi varit delaktiga i under utvecklingsarbetet, köper ett paket med ett abonnemang. I abonnemanget ingår en hemsida med fördefinierade formulär för order, ordersvar och fakturor. Vid konvertering i AMTriX måste formatet vara enligt en viss standard.

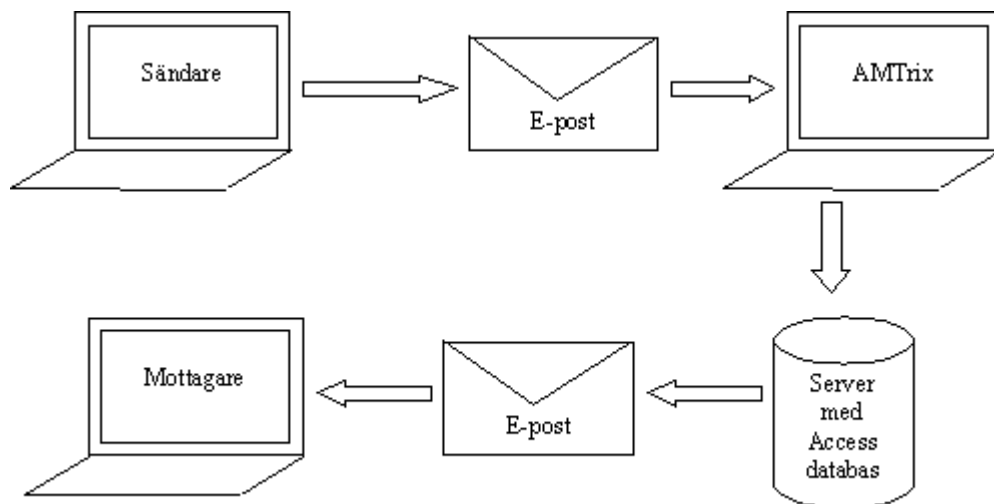
Senare kommer även formulär för leveransavisering och orderändring att ingå på hemsidan. Kundens egen del i databasen, där alla kunders uppgifter lagras, kommer att vara tillgänglig genom lösenords- och användar- identifikation. På det viset kan kunden kontrollera om nya order ankommit och innehållet i alla sina order.

Det finns tre olika slags kunder i nuläget. Det är för det första kunder som använder de fördefinierade formulären, vilkas innehåll skickas via e- post. De order som skickas via e- post identifieras med företagsnamn och kommunkod.

Det är även de kunder som har större volymer av mottagna eller sända order. De kunderna exporterar en fil som innehåller t.ex. många order. Den konverteras om för att passa till databasen. Det förfarandet sker för att kunder ska slippa sända ett formulär för varje order. Säkerheten för sändningarna görs genom kontroll av inloggning och ger dessutom möjlighet att spåra ordern i databasen.

Den tredje kategorin av kunder är de som kör full EDI med egen EDIFact. I det fallet sker loggningen via TCP/IP.

6.1 Meddelandets väg från sändare till mottagare via AMTriX



EDIFact- meddelandet går till AMTriX för konvertering.

AMTriX sänder en ASCII- fil som konverteras ut till en katalog på webb- servern. En programvara känner av om det ankommit några nya order Teckensträngen ligger kvar på servern och det sänds ut en e- post med en länk till hemsidan som avisering till företaget. Samtidigt lagras ordern i databasen. Kunden kommer vidare genom att klicka på länken till inloggningssidan i systemet.

När ordermottagaren kontrollerat sin del i databasen känner en programvara även av det och en orderbekräftelse sänds till beställaren.

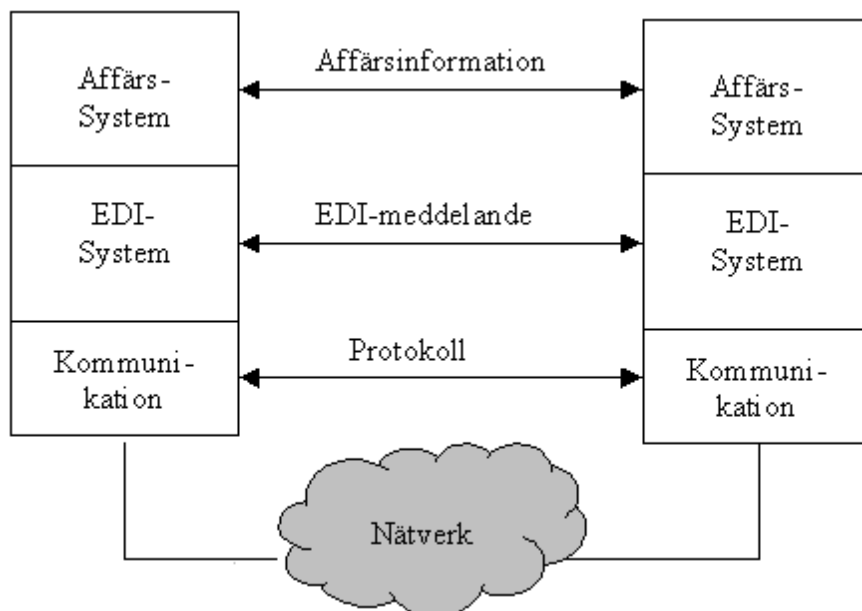
En webb- browser är tillsluten som en säkerhetsåtgärd för att det inte ska kunna gå att läsa, ändra eller förstöra på en annan användares hårddisk. Det är inte ett enkelt sätt att ha ett lokalt register som läser in data från klienten till webbservern eller från webbservern till klienten. Varken klienten eller webbservern mot bläddraren får kommunicera med lokala register.

Vid konvertering från AMTriX till SQL görs semikolon- separering.

En process startas med en länk om avisering som sänds som en URL till en egen del av SQL-databasen. Den egna delen av databasen är skyddad med användarnamn och lösenord.

Säkerheten måste vara pålitlig. När datan kommer till SQL- databasen sker en avisering om en ny order till kunden. En fråga som idag ställs är om det är utvecklingsbart och hur? Svaret på den frågan lär framtiden utvisa.

6.2 Definition av EDI- flöde



Bilden visar hur affärssystem kan utbyta strukturerad information enligt EDI- konceptet via EDI- system, kommunikationsprotokoll och nätverk. ¹

¹ Fredholm, Elektroniska affärer, Handbok

7. SÄKERHET MED AUTOMATISK IDENTIFIERING

Vid kommande hantering av fakturor och ordermottagningar i vårt arbete kommer tvådimensionell kod att användas. Anledningen till det är att öka säkerheten vid identifiering och texthantering, då ett helt EDI- meddelande får plats på en liten kodbit.

Ett viktigt komplement till den elektroniska handeln med varor är automatisk identifiering, även kallat ADC (Automated Data Capture). Det som vanligen förknippas med Automatisk identifiering (Auto ID) är streckkod, vilket endast är en av flera tekniker som används. De olika teknikerna för automatisk identifiering är:

- Streckkoder
- Radiofrekvens (RF)
- Magnetkod
- Optical Character Recognition (OCR)
- Röstinmatning
- Vision eller inläsning med kamerateknik
- Biometrisk verifikation

Själva grundtanken med Auto ID är att varorna får ett unikt nummer som är uppbyggt efter en viss standard. Det här numret läggs som en streckkod som kan läsas genom en läspenna eller scanner direkt in i datorn.

Fördelar med Auto ID

- Den manuella inläsningen av långa artikelnummer försvinner.
- Risken för fel minskar (ger endast ett fel på 3 miljoner lästa tecken).
- Inläsningen går snabbare (12 tecken tar 0,3 sekunder)
- Tekniken är billig
- Robust teknik och utrustning som klarar tuffa industriella miljöer.
- Pris- och artikeländringar, inventeringar och beställningar förenklas.

Streckkoder började användas redan under andra världskriget, men det riktiga genombrottet kom när den amerikanska handeln ville öka effektiviteten vid kassorna genom att utnyttja streckkoden.

I logistiksammanhang har streckkoderna den fördelen att det går att koppla ihop det fysiska flödet med informationsflödet. Det ger en ökad effektivitet vid arbete i lager, lastning under transport och vid godsmottagning. När alla leden använder samma streckkod minskar risken för fel och hela kedjan kan härledas från tillverkaren till kunden.

På det stora hela innebär streckkodsanvändning kombinerat med standardiserade artikel- och kollen- identiteter stora effektiviseringsvinster. Kombinerat med EDI- kommunikation ökas vinsten ytterligare.

Småföretags elektroniska handel över Internet.

Hela tanken med streckkodstekniken är att siffror, bokstäver och andra tecken omvandlas till ett format som kan läsas maskinellt. Det görs genom att olika tjocka mörka streck och dess ljusa mellanrum bildar en streckkodbit. En ljusstråle sveper över denna kodbiten och ger en signal som förstärks, digitaliseras och tolkas av läsutrustningen. Det gör att en vara eller ett kolli kan identifieras. Förfarandet går snabbt och ökar effektiviteten. Det som kan bli intressant med strekkodsanvändning är nyheten med att skapa tvådimensionell kod. Det vanligaste i nuläget är PDF 417. Den här metoden bygger inte alls på streck, utan är mer som ett raster som kan läsas av en scanner.

Fördelar med PDF kod

- Hög driftsäkerhet
- En stor del av koden kan förstöras men ändå läsas
- Koden kan innehålla upp till 700 tecken vilket räcker ganska långt

8. VERKTYGEN OCH MILJÖERNA

Att bygga ett system som användaren tycker är funktionsdugligt är en utmaning. Hänsyn måste tas till att användaren ska kunna förstå och tycka att det är lätt att sköta sin handel via Internet. Det finns många möjligheter och avancerade tekniker att använda sig av vid utvecklingen. P.g.a. att vår tid var begränsad fick vi hålla oss till ett begränsat antal verktyg och miljöer. Här följer en kort beskrivning av de verktyg och miljöer vi använt oss av.

8.1 MILJÖER

8.1.1 ASP

”Active Server Pages (ASP) är webb- sidor som innehåller ”server- side scripts”, förutom den vanliga blandningen av text och HTML- taggar.¹

”Server- side scripts” är speciella kommandon, som läggs in i webb- sidor. Kommandon utförs innan sidorna skickas från den personliga webb- servern till webb- browsern (t.ex. Microsoft Internet Explorer) av någon som besöker webb- sidan .

ASP- scriptkommandon och ”utdata- uttryck” skiljs från text och HTML- taggar med avgränsare.

8.1.2 Jämförelse av en vanlig HTML- sida och en ASP- sida

HTML- fil: När det skrivs en URL i adressrutan eller klickas på en webb- sida, skickas det en förfrågan till en webb- server på en dator någonstans om att skicka en fil till webb- browsern, (klienten) på den aktuella datorn. Om filen är en normal HTML- fil, ser den exakt likadan ut när webb- browser får den, som när webb- servern skickade den. När webb- browsern har fått filen visar den dess innehåll, som en kombination av text, bild och ljud.

ASP- fil: Om det är en ASP- sida är processen likadan, förutom att ett extra steg utförs innan webb- servern sänder filen.

Webb- servern kör först alla ”server- side scripts” som sidan innehåller innan den skickar ASP- sidan till webb- browsern. Scripten kan t.ex. visa aktuellt datum, tid och annan information. En del behandlar information som användaren har skrivit in på ett formulär, vilket kan vara exempelvis användaruppgifter.

För att skilja ASP- sidor från vanliga HTML- sidor ges de ".asp" som filformat, vilket inte är nödvändigt.

¹ Adding Active server pages <http://sarge.marion.ohio-state.edu/IE.stuff/active.htm> (1998-04-10)

8.1.3 Vad kan användas av ASP- sidor?

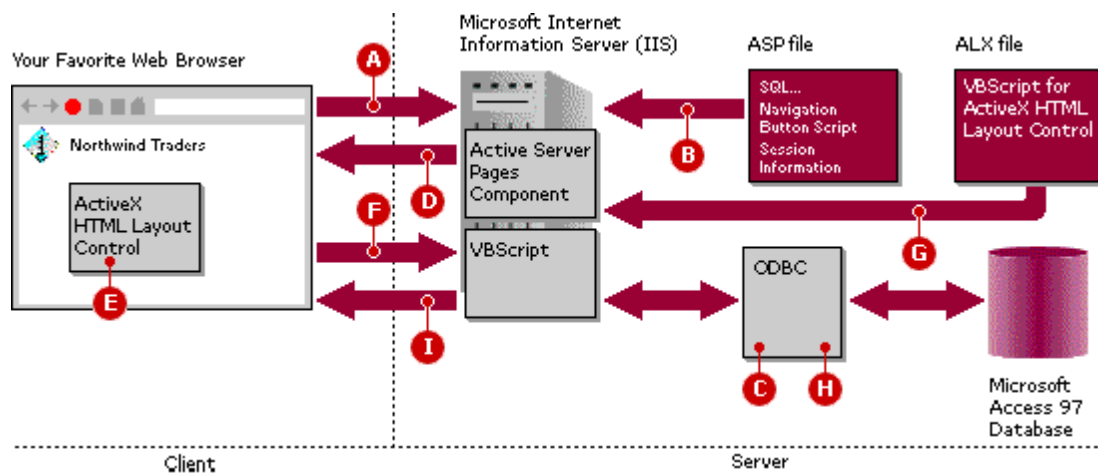
Active Server Pages kan använda det ActiveX- objekt som är installerat på servern. ASP inkluderar fem inbyggda objekt och ungefär tolv ytterligare objekt som kommer med vid installationen av ASP.

ActiveX- objekt skrivna i Visual Basic eller annat språk med en "Active Server Page" kan också användas.

"Active Server Pages" är en komponent av Microsoft Information Server (IIS) 3.0 eller senare.¹

ASP- filen innehåller förutom HTML- taggar, SQL- frågor och Visual Basic Script kod även ODBC- information för att koppla sig till en datakälla, t.ex. Microsoft Access 97.

När databasen publiceras och webb- applikationen installeras på en server händer följande:



Webb- browsern gör en förfrågan till IIS för att få ASP- filen

- A) IIS läser ASP- filen, som innehåller bla SQL- satser och kopplingsinformationer.
- B) VB- script- kod öppnar Microsoft Access- databasen (genom att använda Microsoft Access Desktop ODBC Driver och ASP- fil- kopplingsinformationen) och kör frågan i ASP- filen för att komma åt data.
- C) IIS skickar tillbaka HTML- filen till webb- browsern för att visa den som en webb- sida.

För formulär:

- E) Internet Explorer laddar "the ActiveXtm HTML Layout Control", som används för att sätta positionerna för formulärfält och annan kontroll på HTML- sidan.
- F) Internet Explorer gör en förfrågan om en stödjande ASP- fil.
- G) "The Active Server Pages component" läser den stödjande ASP- filen.
- H) VBScript- kod öppnar Microsoft Access- databasen (genom att använda Microsoft Access Desktop ODBC Driver och ASP- fil- kopplingsinformationen) och kör frågan i ASP- filen för att komma åt data.

¹ "How a Webb Database Application uses ASP Files

<http://www.microsoft.com/accessdev/articles/acdatwebb/page4.htm> (1998-05-03)

Småföretags elektroniska handel över Internet.

ActiveX server komponenter förbättrar scripten genom att göra återanvändning möjlig vid åtkomst av information.¹

Ett exempel är ”The Database Access component” ,som består av olika ADO- objekt. Den gör det möjligt för script att hämta uppgifter från databaser med frågor.

Användning av ”the Database Access component”

”The Database Access component” använder ”Active Data Objects” (ADO) för att lätt komma åt information som lagras i en databas, eller någon annan datastruktur som är kompatibel med ”the Open Database Connectivity” (ODBC) standard.

8.2 VERKTYG

8.2.1 Access

Microsoft Access 97 är ett verktyg för att utveckla relationsdatabaser. Det finns för operativsystemen Windows NT och Windows 95.

I och med den nya versionen, med Microsoft Office 97 finns en del nyheter, som t.ex. möjligheten att skapa HTML- länkar till Internet. Dessutom kan ett worddokument sparas som en HTML- fil, och enkelt visas på Internet.

En ytterligare funktion som kommit med Microsoft Office 97 är Visual Basic For Applications (VBA), som är ett programspråk, som används av alla Microsoft Office 97 – program.

Det går enkelt att integrera Microsoft Access i Visual Basic med ActiveX, som gör att länkar kan skapas till dokument och objekt.

8.2.2 Microsoft VISUAL INTERDEV

Microsoft Visual InterDev är en integrerad miljö för utveckling av webbprogram.² Det är en snabb utvecklingsmiljö med kraftfulla databasverktyg och integrerade funktioner för administration. Med Visual InterDev skapas, publiceras och administreras de nya webbprogrammen. Det går att skapa interaktiva program för flera olika webbläsare och flera olika plattformar. I Visual InterDev ingår Active Server Pages. Programbearbetningen utförs direkt på webbservern. Det används för att göra stabila och skalbara webblösningar. Med Visual InterDev kan webblösningarna dessutom integreras med befintliga klient/server-system. Det går även att koppla webbprogram till gemensamma databaser med bibehållen datasäkerhet och dataintegritet. Verktyg som bygger på ODBC (Open Database Connectivity) ger möjlighet att visa de databaser som används av webbplatsen. Det är ett av den senaste tidens nya utvecklingsverktyg. Det blir mer och mer viktigt att arbeta med kombinationer av verktyg.

¹ <http://iis3.activeserverpages.com/iasdocs/aspdocs/roadmap.asp> (1998-04-21)

² <http://www.microsoft.com> (1998-04-06)

8.2.3 VB Script

VB script är ett scriptspråk utvecklat av Microsoft.¹ Det här språket är inneslutet i HTML kod och tolkas av bläddrare som Netscape och MS Internet Explorer som är kompatibla med VB Script. Det är en speciell del av Visual Basic programmeringsspråk som är helt kompatibelt med Visual Basic och Visual Basic for Applications.

Visual Basic (VB) är ett snabbt utvecklingsprogrammeringsspråk och är ett plattformsberoende språk. VB används ofta i Internet för att skriva program som används för att skicka information till en webb server. Alla VB Script körs på browsern.

För att använda Visual Basic Script i ett HTML dokument måste koden packas in i `<SCRIPT> ... </SCRIPT>` element. Script language måste alltid anges. `<SCRIPT LANGUAGE="VBScript">`. `<!--`De här avgränsningskommentarerna `-->` ser till att koden är gömd för de bläddrare som inte stöder Visual Basic Script.

¹ <http://www.microsoft.com/vbscript> (1998-04-06)

9. DISKUSSION

Vår diskussion kretsar kring framtiden. När så mycket händer inom IT- området är det svårt att förutse vad som kommer att ske framöver.

Att använda sig av elektronisk handel är fortfarande en ny företeelse. Dagens användning är begränsad och tekniken används endast till en liten del av alla affärstransaktioner. Det sker en ständig utveckling av metoder, teknik och standarder. Det lär bli en hel del förändringar i den närmaste framtiden. Handel kommer att bli ett helt nytt begrepp när den blir mer och mer elektronisk och global.

Bakgrunden till den elektroniska handeln var att 1994 lades ett förslag i det årets budgetproposition. Förslaget var att möjligheterna skulle ses över för att gå över till elektronisk handel i den offentliga sektorn. 1995 enades svenska Kommunförbundet, Landstingsförbundet och staten om att införa elektronisk handel före år 2000. Införandet skulle ske inom toppledarforums ram. Anledningen till att elektronisk handel skulle införas var att en faktura i genomsnitt kostar 350 kr att bearbeta och 70 % av alla fakturor är på belopp som är lägre än 3000 kr. Kostnaderna för bearbetningen skulle komma att sänkas betydligt.

Elektronisk handel börjar finnas både bland småföretag och i den offentliga sektorn, inte bara hos de stora företagen. Det innebär krav på ökad kunskap, bättre erfarenhet och enklare tekniska hjälpmedel.

Standardsystem måste i framtiden vara förberedda för funktioner för elektroniska affärer med bl.a. order, lager, fakturering och ekonomi. Inom några år kommer EDI- gränssnitt att vara inbyggt i alla system av betydelse. De administrativa systemen kommer också att förändras då det snart kommer möjligheter till automatiska EDI- förlopp direkt mellan systemen.

Marknaden med EDI- programvara är idag mycket liten. EDI- programvara kommer endast till liten del att köpas direkt av företagen. Programvarumoduler som kommer att användas blir beställningsapplikationer för interna avrop och beställningar, databas över beställningar och leveranser, mottagningsregistrering av leveranser, fakturahanteringsprogram och program som hanterar verifikationsunderlag vid de tillfällen faktura inte används.

EDIFact är redan etablerad som den enda internationella meddelandestandarden och det kommer troligen att förbli så.

Eftersom elektronisk handel över Internet kommer att innebära många fördelar för företagen, kommer säkert utvecklingen att gå framåt med stormsteg.

Företagen ser själva fördelar som minskad pappersanvändning, bättre lagerstyrning, kostnadsreduktion, bättre konkurrensförmåga, ökad kundservice och effektivare informationsflöde enligt Peter Fredholm författare till Elektroniska affärer.

Vi tror att framtidens handel till största delen kommer att ske via Internet. Därför måste även utvecklarna av system hålla sig ajour i det här skedet för att kunna etablera sig på den marknaden. De som har system att sälja redan nu kommer troligen att bli de som blir etablerade inom det här området. Att stora förändringar sker och utvecklingen går snabbt framåt är ingen nyhet för någon. Elektronisk handel är bara ännu ett steg på vägen.

ORDLISTA

ADC - Automated Data Capture är automatisk identifiering

ASP – Active Server Pages, ASP- sidor är textfiler, med script

Cleaning House – Leverantör av mervärdestjänster för elektronisk handel.

DSN – Data Source Name, är databaskällan

EDI – Electronic Data Interchange, elektronisk meddelandeväxling av strukturerad Information , direkt mellan datorsystem, enligt standardiserat format.

EDIFact – Electronic Data Interchange For Administration, Commerce and Transport. Internationell standard för strukturering av transaktioner eller meddelanden.

ISDN – Integrated Services Digital Network. Digitalt uppbyggt nät med förhöjd kvalitet och kapacitet.

ODBC – Open Database Connectivity

ODETTE – Organization for Data Exchange by Tele Transmission in Europe. Den europeiska bilindustrins EDI- projekt.

TCP/IP – TCP står för Transmission and Control Protocol . IP står för Internet Protocol.

VBScript – Visual Basic Script, ett scriptspråk som är väldigt likt Visual Basic.

KÄLLFÖRTECKNING

Litteratur

Fredholm Peter (1997) Elektroniska affärer, Studentlitteratur.
Homer Alex, Enfield Andrew, Gross Christian, Jakab Stephen, Hartwell Bruce, Gill Darren, Francis Brian, Harrison Richard (1997) Professional Active Server Pages, Handbok.
McKelvy Mike, Martinsen Ronald, Webb Jeff (1997) Using Visual Basic 5, Handbok
Wille Christoph (1997) Unlockning Active Server Pages, Handbok.

WEBB- adresser

URL:er på Internet

Building an Active Server Application So easy it's a child's play
<http://www.microsoft.com/devnews/mayjun97/benasp.htm>

Active Server Pages
<http://www.activeserverpages.com/>

Adding Active Server Pages
<http://sarge.marion.ohio-state.edu/IE.stuff/active.htm>

Microsoft Active Server Pages
<http://www.akn-systems.com/IASDocs/ASPDocs/roadmap.asp>

BILAGOR

KODBILAGA

Bilaga 1

Global.asa

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Session_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'--Project Data Connection
Session("DataConn_ConnectionString")
="DBQ=D:\echoshare\edi\Echonet_ed\echonet\EDI\data\Echonet_EDI.mdb;Def
aultDir=D:\echoshare\edi\Echonet_ed\echonet\EDI\data\;Driver={Microsoft
Access Driver (*.mdb)};DriverId=25;FIL=MS
Access;ImplicitCommitSync=Yes;MaxBufferSize=512;MaxScanRows=8;PageTi
meout=5;SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Session("DataConn_ConnectionTimeout") = 15
Session("DataConn_CommandTimeout") = 30
Session("DataConn_RuntimeUserName") = "admin"
Session("DataConn_RuntimePassword") = ""

'==Visual InterDev Generated - DataConnection endspan==
Dim Conn
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open Session("DataConn_ConnectionString")
End Sub
</SCRIPT>
```

När varje Session startar körs koden i Session_OnStart. I Global.asa- filen skapas en koppling som refererar till en användare. Sessionvariabeln DataConn_ConnectionString tilldelas olika parametrar för att de ska kunna använda variabeln som kopplingssträng.

Dels anges hela vägen till databasen, vilken driver som används och ytterligare parametrar.

De här parametrarna har skapats automatiskt.

Sedan dimensioneras en variabel, som används för att referera till kopplingen. Det skapas ett kopplingsobjekt med hjälp av ADO och öppnar kopplingen med Open- metoden.

OrderEnligtKund.asp

```
<% Option Explicit %>
<%
Dim Conn
Dim rs
%>
<HTML>
<HEAD>
<TITLE>Meny</TITLE>
<LINK REL="stylesheet" HREF="Stylesheets/main.css" TYPE="text/css">
```

Den här länken pekar på filen main.css. I main.css finns javascriptkod, som gör att muspekaren blir en hand när musen dras över objektet vilket klickas på.

```
</HEAD>
<BODY BGCOLOR=C0C0C0>
<BASEFONT FACE="Verdana, Arial, Helvetica, sans-serif">
<IMG SRC = "images/econetlogo.gif" WIDTH=297 HEIGHT=74 BORDER=0>
<BR>
<BR>
<%
Set Conn=server.createobject("ADODB.connection")
Conn.Open Session("DataConn_Connectionstring")
%>
```

Här skapas återigen kopplingen till databasen och öppnar den med hjälp av det skapade Session- objektet från Global.asa.

```
<% cmdKundnr = Request("cmdKundnr") %>
<% li_Orderdat = Request("cmbOrderdat") %>
```

För att tilldela variabeln li_Orderdat ett värde från komboboxen skickas värdet till servern med Request- objektet. Även knapparnas värde skickas till servern. Det görs för att kontrollera om knappen har tryckts på eller inte.

```
<% Set rs = Server.CreateObject("ADODB.Recordset")
Set rs = Conn.Execute("select distinct
OrdHead_Ordernr,OrdHead_LevNamn,Ordhead_OrderDatum,Customer_Kundnr
from OrdHead")
%>
```

Ett objekt skapas, som ett recordset. Därefter används kopplingen till databasen för att exekvera en sql- sats i OrdHead- tabellen i databasen.

```
<FORM NAME=OrderEnligtKund ACTION=OrderEnligtKund.asp
METHOD=post>
```

Form- taggen används för att definiera ett formulär och för att specificera den sida som skall köras vid sändning av formulärfälten.

Post- metoden skickar värdena till vald Action- sida. En fördel med att använda Post istället för Get, som också kan användas, är att i Get- metoden är antalet parametrar som kan skickas med i URL:en begränsade.

```
<INPUT TYPE=text NAME=txtKundnr SIZE=22 MAXLENGTH=50

<% If cmdOrderdat <> "" Then %>
    <% li_Orderdat = Request("cmbOrderdat") %>
    <% KundSql = "select distinct
    OrdHead_Ordernr,OrdHead_LevNamn,Customer_Kundnr from
    OrdHead where OrdHead_OrderDatum = '" & li_Orderdat & "'" %>
    <% Set rs=Conn.Execute(KundSql) %>
    VALUE = <%=rs("Customer_Kundnr") %>
<% End If %>

<SELECT NAME=cmbKundnr>
<% If cmdKundnr <> "" Then %>
    <OPTION><% = Request("cmbKundnr") %></OPTION>
<% Else %>
    <% rs.Movefirst %><% '**%' %>
    <% Do While Not rs.EOF %>
    <OPTION><% = rs("Customer_Kundnr") %></OPTION>
    <% rs.MoveNext %>
    <% Loop %>
<% End if %>
</SELECT>
```

Ovanstående kod visar hur en kombobox skapas i HTML. Villkoren innebär att en söknapp är tryckt eller inte.

```
<INPUT CLASS=SubBut TYPE=submit NAME=cmdKundnr VALUE=Sök>
```

SubBut är en function, som hämtas från main.css. Den här funktionen gör att muspekaren blir till en hand.

```
</FORM>
```

```
<FORM NAME=SkickaOrder ACTION=Order.asp METHOD=Post>
&nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
<INPUT CLASS=SubBut TYPE=submit NAME=cmdOrder VALUE=" Order ">
&nbsp; &nbsp;
<INPUT CLASS=SubBut TYPE=button NAME="cmdQuit" VALUE="Avsluta"
OnClick=window.location.href="Huvudmeny.asp">
```

När det klickas på knappen tilldelas URL:en en ny adress.

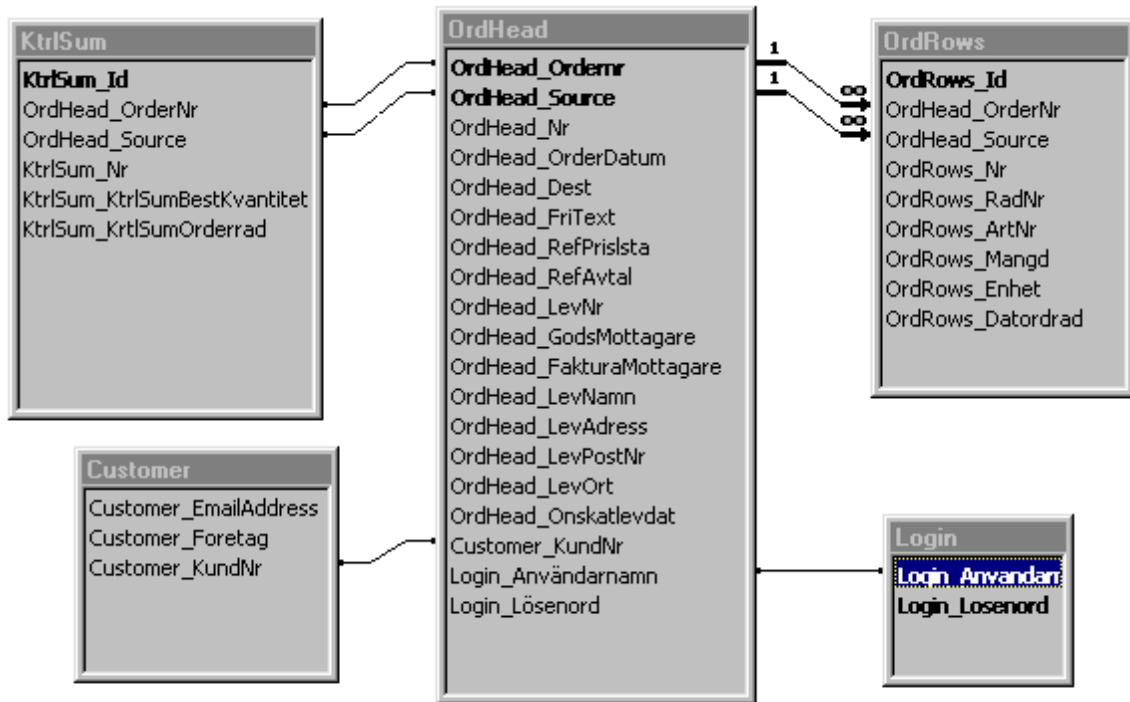

```
<INPUT TYPE=Hidden NAME=HiddenKundnr
<% If cmdKundnr <> "" Then %>
    VALUE="<% = rs("Customer_Kundnr") %>">
<% End If %>
<% If cmdOrderdat <> "" Then %>
    VALUE="<% = rs("Customer_Kundnr") %>">
<% End If %>
<% If cmdLevNamn <> "" Then %>
    VALUE="<% = rs("Customer_Kundnr") %>">
<% End if %>

</FORM>
```

Formuläret SkickaOrder skickar med värden till Order.asp med hjälp av de gömda formulärfält som finns på sidan. Om någon av de tre sökknapparna har tryckts ned tilldelas de gömda formulärfälten värden.

RELATIONSSCHEMA

Bilaga 2



INTERVJU MED UTVECKLARE

Bilaga 3

1. Hur länge har ni arbetet med webb- lösningar?
2. Vilka verktyg arbetar ni i idag ?
3. Vilka andra verktyg har ni arbetat med ?
4. Vilka för- och nackdelar noterade ni med de olika verktygen ?
5. Ser ni några likheter mellan de olika verktygen
6. Vilken miljö föredrar ni?
7. Följer företaget med i den snabba utvecklingen ? Utbildning ?
8. Vilka databaser har ni arbetat med?
9. Vilka är för- och nackdelarna med att arbeta mot en Access- databas jämfört med en SQL- databas ?
10. Använder ni hjälpen ofta?
11. Tycker ni att hjälpen är förståelig och bra?
12. Hur löser ni problem där ni inte kan utnyttja hjälpen?
13. Vilka svårigheter ser ni med att koppla webb- applikationer med databaser ?
14. Hur använder ni connection- objekt, på en eller flera sidor ?
15. Vilken metod använder ni för att ta emot information till databasen från Internet ?

Intervju med Thomas Hartwig, utvecklare i ASP: Sema Group i Malmö 1998-04-29

Thomas Hartwig har arbetat officiellt med webb- lösningar sedan juni 1997, men provade även en del med början hösten 1995.

Då han började 1997 hade ASP precis kommit ut.

De verktyg han arbetar i idag är Microsoft Visual Interdev, Visual Basic, Homesite (ett program för HTML- kod).

Han har även arbetat i Frontpage och Cold Fusion (ett program för kopplingar mot databaser, som är likt ASP)

De nackdelar han ser med att använda Interdev är att det är Microsofts produkt, varför det blir för mycket program från Microsoft. Det är också ett nytt program, som inte har tillräcklig funktionalitet jämfört med Visual Basic.

Fördelar är att det för tillfället är det bästa på marknaden och att det finns en bra hjälp i verktyget.

Fördelar med att använda Frontpage 97 är att det är bra för att skapa gränssnitt. En nackdel är däremot att det inte är bra för dynamiska händelser.

Cold Fusions nackdel är att ASP är mer kraftfullt.

Homesites fördelar är att det är en ”smart” editor, i vilken gränssnitt kan skapas.

Likheter med verktygen är flera, men det finns dels verktyg för programmering (Microsoft Visual Interdev) och verktyg för HTML (det visuella).

Thomas föredrar miljöerna Interdev och Homesite. Han har ej jobbat mycket med Frontpage, vilket kan vara en orsak till att han arbetar med Homesite. Annars skulle han troligen använda Frontpage.

Efter varje projekt sätter de som arbetar sig ned och ser vad som är bra och inte. De ser vad som är nytt på marknaden och provar nya metoder. De ser bara 4 månader framåt, efter 4 månader har det hänt mycket på marknaden och nya metoder bör användas.

De har ej utbildning i ASP, men har tid för att själva sitta och lära sig.

De databaser han har använt är Access, SQL Server, Oracle och Informix.

Fördelar med Access är att det är lätt att flytta med databasen. En fil kan flyttas mellan flera databaser.

Nackdel är att Access ej rymmer mycket information, det är mer en hobbydatabas. Den är slö och det kan ta lång tid att arbeta med Access.

Fördelar med SQL Server är att den är billig sett till marknaden, för att vara professionell. Den är lätt att använda.

Småföretags elektroniska handel över Internet.

Nackdel jämfört med Access är att den då ej är lätt och ej portabel.

Thomas använder hjälpen relativt ofta och tycker den är ganska bra. I början var hjälpen svårbegriplig. Den är ett bra referensverk för den som arbetat ett tag. För att söka efter ett särskilt objekt är den också bra.

Vid problem där hjälpen inte är användbar använder han sig av "Trial and Error", mycket prövning, det finns sällan bra böcker, eftersom ASP är nytt.

Svårigheter med att koppla webb- applikationer mot databaser var i början att förstå tankesättet och användandet. Annars är det relativt lätt, vilket också beror på verktygen. Troligtvis kommer det att bli lättare med nya verktyg.

I början använde han sig av Session- objektet när användaren loggade in och öppnade detta mot databasen. Alltså fick varje användare ett connection- objekt.

Nackdel: Session- objekt försvinner efter en tid och det gör att det blir fel i sena frågor mot databasen.

Nu använder han ett Connection- objekt på varje sida istället, vilket gör att det går långsammare.

Metoder som han använder för att ta emot information till databasen från Internet är Insert-kommandon och Execute.

Han skapar ett Connection- objekt och en executesats med Insert Into och använder ofta lagrade procedurer, d.v.s. befintliga frågor i t.ex. Access. När dessa frågorna anropas kompileras de snabbare.

Intervju med Mark Gustavsson, utvecklare i ASP: Sema Group i Göteborg 1998-05-08

Mark Gustavsson har arbetat med webblösningar sedan 1995.

De verktyg som han arbetar i är Notepad, Visual Basic, Access och SQL Server, men han har också arbetat med Paintshop Pro och Frontpage.

Fördelar med att använda Frontpage är att det är grafiskt och det är lätt att exempelvis rita upp en tabell.

Nackdelar med Frontpage är att det skapas mycket skräpkod, som är svår att överblicka.

Notepads fördelar är att det går att optimera sidorna, t.ex. ta bort onödiga taggar. Det gör att sidorna blir mer överblickbara.

De olika verktygen använder han för en specifik funktion. Frontpage och Notepad är lika. Han använder Frontpage och Notepad ihop. Exempelvis kopierar han kod, som Frontpage genererar och tar bort onödig kod i Notepad.

Mark föredrar Notepad, eftersom han anser att han där har full kontroll på vad som sker.

På Sema Group kommer han att få utbildning i SQL Server. Han hade även möjlighet att delta i en ASP- kurs. Inom Sema Group finns olika internutbildningar.

De databaser han har använt är Access, textbaserad ASCII- databas och SQL Server.

Fördelar med Access är att det är lätt att ta fram en applikation och bygga upp frågor. Det är grafiskt lätt att använda.

SQL Servers nackdel är att kunskaper krävs i SQL- språket. Fördelar är att den är gjord för stora tillämpningar och många användare. Det blir snabbare svarstid och det går att göra större databaser.

Mark använder hjälp bl.a. för att kontrollera syntaxer. Han använder Microsoft Developer Network, som hjälp och tycker att den är mycket bra.

I situationer där han inte kan använda hjälpen frågar han medarbetare. Internet är också ett bra alternativ för att söka hjälp. En webbsida som är användbar är <http://www.webdeveloper.com>. Om ingen kan svara kollar han upp nyhetsgrupper. Finns inte heller där svar försöker han angripa problemet på ett annat sätt. Sista utvägen är att köpa support.

Svårigheter med att koppla webb- applikationer är att ODBC- kopplingen är avgörande. Det är viktigt att SQL- frågorna stämmer, så att resultatet blir rätt.

Han anser att ADO är enkelt vid jämförelse med ISAPI. Det har också blivit lättare om jämförelse görs med CGI- script.

För att skapa en koppling använder han en variabel i Global.asa. Den här variabeln definieras när en ny användare kopplar upp sig. Det är viktigt att skapa variabler så att det endast behöver ändra på ett ställe. För att skicka värden från Internet till databasen använder Mark formulär.